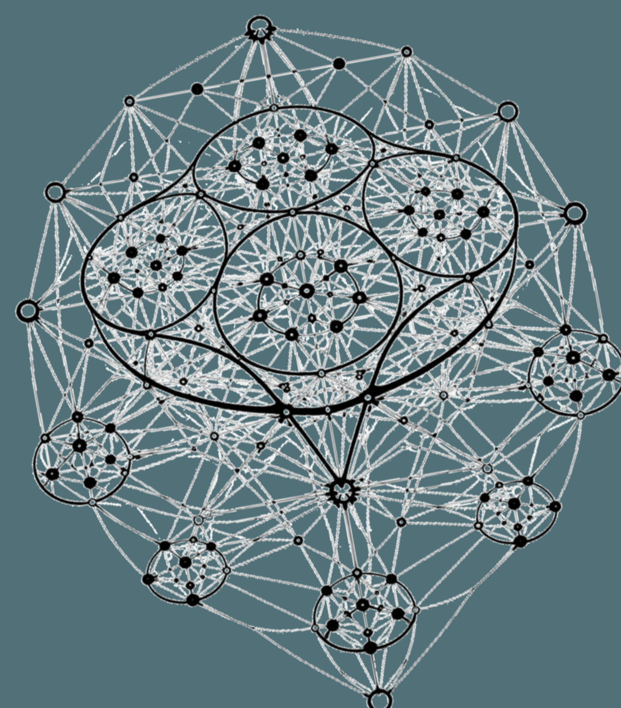
TAKAAKI FUJITA
FLORENTIN SMARANDACHE

HYPERGRAPH AND SUPERHYPERGRAPH THEORY WITH APPLICATIONS VI

GRAPH STRUCTURE

(PATH, TREE, CYCLE, PLANARITY, BIPARTITE, AND MORE)



NSIA
NEUTROSOPHIC SCIENCE
INTERNATIONAL ASSOCIATION
PUBLISHING HOUSE

Takaaki Fujita, Florentin Smarandache

HyperGraph and SuperHyperGraph Theory with Applications

VI

Graph Structure
(Path, Tree, Cycle, Planarity, Bipartite, and More)

Neutrosophic Science International Association (NSIA)
Publishing House

Gallup - Guayaquil
United States of America – Ecuador
2026

Editor:

Neutrosophic Science International Association (NSIA)
Publishing House
https://fs.unm.edu/NSIA/

Division of Mathematics and Sciences
University of New Mexico
705 Gurley Ave., Gallup Campus
NM 87301, United States of America

University of Guayaquil
Av. Kennedy and Av. Delta
"Dr. Salvador Allende" University Campus
Guayaquil 090514, Ecuador

ISBN 978-1-59973-856-7

Table of Contents

HyperGraph and SuperHyperGraph Theory (VI): Graph Structure (Path, Tree, Cycle, Planarity, Bipartite, and More)

Takaaki Fujita [1] * **and Florentin Smarandache**[2]
[1] Independent Researcher, Tokyo, Japan.
Email: Takaaki.fujita060@gmail.com
[2] University of New Mexico, Gallup Campus, NM 87301, USA.
Email: fsmarandache@gmail.com

Abstract

Hypergraphs generalize ordinary graphs by allowing an edge to connect an arbitrary nonempty subset of the vertex set. By iterating the powerset construction, one obtains nested, higher-order vertex objects and arrives at finite *SuperHyperGraphs*, in which vertices are set-valued across multiple layers and edges encode relations among such set-valued vertices. Despite this expressive modeling capacity, the structural theory of SuperHyperGraphs—including systematic studies of their basic properties and parameters—remains comparatively underdeveloped. In this volume, we extend fundamental graph-theoretic notions to the SuperHyperGraph setting, focusing in particular on paths, trees, cycles, planarity, bipartiteness, and closely related concepts, and we investigate their key properties.

Keywords: SuperHyperGraph, HyperGraph, Trees, Cycles, Planarity, Bipartiteness.

Peer-Reviewers:

Maikel Leyva-Vázquez
Facultad de Ciencias Matemáticas y Físicas
Universidad de Guayaquil, Guayas, ECUADOR
maikel.leyvav@ug.edu.ec

Jesús Rafael Hechavarría Hernández
Facultad de Ingenierías, Arquitectura y Ciencias de la Naturaleza
Universidad ECOTEC, ECUADOR
jesus.hechavarriah@ug.edu.ec

Victor Christianto
Malang Institute of Agriculture (IPM), Malang, INDONESIA
victorchristianto@gmail.com

Muhammad Aslam
Faculty of Science, King Abdulaziz University, Jeddah, SAUDI ARABIA
aslam_ravian@hotmail.com

Chapter 1

Introduction

1.1 Graph, HyperGraph, and SuperHyperGraph

Network models are classically expressed by *graphs*, in which objects are represented by vertices and binary relationships by edges [5]. While this abstraction is effective for pairwise interactions, it becomes restrictive when the underlying system exhibits *simultaneous* interactions among three or more entities. *Hypergraphs* resolve this limitation by permitting each hyperedge to join an arbitrary nonempty subset of vertices, thereby representing higher-order relations directly [6–9].

Even so, many real-world datasets and engineered systems display relationships that are not only higher-order but also *layered*, *nested*, and intrinsically *hierarchical*. To capture such multi-level incidence patterns, F. Smarandache introduced the notion of a *SuperHyperGraph*. Informally, a SuperHyperGraph is built via iterative powerset-based constructions, which allow vertices ("supervertices") themselves to be set-valued objects and enable edges to encode nested connectivity across multiple levels [10,11]. Consequently, SuperHyperGraphs have recently attracted growing attention in both theory and applications [12–18].

Graphs and hypergraphs also provide transparent visual metaphors for complex systems and support a broad spectrum of applications in artificial intelligence, network science, data mining, informatics, chemistry, physics, and related fields [19–21]. By explicitly incorporating hierarchical and multi-level relationships, SuperHyperGraphs offer a flexible framework for modeling and analyzing intricate structures in modern networked data (e.g., [16, 22–27]).

Table 1.1 summarizes the essential distinctions among graphs, hypergraphs, and n-SuperHyperGraphs.

A more concrete, side-by-side comparison of graphs, hypergraphs, and n-SuperHyperGraphs is provided in Table 1.2.

Table 1.1.: Key distinctions among graphs, hypergraphs, and n-superhypergraphs.

Concept	*Notation*	*Edge family*	*Core extension principle*
Graph [5]	$G = (V, E)$	$E \subseteq \binom{V}{2}$	Edges encode *pairwise* (binary) relations between vertices.
Hypergraph [28]	$H = (V, \mathcal{E})$	$\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$	Hyperedges may join *any* nonempty subset of vertices, encoding higher-order interactions.
n-SuperHyperGraph [10]	$\mathrm{SHG}^{(n)} = (V, E)$ (on a base set V_0)	$V \subseteq \mathcal{P}^n(V_0) \setminus \{\emptyset\}$, $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$	Vertices are allowed to be *set-valued objects* living in an n-fold powerset hierarchy, while edges remain subsets of the supervertex set; this supports *nested* and *multi-level* incidence patterns.

Notation. $\mathcal{P}(X) = \{A \mid A \subseteq X\}$, $\binom{V}{2} = \{\{u, v\} \subseteq V \mid u \neq v\}$, and $\mathcal{P}^0(X) = X$, $\mathcal{P}^{k+1}(X) = \mathcal{P}(\mathcal{P}^k(X))$.

1.2 Our Contributions

In view of the above, research on SuperHyperGraphs is important. However, systematic study of SuperHyperGraphs is still relatively recent. Consequently, investigations of fundamental structural notions in SuperHyperGraphs—such as paths, trees, cycles, planarity, bipartiteness, and related concepts—remain comparatively limited. To help bridge this gap, in this book we extend these core graph-theoretic structures to the SuperHyperGraph setting and examine their key properties. We hope that these results will contribute, even modestly, to further progress on the structural theory of SuperHyperGraphs.

Table 1.2.: A concrete comparison of graphs, hypergraphs, and n-superhypergraphs.

Aspect	*Graph* $G = (V, E)$	*Hypergraph* $H = (V, \mathcal{E})$	*n-SuperHyperGraph* $\mathrm{SHG}^{(n)} = (V, E)$ (on V_0)
Vertices	$v \in V$	$v \in V$	$x \in V \subseteq \mathcal{P}^n(V_0) \setminus \{\emptyset\}$ (set-valued supervertices)
Edges	$E \subseteq \binom{V}{2}$	$\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$	$E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$
Incidence	$v \in e$ for $e \in E$	$v \in e$ for $e \in \mathcal{E}$	$x \in \varepsilon$ for $\varepsilon \in E$
Adjacency (typ.)	$u \sim v \iff \{u, v\} \in E$	$u \sim v \iff \exists e \in \mathcal{E} : \{u, v\} \subseteq e$	$x \sim y \iff \exists \varepsilon \in E : \{x, y\} \subseteq \varepsilon$
One edge encodes	a binary relation	a multiway relation among vertices	a multiway relation among *supervertices* (which may themselves be nested)
Distance (typ.)	shortest path length in G	Berge distance or primal/2-section distance	super-Berge distance or primal/2-section distance on (V, E)
Use (typ.)	binary links	higher-order groups	hierarchical / nested incidence and multi-level grouping

Notation. $\mathcal{P}^0(X) = X$ and $\mathcal{P}^{k+1}(X) = \mathcal{P}(\mathcal{P}^k(X))$.

Chapter 2

Preliminaries

This chapter establishes notation and reviews the fundamental structures used throughout the book.

2.1 SuperHyperGraphs

Classical graph theory models a system of *vertices* linked by *edges*, and studies connectivity, structural invariants, and algorithmic problems motivated by mathematics, computer science, and many applied domains [5]. A *hypergraph* broadens this framework by allowing a single edge to connect an arbitrary nonempty subset of the vertex set; hence it is well suited to represent intrinsically multiway interactions (e.g., relations of arity greater than two) [6,8]. Such higher-order relations have become especially prominent in contemporary learning and modeling pipelines, including neural architectures that directly leverage hypergraph incidence patterns [6, 29–32].

By iterating the powerset operation, one can also permit *nested* set-valued entities at the vertex level. This leads to finite *SuperHyperGraphs*, in which both vertices and edges may occur at multiple levels of set nesting [33, 34]. Such hierarchical representations arise naturally in layered or multiscale relational settings, for instance in molecular design, complex-network analysis, and neural-network modeling, among other applications [23, 35–37]. Several related generalizations have also been investigated, including Directed SuperHyperGraphs [38, 39] and MetaSuperHyperGraphs [40].

Definition 2.1.1 (Base set)**.** A *base set* S is the ambient universe of discourse:

$$S = \{ x \mid x \text{ is an admissible object in the context under consideration} \}.$$

All sets in $\mathcal{P}(S)$ and in the iterated powersets $\mathcal{P}^n(S)$ are ultimately formed from elements of S.

Definition 2.1.2 (Powerset)**.** (see [41]) For a set S, the *powerset* of S is

$$\mathcal{P}(S) = \{ A \mid A \subseteq S \}.$$

In particular, $\emptyset \in \mathcal{P}(S)$ and $S \in \mathcal{P}(S)$.

Definition 2.1.3 (Hypergraph)**.** [28,42] A *hypergraph* is a pair $H = (V, E)$ such that:

- V is a finite set of *vertices*, and
- E is a finite family of nonempty subsets of V, called *hyperedges*.

Thus, a hyperedge may contain more than two vertices, capturing genuinely multiway relations.

Example 2.1.4 (Real-life example of a hypergraph)**.** Consider the problem of organizing university courses and student enrollments. Let V be the set of all students in a department. For each course c, define a hyperedge

$$e_c := \{ s \in V \mid s \text{ is enrolled in course } c \}.$$

Then $H = (V, E)$, where $E = \{e_c : c \text{ is a course}\}$, is a hypergraph. Each hyperedge represents a multiway relation: all students jointly participating in the same course. For reference, the graph is shown in Fig. 2.1.

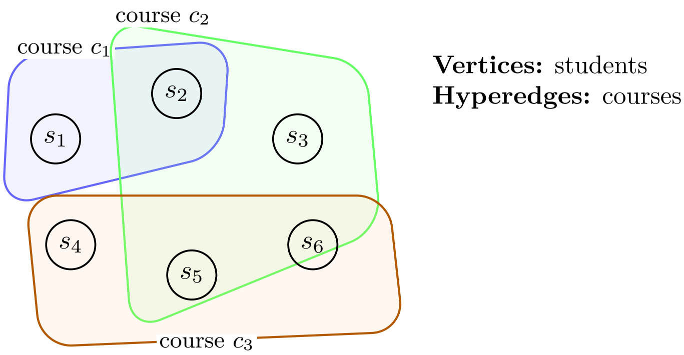


Figure 2.1.: A course-enrollment hypergraph: each course c_i induces a hyperedge $e_{c_i} \subseteq V$ consisting of its enrolled students.

Definition 2.1.5 (Iterated powerset and flattening)**.** [43] Let V_0 be a finite nonempty set. Define $\mathcal{P}^0(V_0) := V_0$ and

$$\mathcal{P}^{k+1}(V_0) := \mathcal{P}\big(\mathcal{P}^k(V_0)\big) \qquad (k \geq 0).$$

For each $k \geq 0$, define the flattening map

$$\mathrm{Flat}_k : \mathcal{P}^k(V_0) \setminus \{\emptyset\} \longrightarrow \mathcal{P}(V_0) \setminus \{\emptyset\}$$

recursively by

$$\mathrm{Flat}_0(x) := \{x\} \quad (x \in V_0), \qquad \mathrm{Flat}_{k+1}(X) := \bigcup_{Y \in X} \mathrm{Flat}_k(Y) \quad (X \in \mathcal{P}^{k+1}(V_0) \setminus \{\emptyset\}).$$

Definition 2.1.6 (n-SuperHyperGraph)**.** (see [10]) Let V_0 be a finite, nonempty base set. Define

$$\mathcal{P}^0(V_0) := V_0, \qquad \mathcal{P}^{k+1}(V_0) := \mathcal{P}\big(\mathcal{P}^k(V_0)\big) \quad (k \in \mathbb{N}).$$

For $n \geq 0$, an *n-SuperHyperGraph on V_0* is a pair

$$\mathrm{SHG}^{(n)} = (V, E)$$

such that

$$V \subseteq \mathcal{P}^n(V_0) \quad \text{and} \quad E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Elements of V are called *n-supervertices*, and elements of E are called *n-superedges* (that is, each n-superedge is a nonempty subset of V).

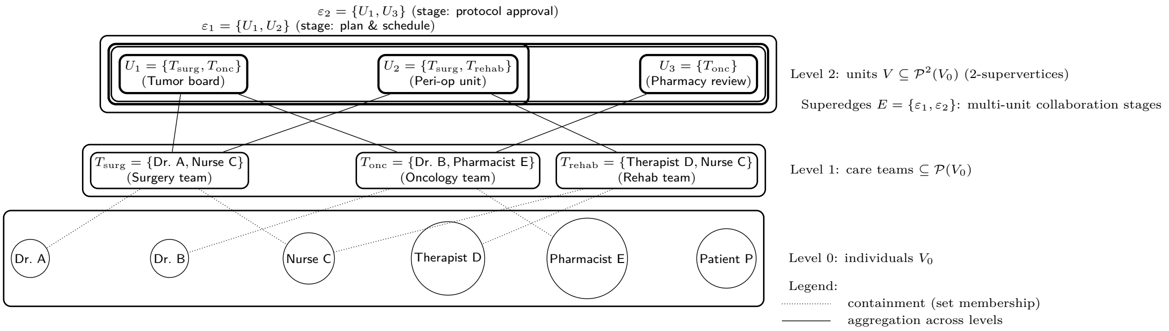


Figure 2.2.: A schematic $n = 2$ SuperHyperGraph model for healthcare coordination: individuals (V_0) form care teams ($\subseteq \mathcal{P}(V_0)$); multidisciplinary units are 2-supervertices ($\subseteq \mathcal{P}^2(V_0)$); superedges encode treatment stages that require joint participation of multiple units.

Example 2.1.7 (Real-life example of an n-SuperHyperGraph)**. Healthcare coordination across nested teams.** Let V_0 be a finite set of individuals in a hospital system, e.g.,

$$V_0 = \{\text{Dr. A}, \text{Dr. B}, \text{Nurse C}, \text{Therapist D}, \text{Pharmacist E}, \text{Patient P}, \dots\}.$$

A *care team* is a subset of V_0, hence an element of $\mathcal{P}(V_0) = \mathcal{P}^1(V_0)$. A *multidisciplinary unit* (for example, a tumor board) is a set of care teams, hence an element of $\mathcal{P}(\mathcal{P}(V_0)) = \mathcal{P}^2(V_0)$. Taking $n = 2$, choose

$$V \subseteq \mathcal{P}^2(V_0)$$

to be a finite collection of such units (each unit is a set of teams), and let

$$E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$$

consist of *treatment stages* that require joint participation of several units (e.g., a stage combining a surgery unit, an oncology unit, and a rehabilitation unit). Then $\mathrm{SHG}^{(2)} = (V, E)$ models higher-order coordination among nested organizational entities: supervertices are sets of teams of individuals, and superedges represent multi-unit collaboration requirements. An example of the graph in this illustrative case is shown in Figure 2.2.

Remark 2.1.8 (Incidence as part of the data: general vs. standard form)**.** Throughout this paper, it is convenient to regard an n-SuperHyperGraph as a triple

$$\mathbb{H}^{(n)} = (\mathbb{V}, \mathbb{E}, \partial),$$

where $\partial : \mathbb{E} \to \mathcal{P}(\mathbb{V}) \setminus \{\varnothing\}$ assigns to each superedge $\varepsilon \in \mathbb{E}$ its (nonempty) incidence set $\partial(\varepsilon) \subseteq \mathbb{V}$. We call this the *general* form.

The *standard* (set-edge) form is the special case in which $\mathbb{E} \subseteq \mathcal{P}(\mathbb{V}) \setminus \{\varnothing\}$ and

$$\partial(\varepsilon) = \varepsilon \qquad (\forall\, \varepsilon \in \mathbb{E}).$$

In particular, any statement or construction (e.g., an incidence graph) defined for triples $(\mathbb{V}, \mathbb{E}, \partial)$ applies to the standard form by taking ∂ to be the identity inclusion.

Chapter 3

Path in SuperHyperGraphs

This chapter explains paths in SuperHyperGraphs and related concepts.

3.1 Paths in Graphs, Hypergraphs, and SuperHyperGraphs

This chapter fixes a standard and widely used notion of a *path* in each of the three settings: graphs, hypergraphs, and n-SuperHyperGraphs. For hypergraphs (and hence for n-SuperHyperGraphs, which are hypergraphs on supervertices), we adopt the classical *Berge* notion of a path, since it applies to arbitrary (non-uniform) hyperedges and is compatible with many distance-based parameters.

Definition 3.1.1 (Walk and path in a graph)**.** Let $G = (V, E)$ be a finite simple undirected graph.

(i) A *walk* in G is a finite sequence of vertices

$$W = (v_0, v_1, \ldots, v_k) \qquad (k \geq 0),$$

such that $\{v_{i-1}, v_i\} \in E$ for every $i = 1, \ldots, k$. The integer k is the *length* of W (the number of edges traversed).

(ii) A *(simple) path* in G is a walk $P = (v_0, v_1, \ldots, v_k)$ in which all vertices are distinct:

$$v_i \neq v_j \quad \text{for all } 0 \leq i < j \leq k.$$

We also say that P is a *path from* v_0 *to* v_k.

(iii) For $u, v \in V$, we write $\mathrm{Path}_G(u, v) \neq \emptyset$ if there exists a path in G from u to v.

Remark 3.1.2. Some authors call any vertex-distinct sequence above a "path" and reserve "simple path" for edge-distinctness. In this book, *path* means vertex-distinct, which is the most common convention in graph theory.

Definition 3.1.3 (Berge walk and Berge path in a hypergraph)**.** (cf. [44, 45]) Let $H = (V, \mathcal{E})$ be a finite hypergraph, where $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$.

(i) A *Berge walk* in H is an alternating sequence

$$\mathsf{W} = (v_0, e_1, v_1, e_2, \dots, e_k, v_k) \qquad (k \geq 0),$$

where $v_0, \dots, v_k \in V$ and $e_1, \dots, e_k \in \mathcal{E}$ satisfy

$$\{v_{i-1}, v_i\} \subseteq e_i \quad \text{for each } i = 1, \dots, k.$$

The integer k is the *length* of W (the number of hyperedges used).

(ii) A *Berge path* in H is a Berge walk $\mathsf{P} = (v_0, e_1, v_1, \dots, e_k, v_k)$ such that all vertices $v_0, \dots, v_k$ are distinct and all hyperedges $e_1, \dots, e_k$ are distinct.

(iii) For $u, v \in V$, we write $\mathrm{Path}_H(u, v) \neq \emptyset$ if there exists a Berge path in H from u to v.

Remark 3.1.4. In the literature on uniform hypergraphs, other notions (e.g., *tight* paths and *loose* paths) are also common. The Berge notion in Definition 3.1.3 is the most general one and works for arbitrary hypergraphs.

Example 3.1.5 (Real-life example of a Berge walk/path in a hypergraph)**. Co-authorship across research groups.** Let V be a set of researchers in a department. For each paper p, define a hyperedge $e_p \subseteq V$ as the set of all coauthors of p. Thus $H = (V, \mathcal{E})$ with $\mathcal{E} = \{e_p : p \text{ is a paper}\}$ is a hypergraph.

Suppose Ayako, Satoshi, Carol, and Dave are distinct researchers and there exist papers p_1, p_2, p_3 such that

$$\{\mathsf{Ayako}, \mathsf{Satoshi}\} \subseteq e_{p_1}, \qquad \{\mathsf{Satoshi}, \mathsf{Carol}\} \subseteq e_{p_2}, \qquad \{\mathsf{Carol}, \mathsf{Dave}\} \subseteq e_{p_3}.$$

Then

$$\mathsf{P} = (\mathsf{Ayako},\ e_{p_1},\ \mathsf{Satoshi},\ e_{p_2},\ \mathsf{Carol},\ e_{p_3},\ \mathsf{Dave})$$

is a Berge path in H from Ayako to Dave: consecutive researchers in the sequence jointly belong to the corresponding paper-hyperedge.

Definition 3.1.6 (Super-Berge walk and Super-Berge path)**.** Let V_0 be a finite nonempty base set, and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0, where $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Thus (V, E) is a hypergraph whose vertices are n-supervertices.

(i) A *Super-Berge walk* in $\mathrm{SHG}^{(n)}$ is an alternating sequence

$$\mathsf{W} = (X_0, \varepsilon_1, X_1, \varepsilon_2, \dots, \varepsilon_k, X_k) \qquad (k \geq 0),$$

where $X_0, \dots, X_k \in V$ and $\varepsilon_1, \dots, \varepsilon_k \in E$ satisfy

$$\{X_{i-1}, X_i\} \subseteq \varepsilon_i \quad \text{for each } i = 1, \dots, k.$$

Its *length* is k.

(ii) A *Super-Berge path* in $\mathrm{SHG}^{(n)}$ is a Super-Berge walk $\mathsf{P} = (X_0, \varepsilon_1, X_1, \ldots, \varepsilon_k, X_k)$ in which all supervertices $X_0, \ldots, X_k$ are distinct and all superedges $\varepsilon_1, \ldots, \varepsilon_k$ are distinct.

(iii) For supervertices $X, Y \in V$, we write $\mathrm{Path}_{\mathrm{SHG}^{(n)}}(X, Y) \neq \emptyset$ if there exists a Super-Berge path from X to Y.

Remark 3.1.7 (Compatibility with flattening)**.** Because each $X \in V \subseteq \mathcal{P}^n(V_0)$ is a nested object, one may also study induced notions of "base-level connectivity" by applying the flattening map Flat_n (Definition 2.1.5) to supervertices, for example by tracking how $\mathrm{Flat}_n(X_i) \subseteq V_0$ evolves along a Super-Berge path. In this book, unless stated otherwise, the term *path in an n-SuperHyperGraph* refers to the Super-Berge notion in Definition 3.1.6.

Example 3.1.8 (Real-life example of a Super-Berge walk/path in an n-SuperHyperGraph)**. Coordinated release across nested software teams.** Let V_0 be a finite set of software engineers. A *team* is a subset of V_0, and a *service group* is a set of teams (e.g., a platform group composed of several feature teams). Taking $n = 2$, each service group is an element of $\mathcal{P}^2(V_0)$.

Let $V \subseteq \mathcal{P}^2(V_0)$ be a collection of service groups, and let each superedge $\varepsilon \in E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$ represent a *release window* that must include several service groups simultaneously (for example, authentication, payments, and logging must be deployed together).

If $X_0, X_1, X_2 \in V$ are distinct service groups and there exist distinct release windows $\varepsilon_1, \varepsilon_2 \in E$ such that

$$\{X_0, X_1\} \subseteq \varepsilon_1 \qquad \text{and} \qquad \{X_1, X_2\} \subseteq \varepsilon_2,$$

then

$$\mathsf{P} = (X_0,\, \varepsilon_1,\, X_1,\, \varepsilon_2,\, X_2)$$

is a Super-Berge path in $\mathrm{SHG}^{(2)} = (V, E)$ from X_0 to X_2. Intuitively, the organization moves from one nested service group to another through coordinated multi-group release events.

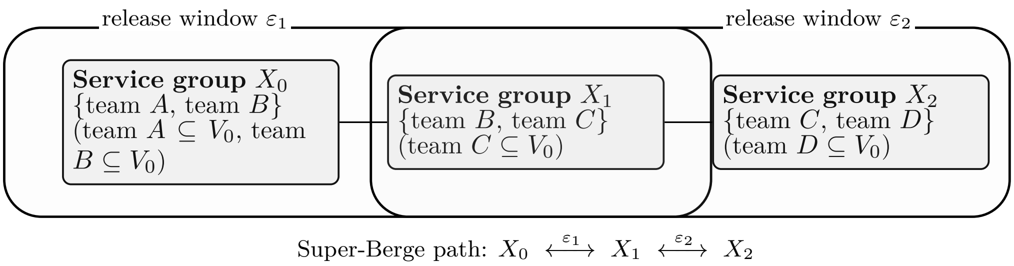


Figure 3.1.: A Super-Berge path in $\mathrm{SHG}^{(2)}$: service groups $X_0, X_1, X_2 \in V \subseteq \mathcal{P}^2(V_0)$ are linked by release-window superedges $\varepsilon_1, \varepsilon_2 \in E$.

3.2 Shortest paths in graphs, hypergraphs, and n-SuperHyperGraphs

In this section we define the notion of a *shortest path* (and the induced *distance*) for weighted graphs, weighted hypergraphs, and weighted n-SuperHyperGraphs. Throughout, all weights are assumed to be nonnegative so that path-length minimization is well-posed.

3.2.1 Shortest paths in graphs

A shortest path in a graph is a minimum-length path between two vertices, where length is the sum of edge weights or edge count [46–49]. Related notions are also known, such as shortest paths in directed graphs, shortest paths in fuzzy graphs, and shortest paths in neutrosophic graphs.

Definition 3.2.1 (Weighted graph and length of a path)**.** Let $G = (V, E)$ be a finite simple undirected graph. A *weight function* on G is a map

$$\mathrm{wt} : E \longrightarrow \mathbb{R}_{\geq 0}.$$

A *(simple) path* P in G is a vertex sequence

$$P = (v_0, v_1, \ldots, v_k) \qquad (k \geq 0),$$

such that $\{v_{i-1}, v_i\} \in E$ for each $i = 1, \ldots, k$, and all $v_0, \ldots, v_k$ are distinct. The wt-*length* of P is

$$\ell_{\mathrm{wt}}(P) \;:=\; \sum_{i=1}^{k} \mathrm{wt}\big(\{v_{i-1}, v_i\}\big).$$

Definition 3.2.2 (Shortest path and graph distance)**.** (cf. [50]) Let (G, wt) be a weighted graph as in Definition 3.2.1. For $u, v \in V$, define the *graph distance* from u to v by

$$\mathrm{dist}_G(u, v) \;:=\; \min\big\{\ell_{\mathrm{wt}}(P) \;:\; P \text{ is a path in } G \text{ from } u \text{ to } v\big\},$$

with the convention $\mathrm{dist}_G(u, v) = +\infty$ if no such path exists. A *shortest u–v path* is any path P^* from u to v satisfying

$$\ell_{\mathrm{wt}}(P^*) = \mathrm{dist}_G(u, v).$$

Remark 3.2.3. If $\mathrm{wt}(e) \equiv 1$ for all $e \in E$, then $\ell_{\mathrm{wt}}(P) = k$ equals the number of edges of P, and $\mathrm{dist}_G(u, v)$ is the minimum number of edges among all u–v paths.

3.2.2 Shortest Berge paths in hypergraphs

A shortest Berge path in a hypergraph is a minimum-length path between two vertices, where length is the sum of edge weights or edge count.

Definition 3.2.4 (Weighted hypergraph and Berge-path length)**.** Let $H = (V, E)$ be a finite hypergraph, where $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. A *weight function* on H is a map

$$\mathrm{wt} : E \longrightarrow \mathbb{R}_{\geq 0}.$$

A *Berge path* in H is an alternating sequence

$$\mathsf{P} = (v_0, e_1, v_1, e_2, \ldots, e_k, v_k) \qquad (k \geq 0),$$

where $v_0, \ldots, v_k \in V$ are distinct vertices, $e_1, \ldots, e_k \in E$ are distinct hyperedges, and

$$\{v_{i-1}, v_i\} \subseteq e_i \quad \text{for each } i = 1, \ldots, k.$$

The wt-*length* of P is

$$\ell_{\mathrm{wt}}(\mathsf{P}) \;:=\; \sum_{i=1}^{k} \mathrm{wt}(e_i).$$

Definition 3.2.5 (Shortest Berge path and hypergraph distance)**.** Let (H, wt) be a weighted hypergraph as in Definition 3.2.4. For $u, v \in V$, define the *Berge distance* from u to v by

$$\mathrm{dist}_H(u, v) \ := \ \min\{\ell_{\mathrm{wt}}(\mathsf{P}) \ : \ \mathsf{P} \text{ is a Berge path in } H \text{ from } u \text{ to } v\},$$

with the convention $\mathrm{dist}_H(u, v) = +\infty$ if no Berge path from u to v exists. A *shortest Berge u–v path* is any Berge path P^* from u to v such that

$$\ell_{\mathrm{wt}}(\mathsf{P}^*) = \mathrm{dist}_H(u, v).$$

Remark 3.2.6. There are other hypergraph path notions (e.g., tight/loose paths for uniform hypergraphs, or paths in the primal/2-section graph). Definitions 3.2.4–3.2.5 use the Berge notion, which applies to arbitrary (non-uniform) hypergraphs and is widely used to define distance-like parameters.

3.2.3 Shortest Super-Berge paths in n-SuperHyperGraphs

A shortest Super-Berge path in an n-superhypergraph is a minimum-length alternating supervertex–superedge path, minimizing the total superedge weights or the number of superedges.

Definition 3.2.7 (Weighted n-SuperHyperGraph)**.** Let V_0 be a finite nonempty base set and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0, with $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. A *weight function* on $\mathrm{SHG}^{(n)}$ is a map

$$\mathrm{wt} : E \longrightarrow \mathbb{R}_{\geq 0}.$$

Definition 3.2.8 (Super-Berge path length, shortest path, and SuperHyperGraph distance)**.** Let $(\mathrm{SHG}^{(n)}, \mathrm{wt})$ be a weighted n-SuperHyperGraph as in Definition 3.2.7. A *Super-Berge path* from X to Y (where $X, Y \in V$) is an alternating sequence

$$\mathsf{P} = (X_0, \varepsilon_1, X_1, \varepsilon_2, \ldots, \varepsilon_k, X_k) \qquad (k \geq 0),$$

with $X_0 = X$, $X_k = Y$, where $X_0, \ldots, X_k \in V$ are distinct supervertices, $\varepsilon_1, \ldots, \varepsilon_k \in E$ are distinct superedges, and

$$\{X_{i-1}, X_i\} \subseteq \varepsilon_i \quad \text{for each } i = 1, \ldots, k.$$

Its wt-*length* is

$$\ell_{\mathrm{wt}}(\mathsf{P}) \ := \ \sum_{i=1}^{k} \mathrm{wt}(\varepsilon_i).$$

For $X, Y \in V$, define the *SuperHyperGraph distance* by

$$\mathrm{dist}_{\mathrm{SHG}^{(n)}}(X, Y) \ := \ \min\{\ell_{\mathrm{wt}}(\mathsf{P}) \ : \ \mathsf{P} \text{ is a Super-Berge path in } \mathrm{SHG}^{(n)} \text{ from } X \text{ to } Y\},$$

with the convention $\mathrm{dist}_{\mathrm{SHG}^{(n)}}(X, Y) = +\infty$ if no Super-Berge path from X to Y exists.

A *shortest Super-Berge X–Y path* is any Super-Berge path P^* from X to Y such that

$$\ell_{\mathrm{wt}}(\mathsf{P}^*) = \mathrm{dist}_{\mathrm{SHG}^{(n)}}(X, Y).$$

Remark 3.2.9. If $\mathrm{wt}(\varepsilon) \equiv 1$ for all $\varepsilon \in E$, then $\ell_{\mathrm{wt}}(\mathsf{P}) = k$ equals the number of superedges used, and $\mathrm{dist}_{\mathrm{SHG}^{(n)}}(X, Y)$ is the minimum number of superedges among all Super-Berge paths from X to Y.

Example 3.2.10 (Real-life example of a shortest Super-Berge path)**. Minimum-cost coordination between nested organizations.** Let V_0 be a finite set of employees in a company. A *team* is a subset of V_0, and a *department program* is a set of teams (e.g., a program that groups several teams working on related features). Take $n = 2$ so that such programs are elements of $\mathcal{P}^2(V_0)$. Let $V \subseteq \mathcal{P}^2(V_0)$ be a finite collection of programs (supervertices).

Suppose superedges represent *coordination meetings* that can jointly involve several programs, and let the weight $\mathrm{wt}(\varepsilon)$ be the estimated cost (e.g., staff-hours) of running meeting ε. Form $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$ and $\mathrm{wt} : E \to \mathbb{R}_{\geq 0}$.

Assume we have three programs $X, Y, Z \in V$ and two meetings $\varepsilon_1, \varepsilon_2 \in E$ with

$$\{X, Y\} \subseteq \varepsilon_1, \quad \mathrm{wt}(\varepsilon_1) = 2, \qquad \{Y, Z\} \subseteq \varepsilon_2, \quad \mathrm{wt}(\varepsilon_2) = 3,$$

so $(X, \varepsilon_1, Y, \varepsilon_2, Z)$ is a Super-Berge path from X to Z of total weight $2 + 3 = 5$. Also assume there is a direct meeting $\varepsilon_3 \in E$ with

$$\{X, Z\} \subseteq \varepsilon_3, \quad \mathrm{wt}(\varepsilon_3) = 4.$$

Then the one-step path

$$\mathsf{P}^* = (X, \varepsilon_3, Z)$$

is a *shortest* Super-Berge path from X to Z, since it achieves the minimum total coordination cost $\mathrm{dist}_{\mathrm{SHG}^{(2)}}(X, Z) = 4$ among all Super-Berge paths connecting X and Z.

3.3 Longest paths in graphs, hypergraphs, and n-SuperHyperGraphs

In this section we define the notion of a *longest path* (and the associated *maximum path length*) for weighted graphs, weighted hypergraphs, and weighted n-SuperHyperGraphs [51–54]. In all three settings we restrict attention to *simple* paths (no repeated vertices, and in the hypergraph settings also no repeated hyperedges), so that the family of feasible paths is finite and a maximum is well-defined.

3.3.1 Longest paths in graphs

A longest path in a graph is a simple path with maximum length, measured by number of edges or total edge weight among all simple paths.

Definition 3.3.1 (Weighted path length in a graph)**.** Let $G = (V, E)$ be a finite simple undirected graph, and let $\mathrm{wt} : E \to \mathbb{R}$ be a weight function.

A *(simple) path* in G is a vertex sequence

$$P = (v_0, v_1, \ldots, v_k) \qquad (k \geq 0),$$

such that $\{v_{i-1}, v_i\} \in E$ for each $i = 1, \dots, k$ and all vertices $v_0, \dots, v_k$ are distinct. The wt-*length* of P is

$$\ell_{\mathrm{wt}}(P) := \sum_{i=1}^{k} \mathrm{wt}(\{v_{i-1}, v_i\}).$$

In the *unweighted* case ($\mathrm{wt} \equiv 1$), one has $\ell_{\mathrm{wt}}(P) = k$, the number of edges of P.

Definition 3.3.2 (Longest u–v path and longest-path value)**.** Let (G, wt) be as in Definition 3.3.1. For $u, v \in V$, define the *longest-path value* between u and v by

$$\mathrm{dist}_G^{\max}(u, v) := \max\{\ell_{\mathrm{wt}}(P) : P \text{ is a simple path in } G \text{ from } u \text{ to } v\},$$

with the convention $\mathrm{dist}_G^{\max}(u, v) = -\infty$ if there is no simple path from u to v. A *longest u–v path* is any u–v path P^* such that

$$\ell_{\mathrm{wt}}(P^*) = \mathrm{dist}_G^{\max}(u, v).$$

The *(global) longest-path value* of G is

$$\mathrm{dist}^{\max}(G) := \max\{\ell_{\mathrm{wt}}(P) : P \text{ is a simple path in } G\} = \max_{u,v \in V} \mathrm{dist}_G^{\max}(u, v),$$

and any simple path attaining $\mathrm{dist}^{\max}(G)$ is called a *longest path of* G.

Remark 3.3.3. The optimization problem of finding a longest simple path is computationally difficult in general graphs (NP-hard), but the definition above is purely structural and does not depend on algorithmic solvability.

3.3.2 Longest Berge paths in hypergraphs

A longest path in a hypergraph is a longest Berge path: an alternating vertex–hyperedge sequence with no repeated vertices or hyperedges, maximizing edge-count or hyperedge-weight sum.

Definition 3.3.4 (Weighted Berge-path length in a hypergraph)**.** Let $H = (V, \mathcal{E})$ be a finite hypergraph, where $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$, and let $\mathrm{wt} : \mathcal{E} \to \mathbb{R}$ be a weight function.

A *Berge path* in H is an alternating sequence

$$\mathsf{P} = (v_0, e_1, v_1, e_2, \dots, e_k, v_k) \qquad (k \geq 0),$$

such that:

- $v_0, \dots, v_k \in V$ are pairwise distinct vertices;
- $e_1, \dots, e_k \in \mathcal{E}$ are pairwise distinct hyperedges;

- $\{v_{i-1}, v_i\} \subseteq e_i$ for each $i = 1, \ldots, k$.

The wt-*length* of P is

$$\ell_{\mathrm{wt}}(\mathsf{P}) \;:=\; \sum_{i=1}^{k} \mathrm{wt}(e_i).$$

In the unweighted case ($\mathrm{wt} \equiv 1$), $\ell_{\mathrm{wt}}(\mathsf{P}) = k$, the number of hyperedges used.

Definition 3.3.5 (Longest Berge u–v path and hypergraph longest-path value)**.** Let (H, wt) be as in Definition 3.3.4. For $u, v \in V$, define the *Berge longest-path value* by

$$\mathrm{dist}_H^{\max}(u, v) \;:=\; \max\bigl\{\ell_{\mathrm{wt}}(\mathsf{P}) \;:\; \mathsf{P} \text{ is a Berge path in } H \text{ from } u \text{ to } v\bigr\},$$

with the convention $\mathrm{dist}_H^{\max}(u, v) = -\infty$ if no Berge path from u to v exists. A *longest Berge u–v path* is any Berge path P^* from u to v satisfying

$$\ell_{\mathrm{wt}}(\mathsf{P}^*) = \mathrm{dist}_H^{\max}(u, v).$$

The *(global) Berge longest-path value* of H is

$$\mathrm{dist}^{\max}(H) \;:=\; \max\bigl\{\ell_{\mathrm{wt}}(\mathsf{P}) \;:\; \mathsf{P} \text{ is a Berge path in } H\bigr\} \;=\; \max_{u,v \in V} \mathrm{dist}_H^{\max}(u, v),$$

and any Berge path attaining $\mathrm{dist}^{\max}(H)$ is called a *longest Berge path of H*.

Remark 3.3.6. Other hypergraph path notions exist (tight/loose, primal-graph paths, etc.). Definitions 3.3.4–3.3.5 use Berge paths because they apply to arbitrary hypergraphs (no uniformity assumptions) and are compatible with many distance-based invariants.

3.3.3 Longest Super-Berge paths in n-SuperHyperGraphs

A longest path in a superhypergraph is a longest Super-Berge path: an alternating supervertex–superedge sequence without repeats, maximizing the number or total weight of superedges used.

Definition 3.3.7 (Weighted n-SuperHyperGraph)**.** [55] Let V_0 be a finite nonempty base set and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0, with $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. A *weight function* on $\mathrm{SHG}^{(n)}$ is a map

$$\mathrm{wt} : E \to \mathbb{R}.$$

Definition 3.3.8 (Longest Super-Berge path and SuperHyperGraph longest-path value)**.** Let $(\mathrm{SHG}^{(n)}, \mathrm{wt})$ be as in Definition 3.3.7. A *Super-Berge path* from X to Y (where $X, Y \in V$) is an alternating sequence

$$\mathsf{P} = (X_0, \varepsilon_1, X_1, \varepsilon_2, \ldots, \varepsilon_k, X_k) \qquad (k \geq 0),$$

such that:

- $X_0, \ldots, X_k \in V$ are pairwise distinct n-supervertices;
- $\varepsilon_1, \ldots, \varepsilon_k \in E$ are pairwise distinct n-superedges;
- $\{X_{i-1}, X_i\} \subseteq \varepsilon_i$ for each $i = 1, \ldots, k$.

Its wt-*length* is

$$\ell_{\mathrm{wt}}(\mathsf{P}) \;:=\; \sum_{i=1}^{k} \mathrm{wt}(\varepsilon_i).$$

For $X, Y \in V$, define the *SuperHyperGraph longest-path value* by

$$\mathrm{dist}^{\max}_{\mathrm{SHG}^{(n)}}(X, Y) \;:=\; \max\bigl\{\ell_{\mathrm{wt}}(\mathsf{P}) \;:\; \mathsf{P} \text{ is a Super-Berge path in } \mathrm{SHG}^{(n)} \text{ from } X \text{ to } Y\bigr\},$$

with the convention $\mathrm{dist}^{\max}_{\mathrm{SHG}^{(n)}}(X, Y) = -\infty$ if no Super-Berge path from X to Y exists. A *longest Super-Berge X–Y path* is any Super-Berge path P^* from X to Y such that

$$\ell_{\mathrm{wt}}(\mathsf{P}^*) = \mathrm{dist}^{\max}_{\mathrm{SHG}^{(n)}}(X, Y).$$

The *(global) SuperHyperGraph longest-path value* of $\mathrm{SHG}^{(n)}$ is

$$\mathrm{dist}^{\max}\bigl(\mathrm{SHG}^{(n)}\bigr) \;:=\; \max\bigl\{\ell_{\mathrm{wt}}(\mathsf{P}) \;:\; \mathsf{P} \text{ is a Super-Berge path in } \mathrm{SHG}^{(n)}\bigr\} \;=\; \max_{X,Y \in V} \mathrm{dist}^{\max}_{\mathrm{SHG}^{(n)}}(X, Y),$$

and any Super-Berge path attaining $\mathrm{dist}^{\max}\bigl(\mathrm{SHG}^{(n)}\bigr)$ is called a *longest Super-Berge path.*

Remark 3.3.9. Since (V, E) is a hypergraph whose vertices are nested objects (n-supervertices), Definition 3.3.8 is the direct Berge-type extension of the longest-path notion. If desired, one may additionally study *base-level* effects along P via flattening ($\mathrm{Flat}_n(X_i) \subseteq V_0$), but the longest-path notion above is defined intrinsically at the supervertex level.

Example 3.3.10 (Real-life example of a longest Super-Berge path)**. Maximizing distinct cross-team coordination steps in a large software rollout.** Let V_0 be a finite set of engineering groups in a company. A *micro-team* is a subset of V_0, and a *service organization* is a set of micro-teams. Take $n = 2$, so each service organization is an element of $\mathcal{P}^2(V_0)$. Let $V \subseteq \mathcal{P}^2(V_0)$ be a set of service organizations (supervertices).

Assume superedges represent *cross-organization rollout checkpoints* for a major release. Each checkpoint $\varepsilon \in E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$ contains the set of organizations that must synchronize at that checkpoint, and $\mathrm{wt}(\varepsilon) \ge 0$ is the estimated effort (e.g., total staff-hours of meetings and reviews) required to complete that checkpoint.

Consider five distinct organizations $X_0, X_1, X_2, X_3, X_4 \in V$, and four distinct checkpoints $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4 \in E$ such that

$$\{X_0, X_1\} \subseteq \varepsilon_1, \quad \{X_1, X_2\} \subseteq \varepsilon_2, \quad \{X_2, X_3\} \subseteq \varepsilon_3, \quad \{X_3, X_4\} \subseteq \varepsilon_4,$$

with weights

$$\mathrm{wt}(\varepsilon_1) = 3, \quad \mathrm{wt}(\varepsilon_2) = 2, \quad \mathrm{wt}(\varepsilon_3) = 4, \quad \mathrm{wt}(\varepsilon_4) = 1.$$

Then

$$\mathsf{P} = (X_0, \varepsilon_1, X_1, \varepsilon_2, X_2, \varepsilon_3, X_3, \varepsilon_4, X_4)$$

is a Super-Berge path from X_0 to X_4 of total weight

$$\ell_{\mathrm{wt}}(\mathsf{P}) = 3 + 2 + 4 + 1 = 10.$$

Assume further that *no* Super-Berge path in $\mathrm{SHG}^{(2)}$ can use more than four distinct checkpoints without repeating a supervertex or a checkpoint (for example, because the remaining checkpoints all involve one of $X_0, \ldots, X_4$ again). Then P is a *longest* Super-Berge path in this rollout model: it maximizes the number of distinct cross-organization checkpoints (and, among such paths, it can also be compared by total effort ℓ_{wt}).

Chapter 4

SuperHyperTrees

In this chapter, we investigate SuperHyperTrees and related concepts.

4.1 Basic Concepts in SuperHyperTrees

SuperHyperTrees are connected SuperHyperGraphs whose incidence structure is acyclic across hierarchical vertices and superedges, generalizing trees by allowing nested, set-valued vertices and edges at all. This section describes SuperHyperTrees.

4.1.1 Trees in graphs

A tree in graphs is a connected, acyclic undirected graph; equivalently, it has a unique simple path between any two vertices [56–58].

Definition 4.1.1 (Tree)**.** A *tree* is a finite simple undirected graph $T = (V, E)$ that is *connected* and *acyclic*, i.e., T contains no (simple) cycle.

Equivalently, T is a tree if and only if for every pair of distinct vertices $u, v \in V$ there exists a *unique* simple u–v path in T.

Remark 4.1.2. For a finite simple graph $T = (V, E)$, the following are equivalent and are often used as definitions: (i) T is connected and acyclic; (ii) T is connected and $|E| = |V| - 1$; (iii) T is acyclic and $|E| = |V| - 1$.

4.1.2 Hypertrees in hypergraphs

A hypertree is a hypergraph whose incidence graph is a tree, so it is Berge-acyclic and connected under incidence-based adjacency [59, 60].

Definition 4.1.3 (Incidence graph of a hypergraph)**.** Let $H = (V, E)$ be a finite hypergraph, where $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. The *incidence graph* (or *Levi graph*) of H is the bipartite simple graph

$$\mathrm{Inc}(H) \;=\; \bigl(V \;\sqcup\; E,\; F\bigr),$$

whose bipartition is V and E, and whose edge set is

$$F \;=\; \{\; \{v, e\} \;:\; v \in V,\; e \in E,\; v \in e \;\}.$$

Definition 4.1.4 (Berge cycle)**.** Let $H = (V, E)$ be a hypergraph. A *Berge cycle* of length $k \geq 2$ is an alternating sequence

$$\mathsf{C} = (v_0, e_1, v_1, e_2, \ldots, e_k, v_k)$$

such that:

- $v_0, \ldots, v_{k-1} \in V$ are pairwise distinct and $v_k = v_0$;
- $e_1, \ldots, e_k \in E$ are pairwise distinct;
- $\{v_{i-1}, v_i\} \subseteq e_i$ for each $i = 1, \ldots, k$.

Definition 4.1.5 (HyperTree)**.** [60] A finite hypergraph $H = (V, E)$ is called a *HyperTree* (also called a *Berge-tree* or *incidence-tree*) if its incidence graph $\mathrm{Inc}(H)$ is a tree.

Equivalently, H is a HyperTree if and only if $\mathrm{Inc}(H)$ is connected and H contains no Berge cycle (Definition 4.1.4).

Remark 4.1.6. In hypergraph theory there are several non-equivalent acyclicity notions (e.g., α-, β-, γ-acyclicity) and corresponding "hypertree" concepts. Definition 4.1.5 adopts the incidence-graph (Berge) notion, which is conceptually closest to ordinary trees and is well suited for path/cycle-style generalizations.

4.2 SuperHyperTrees in n-SuperHyperGraphs

A SuperHyperTree is an n-SuperHyperGraph whose incidence graph is a tree, so supervertices and superedges form a connected, cycle-free hierarchy [18, 61, 62].

Definition 4.2.1 (Incidence graph of an n-SuperHyperGraph)**.** Let V_0 be a finite nonempty base set and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0, where $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. The *incidence graph* of $\mathrm{SHG}^{(n)}$ is the bipartite simple graph

$$\mathrm{Inc}\big(\mathrm{SHG}^{(n)}\big) \;=\; \big(V \;\sqcup\; E,\; F\big), \qquad F \;=\; \big\{\; \{X, \varepsilon\} \;:\; X \in V,\; \varepsilon \in E,\; X \in \varepsilon \;\big\}.$$

Example 4.2.2 (Real-life example of an incidence graph of an n-SuperHyperGraph)**. Services and coordinated release windows.** Let V_0 be a finite set of software repositories:

$$V_0 = \{\mathsf{auth\text{-}repo}, \mathsf{payments\text{-}repo}, \mathsf{logging\text{-}repo}, \mathsf{search\text{-}repo}\}.$$

Take $n = 1$. Define supervertices (modules) as subsets of repositories:

$$X_1 := \{\mathsf{auth\text{-}repo}, \mathsf{logging\text{-}repo}\}, \quad X_2 := \{\mathsf{payments\text{-}repo}\}, \quad X_3 := \{\mathsf{search\text{-}repo}, \mathsf{logging\text{-}repo}\},$$

and set $V := \{X_1, X_2, X_3\} \subseteq \mathcal{P}(V_0)$. Let superedges represent coordinated release windows:

$$\varepsilon_A := \{X_1, X_2\}, \qquad \varepsilon_B := \{X_2, X_3\}, \qquad E := \{\varepsilon_A, \varepsilon_B\}.$$

Then the incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(1)})$ has bipartition $V \sqcup E$, and edges

$$\{X_1, \varepsilon_A\},\; \{X_2, \varepsilon_A\},\; \{X_2, \varepsilon_B\},\; \{X_3, \varepsilon_B\},$$

encoding which service-modules X_i participate in which release windows $\varepsilon_A, \varepsilon_B$.

Definition 4.2.3 (Super-Berge cycle)**.** Let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph. A *Super-Berge cycle* of length $k \geq 2$ is an alternating sequence

$$\mathsf{C} = (X_0, \varepsilon_1, X_1, \varepsilon_2, \ldots, \varepsilon_k, X_k)$$

such that:

- $X_0, \ldots, X_{k-1} \in V$ are pairwise distinct and $X_k = X_0$;
- $\varepsilon_1, \ldots, \varepsilon_k \in E$ are pairwise distinct;
- $\{X_{i-1}, X_i\} \subseteq \varepsilon_i$ for each $i = 1, \ldots, k$.

Example 4.2.4 (Real-life example of a Super-Berge cycle)**. Circular dependency of coordination among three organizations.** Let V be a set of three distinct organizations in a multi-stakeholder project:

$$X_0 := \mathsf{Ops}, \qquad X_1 := \mathsf{Security}, \qquad X_2 := \mathsf{Compliance}.$$

Let superedges represent joint approval meetings:

$$\varepsilon_1 := \{X_0, X_1\} \text{ (Ops–Security approval)},$$

$$\varepsilon_2 := \{X_1, X_2\} \text{ (Security–Compliance approval)},$$

$$\varepsilon_3 := \{X_2, X_0\} \text{ (Compliance–Ops approval)}.$$

Then

$$\mathsf{C} = (X_0, \varepsilon_1, X_1, \varepsilon_2, X_2, \varepsilon_3, X_0)$$

is a Super-Berge cycle of length 3: each consecutive pair of organizations appears together in the corresponding meeting, and all supervertices and superedges in the cycle are distinct except that the start and end supervertex coincide.

Definition 4.2.5 (n-SuperHyperTree)**.** [63] An n-SuperHyperGraph $\mathrm{SHG}^{(n)} = (V, E)$ is called an *n-SuperHyperTree* (or simply a *SuperHyperTree* when n is understood) if its incidence graph $\mathrm{Inc}\big(\mathrm{SHG}^{(n)}\big)$ is a tree.

Equivalently, $\mathrm{SHG}^{(n)}$ is an n-SuperHyperTree if and only if $\mathrm{Inc}\big(\mathrm{SHG}^{(n)}\big)$ is connected and $\mathrm{SHG}^{(n)}$ contains no Super-Berge cycle (Definition 4.2.3).

Remark 4.2.6. Definition 4.2.5 is intrinsic at the supervertex level. When needed, one may additionally study base-level effects of a SuperHyperTree by applying flattening (Flat_n) to supervertices $X \in V \subseteq \mathcal{P}^n(V_0)$, but such base-level considerations are not required to define the SuperHyperTree structure itself.

Example 4.2.7 (2-SuperHyperTree (a simple path-shaped incidence tree))**.** Let the base set be

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

Define three level-1 teams (subsets of V_0):

$$T_1 = \{a, b\}, \qquad T_2 = \{b, c\}, \qquad T_3 = \{c, d\}.$$

A level-2 supervertex (a "service group") is a set of teams, hence an element of $\mathcal{P}(\mathcal{P}(V_0)) = \mathcal{P}^2(V_0)$. Put

$$X_0 = \{T_1, T_2\}, \qquad X_1 = \{T_2\}, \qquad X_2 = \{T_2, T_3\},$$

and set

$$V = \{X_0, X_1, X_2\} \subseteq \mathcal{P}^2(V_0).$$

Now define two superedges (release windows) by

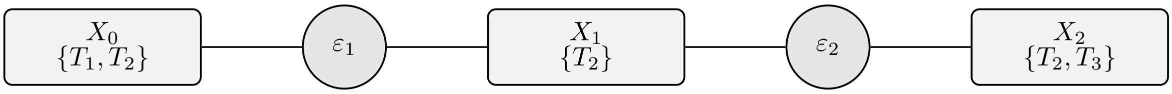
$$\varepsilon_1 = \{X_0, X_1\}, \qquad \varepsilon_2 = \{X_1, X_2\}, \qquad E = \{\varepsilon_1, \varepsilon_2\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Then $\mathrm{SHG}^{(2)} = (V, E)$ is a 2-SuperHyperGraph.

Verification. The incidence graph $\mathrm{Inc}\big(\mathrm{SHG}^{(2)}\big)$ has vertex set $V \cup E$ and edges $X - \varepsilon$ whenever $X \in \varepsilon$. Figure 4.1 shows that $\mathrm{Inc}\big(\mathrm{SHG}^{(2)}\big)$ is the path

$$X_0 — \varepsilon_1 — X_1 — \varepsilon_2 — X_2,$$

and therefore a tree. Consequently, $\mathrm{SHG}^{(2)}$ is a 2-SuperHyperTree in the sense of Definition 4.2.5; equivalently, it is connected and contains no Super-Berge cycle.



Rectangles: supervertices $X_i \in V$. Circles: superedges $\varepsilon_j \in E$.

Figure 4.1.: The incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(2)})$ for Example 4.2.7. It is a path, hence a tree.

We recall that for an n-SuperHyperGraph $\mathrm{SHG}^{(n)} = (V, E)$ (with $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$), its *incidence graph* $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ is the (simple) bipartite graph with bipartition (V, E) and

$$V\big(\mathrm{Inc}(\mathrm{SHG}^{(n)})\big) = V \sqcup E, \qquad \{\, X, \varepsilon \,\} \in E\big(\mathrm{Inc}(\mathrm{SHG}^{(n)})\big) \iff X \in \varepsilon,$$

for $X \in V$ and $\varepsilon \in E$. An n-SuperHyperGraph is an n-SuperHyperTree if $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ is a tree (Definition 4.2.5).

Lemma 4.2.8 (Bipartite structure and incidence count)**.** *Let* $\mathrm{SHG}^{(n)} = (V, E)$ *be an n-SuperHyperGraph and let I denote its incidence set*

$$I := \{(X, \varepsilon) \in V \times E \mid X \in \varepsilon\}.$$

Then $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ *is bipartite with parts (V, E), and*

$$\big|E(\mathrm{Inc}(\mathrm{SHG}^{(n)}))\big| = \sum_{\varepsilon \in E} |\varepsilon| = |I|.$$

If moreover $\mathrm{SHG}^{(n)}$ *is finite and is an n-SuperHyperTree, then*

$$\sum_{\varepsilon \in E} |\varepsilon| = |V| + |E| - 1.$$

Proof. By construction, every edge of $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ joins a vertex in V to a vertex in E, hence the graph is bipartite with bipartition (V, E). Each incidence pair (X, ε) with $X \in \varepsilon$ corresponds to exactly one edge $\{X, \varepsilon\}$ in the incidence graph, so $|E(\mathrm{Inc}(\mathrm{SHG}^{(n)}))| = |I| = \sum_{\varepsilon \in E} |\varepsilon|$.

If $\mathrm{SHG}^{(n)}$ is finite and $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ is a tree, then the standard tree identity gives $|E(\mathrm{Inc}(\mathrm{SHG}^{(n)}))| = |V(\mathrm{Inc}(\mathrm{SHG}^{(n)}))| - 1 = (|V| + |E|) - 1$. Combining this with the first part yields $\sum_{\varepsilon \in E} |\varepsilon| = |V| + |E| - 1$. □

Lemma 4.2.9 (Cycles in $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ and Super-Berge cycles)**.** *An n-SuperHyperGraph* $\mathrm{SHG}^{(n)} = (V, E)$ *contains a Super-Berge cycle (in the sense of Definition 4.2.3) if and only if its incidence graph* $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ *contains a graph cycle.*

Proof. A Super-Berge cycle is, by definition, an alternating cyclic sequence

$$X_0,\ \varepsilon_1,\ X_1,\ \varepsilon_2,\ \ldots,\ \varepsilon_k,\ X_k (= X_0),$$

where $X_{i-1}, X_i \in \varepsilon_i$ for each i, and where the involved supervertices and superedges are distinct except for the closure $X_k = X_0$. This is precisely a cycle in the bipartite graph with vertex set $V \sqcup E$ and edges $\{X, \varepsilon\}$ whenever $X \in \varepsilon$, namely $\mathrm{Inc}(\mathrm{SHG}^{(n)})$.

Conversely, any cycle in $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ alternates between V and E, hence reads as an alternating sequence of supervertices and superedges satisfying membership at each step, i.e., a Super-Berge cycle. □

Lemma 4.2.10 (No multiple intersections in a SuperHyperTree)**.** *Let* $\mathrm{SHG}^{(n)} = (V, E)$ *be an n-SuperHyperTree. Then:*

1. *For any distinct superedges* $\varepsilon, \delta \in E$*, one has* $|\varepsilon \cap \delta| \leq 1$*.*

2. *For any distinct supervertices* $X, Y \in V$*, there exists at most one superedge* $\varepsilon \in E$ *such that* $\{X, Y\} \subseteq \varepsilon$*.*

Proof. (i) Suppose $\varepsilon \neq \delta$ and $\varepsilon \cap \delta$ contains two distinct supervertices $X \neq Y$. Then in $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ we have the 4-cycle

$$X - \varepsilon - Y - \delta - X,$$

contradicting that $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ is a tree. Hence $|\varepsilon \cap \delta| \leq 1$.

(ii) If there were two distinct superedges $\varepsilon \neq \delta$ with $\{X, Y\} \subseteq \varepsilon \cap \delta$, then $\varepsilon \cap \delta$ would contain X and Y, violating (i). □

Lemma 4.2.11 (Unique Super-Berge path between supervertices)**.** *Let* $\mathrm{SHG}^{(n)} = (V, E)$ *be an n-SuperHyperTree and let* $X, Y \in V$ *be distinct. Then there exists a* unique *simple Super-Berge path from* X *to* Y*. Equivalently, there exists a unique simple path from* X *to* Y *in* $\mathrm{Inc}(\mathrm{SHG}^{(n)})$*, and reading it as an alternating* V*-*E*-*V*-*$\cdots$ *sequence yields the unique Super-Berge path.*

Proof. Since $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ is a tree, it is connected and contains a unique simple path between any two distinct vertices. In particular, there is a unique simple graph path between X and Y in $\mathrm{Inc}(\mathrm{SHG}^{(n)})$. Because $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ is bipartite with parts (V, E), that path alternates between supervertices and superedges:

$$X = X_0,\ \varepsilon_1,\ X_1,\ \varepsilon_2,\ \ldots,\ \varepsilon_k,\ X_k = Y,$$

with $X_{i-1}, X_i \in \varepsilon_i$ for each i. This alternating sequence is exactly a simple Super-Berge path from X to Y. Uniqueness follows from the uniqueness of the simple path in the tree $\mathrm{Inc}(\mathrm{SHG}^{(n)})$. □

Lemma 4.2.12 (Leaves and pendant objects)**.** *Let* $\mathrm{SHG}^{(n)} = (V, E)$ *be a finite n-SuperHyperTree with* $|V| + |E| \geq 2$*. Then* $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ *has at least two leaves. Consequently, at least one of the following holds:*

1. *there exists a* pendant supervertex $X \in V$ *incident with exactly one superedge, i.e.,* $|\{\varepsilon \in E : X \in \varepsilon\}| = 1$*;*

2. *there exists a* pendant superedge $\varepsilon \in E$ *containing exactly one supervertex, i.e.,* $|\varepsilon| = 1$*.*

Proof. A finite tree with at least two vertices has at least two leaves. Apply this to $\mathrm{Inc}(\mathrm{SHG}^{(n)})$, whose vertex set is $V \sqcup E$. A leaf in the incidence graph lies either in V or in E. If the leaf is $X \in V$, then X is adjacent to exactly one $\varepsilon \in E$, meaning X belongs to exactly one superedge. If the leaf is $\varepsilon \in E$, then ε is adjacent to exactly one supervertex $X \in V$, meaning $|\varepsilon| = 1$. □

Lemma 4.2.13 (Pruning a pendant supervertex)**.** *Let* $\mathrm{SHG}^{(n)} = (V, E)$ *be an* n*-SuperHyperTree and let* $X \in V$ *be a pendant supervertex. Write* $\{\varepsilon\} = \{\eta \in E : X \in \eta\}$ *for its unique incident superedge. Define a new pair* (V', E') *by*

$$V' := V \setminus \{X\}, \qquad E' := \big(E \setminus \{\varepsilon\}\big) \cup \begin{cases} \{\varepsilon \setminus \{X\}\}, & \text{if } |\varepsilon| \ge 2, \\ \emptyset, & \text{if } |\varepsilon| = 1. \end{cases}$$

Then $\mathrm{SHG}'^{(n)} := (V', E')$ *is again an* n*-SuperHyperTree (provided* $V' \neq \emptyset$*).*

Proof. In $\mathrm{Inc}(\mathrm{SHG}^{(n)})$, the pendant condition says that X is a leaf adjacent only to ε. Removing the leaf X and its unique incident edge from a tree yields a (smaller) tree on the remaining vertices.

If $|\varepsilon| \ge 2$, then $\varepsilon \setminus \{X\} \neq \emptyset$, and replacing ε by $\varepsilon \setminus \{X\}$ corresponds exactly to deleting the incidence edge $\{X, \varepsilon\}$ while keeping the node ε adjacent to its remaining incident supervertices. Thus $\mathrm{Inc}(\mathrm{SHG}'^{(n)})$ is obtained from $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ by deleting the leaf X and the edge $\{X, \varepsilon\}$, hence it is a tree.

If $|\varepsilon| = 1$, then $\varepsilon = \{X\}$. Deleting X forces deletion of ε as well to keep the edge family nonempty-subset-valued. In the incidence tree this removes the leaf X and its neighbor ε, again leaving a tree (or the empty graph if all vertices were removed). Therefore, whenever $V' \neq \emptyset$, the resulting incidence graph is a tree and $\mathrm{SHG}'^{(n)}$ is an n-SuperHyperTree. □

Lemma 4.2.14 (Two-coloring supervertices in the tree case)**.** *Let* $\mathrm{SHG}^{(n)} = (V, E)$ *be a finite* n*-SuperHyperTree such that* $|\varepsilon| \ge 2$ *for every* $\varepsilon \in E$*. Then* $\mathrm{SHG}^{(n)}$ *is bipartite in the supervertex sense: there exists a partition* $V = V_1 \sqcup V_2$ *with* $V_1, V_2 \neq \emptyset$ *such that*

$$\forall \varepsilon \in E : \ \varepsilon \cap V_1 \neq \emptyset \ \text{ and } \ \varepsilon \cap V_2 \neq \emptyset.$$

Proof. Since $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ is a finite tree, fix an arbitrary root superedge $\varepsilon_0 \in E$ in the incidence graph. Because $|\varepsilon_0| \ge 2$, choose distinct $X^+, X^- \in \varepsilon_0$ and set $X^+ \in V_1$, $X^- \in V_2$. Color any remaining supervertices of ε_0 arbitrarily, ensuring ε_0 meets both colors.

Now traverse the incidence tree outward from ε_0. When we first encounter an unprocessed superedge ε, it is adjacent (in the incidence tree) to the already processed part through a unique supervertex $X \in \varepsilon$ (uniqueness follows because a tree has unique simple paths, and because by Lemma 4.2.10(i) an edge can intersect the processed region in at most one supervertex). At that moment, X is already colored (in V_1 or V_2). Since $|\varepsilon| \ge 2$, choose some $Y \in \varepsilon \setminus \{X\}$ and assign Y the *opposite* color to X, so that ε contains both colors. Any remaining uncolored supervertices of ε may be colored arbitrarily.

Because each new superedge is attached to the previously processed part through a *single* already colored supervertex, no edge is ever forced to contain two pre-colored vertices with conflicting requirements. Proceeding until all superedges are processed yields a partition $V = V_1 \sqcup V_2$ such that every $\varepsilon \in E$ meets both sides. □

4.3 Rooted Tree, Rooted HyperTree, and Rooted SuperHyperTree

In this section we formalize *rooted* versions of trees, hypertrees, and n-SuperHyperTrees. The guiding principle is the same in all three settings: a rooted structure is obtained by designating a distinguished *root node* and using the uniqueness of paths in a tree to induce a natural parent–child hierarchy (and, if desired, a canonical orientation away from the root).

4.3.1 Rooted trees in graphs

A rooted tree is a tree with a distinguished root vertex, inducing parent–child relations, unique root-to-vertex paths, and a natural hierarchical partial order (cf. [64–67]).

Definition 4.3.1 (Rooted tree)**.** A *rooted tree* is a pair (T, r) where $T = (V, E)$ is a tree (finite, simple, undirected) and $r \in V$ is a distinguished vertex called the *root*.

Definition 4.3.2 (Tree order, parent, and children)**.** Let (T, r) be a rooted tree with $T = (V, E)$. For each $v \in V$, there exists a unique simple path $P_{r\to v}$ from r to v.

(i) The *tree order* $\preceq_r$ on V is defined by

$$u \preceq_r v \quad \Longleftrightarrow \quad u \text{ lies on the unique path } P_{r\to v}.$$

(ii) For $v \neq r$, the *parent* of v, denoted $\mathrm{Pa}(v)$, is the unique neighbor of v on the path $P_{r\to v}$. The *children* of a vertex u are

$$\mathrm{Ch}(u) \;:=\; \{\, v \in V :\; \mathrm{Pa}(v) = u \,\}.$$

Remark 4.3.3 (Canonical orientation)**.** A rooted tree (T, r) admits a canonical orientation away from r by directing each undirected edge $\{u, v\} \in E$ as $u \to v$ whenever $u = \mathrm{Pa}(v)$. The resulting directed graph is an arborescence (out-tree) rooted at r.

4.3.2 Rooted hypertrees in hypergraphs

A rooted hypertree is a hypertree whose incidence graph is rooted at a chosen vertex, inducing parent–child relations alternating between vertex nodes and hyperedge nodes [68, 69]. We use the incidence-graph (Levi-graph) notion of hypertree: a hypergraph is a hypertree if its incidence graph is a tree. Rooting is then defined by choosing a root on the vertex side of the incidence bipartition.

Definition 4.3.4 (Rooted HyperTree)**.** Let $H = (V, E)$ be a finite hypergraph. Assume H is a *HyperTree*, i.e., its incidence graph $\mathrm{Inc}(H)$ (Definition 4.1.3) is a tree. A *rooted HyperTree* is a pair (H, r) where $r \in V$ is a distinguished *root vertex*.

Definition 4.3.5 (Incidence-tree order, parent, and children)**.** Let (H, r) be a rooted HyperTree, and write

$$\mathrm{Inc}(H) \;=\; (\, V \sqcup E,\; F\,)$$

for its incidence graph, rooted at $r \in V \subseteq V \sqcup E$. Because $\mathrm{Inc}(H)$ is a tree, for every node $x \in V \sqcup E$ there is a unique simple path $P_{r\to x}$ in $\mathrm{Inc}(H)$.

(i) The *incidence-tree order* $\preceq_r$ on $V \sqcup E$ is defined by

$$x \preceq_r y \quad\Longleftrightarrow\quad x \text{ lies on the unique path } P_{r\to y} \text{ in } \mathrm{Inc}(H).$$

(ii) For any node $x \neq r$ in $\mathrm{Inc}(H)$, its *parent* $\mathrm{Pa}(x)$ is the unique neighbor of x on the path $P_{r\to x}$. Its *children* are

$$\mathrm{Ch}(x) \;:=\; \{\, y \in V \sqcup E :\; \mathrm{Pa}(y) = x \,\}.$$

Remark 4.3.6 (Canonical orientation on the incidence graph)**.** Rooting $\mathrm{Inc}(H)$ at r induces a canonical orientation away from r by directing each incidence edge $\{x, y\} \in F$ as $x \to y$ whenever $x = \mathrm{Pa}(y)$. This yields a directed bipartite tree that encodes a hierarchical structure alternating between vertices and hyperedges.

Remark 4.3.7 (Alternative conventions)**.** Some authors allow the root to be either a vertex ($r \in V$) or a hyperedge ($r \in E$), i.e., any node of $\mathrm{Inc}(H)$. Definition 4.3.4 fixes $r \in V$ to emphasize vertex-rooted hierarchies.

4.3.3 Rooted SuperHyperTrees in n-SuperHyperGraphs

A rooted superhypertree in an n-SuperHyperGraph is a rooted incidence tree on supervertices and superedges, with a distinguished root supervertex defining hierarchical parent–child structure.

Definition 4.3.8 (Rooted n-SuperHyperTree)**.** Let V_0 be a finite nonempty base set, and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0. Assume $\mathrm{SHG}^{(n)}$ is an *n-SuperHyperTree*, i.e., its incidence graph $\mathrm{Inc}\big(\mathrm{SHG}^{(n)}\big)$ (Definition 4.2.1) is a tree. A *rooted n-SuperHyperTree* is a pair $\big(\mathrm{SHG}^{(n)}, R\big)$ where $R \in V$ is a distinguished *root supervertex*.

Definition 4.3.9 (Incidence-tree order, parent, and children in a rooted n-SuperHyperTree)**.** Let $\big(\mathrm{SHG}^{(n)}, R\big)$ be a rooted n-SuperHyperTree, and write its incidence graph as

$$\mathrm{Inc}\big(\mathrm{SHG}^{(n)}\big) \;=\; \big(V \sqcup E,\; F\big), \qquad F = \big\{\{X, \varepsilon\} :\; X \in V,\; \varepsilon \in E,\; X \in \varepsilon\big\},$$

rooted at $R \in V \subseteq V \sqcup E$. Because $\mathrm{Inc}\big(\mathrm{SHG}^{(n)}\big)$ is a tree, for each node $x \in V \sqcup E$ there is a unique path $P_{R\to x}$.

(i) The *incidence-tree order* $\preceq_R$ on $V \sqcup E$ is

$$x \preceq_R y \quad\Longleftrightarrow\quad x \text{ lies on the unique path } P_{R\to y} \text{ in } \mathrm{Inc}\big(\mathrm{SHG}^{(n)}\big).$$

(ii) For $x \neq R$, the *parent* $\mathrm{Pa}(x)$ is the unique neighbor of x on $P_{R\to x}$, and the *children* are

$$\mathrm{Ch}(x) \;:=\; \{\, y \in V \sqcup E : \; \mathrm{Pa}(y) = x \,\}.$$

Example 4.3.10 (Real-life example of a rooted n-SuperHyperTree)**. Incident-response escalation rooted at the first responder group.** Let V_0 be a finite set of individual responders in an organization, for example,

$$V_0 = \{\mathsf{Ayako}, \mathsf{Satoshi}, \mathsf{Chen}, \mathsf{Dina}, \mathsf{Evan}, \mathsf{Fiona}\}.$$

Take $n = 1$. Define supervertices (each is a set of individuals) by

$$R := X_0 := \{\mathsf{Ayako}, \mathsf{Satoshi}\} \quad (\text{on-call responders}),$$
$$X_1 := \{\mathsf{Chen}, \mathsf{Dina}\} \quad (\text{triage team}),$$
$$X_2 := \{\mathsf{Evan}\} \quad (\text{senior incident lead}),$$
$$X_3 := \{\mathsf{Fiona}\} \quad (\text{executive sponsor}),$$

and set

$$V := \{X_0, X_1, X_2, X_3\} \subseteq \mathcal{P}(V_0).$$

Let superedges represent escalation stages:

$$\varepsilon_1 := \{X_0, X_1\} \quad (\text{handoff from responders to triage}),$$
$$\varepsilon_2 := \{X_1, X_2\} \quad (\text{escalation from triage to senior lead}),$$
$$\varepsilon_3 := \{X_2, X_3\} \quad (\text{escalation from senior lead to executive}),$$

and define

$$E := \{\varepsilon_1, \varepsilon_2, \varepsilon_3\}.$$

Then the incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(1)})$ is the path

$$X_0 - \varepsilon_1 - X_1 - \varepsilon_2 - X_2 - \varepsilon_3 - X_3,$$

which is a tree. Hence $\mathrm{SHG}^{(1)} = (V, E)$ is a 1-SuperHyperTree, and $\big(\mathrm{SHG}^{(1)}, R\big)$ is a rooted 1-SuperHyperTree.

In the rooted incidence tree (rooted at $R = X_0$), the parent/child relations are:

$$\mathrm{Pa}(\varepsilon_1) = X_0,\; \mathrm{Pa}(X_1) = \varepsilon_1,\; \mathrm{Pa}(\varepsilon_2) = X_1,\; \mathrm{Pa}(X_2) = \varepsilon_2,\; \mathrm{Pa}(\varepsilon_3) = X_2,\; \mathrm{Pa}(X_3) = \varepsilon_3.$$

For example, the children sets include

$$\mathrm{Ch}(X_0) = \{\varepsilon_1\}, \quad \mathrm{Ch}(\varepsilon_1) = \{X_1\}, \quad \mathrm{Ch}(X_1) = \{\varepsilon_2\}, \quad \mathrm{Ch}(\varepsilon_2) = \{X_2\}.$$

Thus the rooted n-SuperHyperTree cleanly models a one-directional escalation hierarchy with no cycles.

4.4 k-ary Tree, k-ary HyperTree, and k-ary SuperHyperTree

This section defines k-ary variants of rooted trees, rooted hypertrees, and rooted n-SuperHyperTrees. The common idea is to impose a *branching bound* k on the number of children of each node. For hypergraphs and n-SuperHyperGraphs, we use the rooted *incidence graph* (Levi graph) as the underlying tree on which the branching constraint is imposed.

4.4.1 k-ary rooted trees in graphs

A k-ary rooted tree is a rooted tree where every vertex has at most k children, enforcing a bounded branching factor throughout (cf. [70–75]).

Definition 4.4.1 (k-ary rooted tree)**.** [70–72] Fix an integer $k \geq 0$. A *k-ary rooted tree* is a rooted tree (T, r) (Definitions 4.1.1 and 4.3.1) such that every vertex has at most k children, i.e.,

$$|\mathrm{Ch}(v)| \ \leq \ k \qquad (\forall v \in V(T)).$$

Remark 4.4.2. For $k = 0$, a 0-ary rooted tree consists of a single vertex (the root). For $k = 2$, one obtains binary rooted trees; for $k = 3$, ternary rooted trees.

4.4.2 k-ary rooted hypertrees

A k-ary rooted hypertree is a rooted hypertree whose incidence tree is k-ary, so each vertex-node and hyperedge-node has at most k children.

Definition 4.4.3 (k-ary rooted HyperTree)**.** Fix $k \geq 0$. Let (H, r) be a rooted HyperTree (Definition 4.3.4), so that its incidence graph $\mathrm{Inc}(H) = (V \sqcup E, F)$ is a tree rooted at $r \in V$. We call (H, r) a *k-ary rooted HyperTree* if every node of the rooted incidence tree has at most k children, i.e.,

$$|\mathrm{Ch}(x)| \ \leq \ k \qquad (\forall x \in V \sqcup E),$$

where $\mathrm{Ch}(\cdot)$ is computed in the rooted incidence tree (Definition 4.3.5).

Remark 4.4.4. The bound in Definition 4.4.3 is imposed on the *incidence hierarchy*, so it controls both how many hyperedges can appear "below" a vertex node and how many vertices can appear "below" a hyperedge node in the rooted incidence tree. This is the most direct generalization of k-ary branching to hypergraphs under the incidence-tree model.

4.4.3 k-ary rooted n-SuperHyperTrees

A k-ary rooted n-SuperHyperTree is a rooted n-SuperHyperTree with an incidence hierarchy where every supervertex-node and superedge-node has at most k children.

Definition 4.4.5 (k-ary rooted n-SuperHyperTree)**.** Fix integers $k \geq 0$ and $n \geq 0$. Let $\left(\mathrm{SHG}^{(n)}, R\right)$ be a rooted n-SuperHyperTree (Definition 4.3.8), so that its incidence graph

$$\mathrm{Inc}\left(\mathrm{SHG}^{(n)}\right) \ = \ (V \sqcup E, F)$$

is a tree rooted at the root supervertex $R \in V$. We call $\left(\mathrm{SHG}^{(n)}, R\right)$ a *k-ary rooted n-SuperHyperTree* if every node of this rooted incidence tree has at most k children, i.e.,

$$|\mathrm{Ch}(x)| \ \leq \ k \qquad (\forall x \in V \sqcup E),$$

where $\mathrm{Ch}(\cdot)$ is computed in the rooted incidence tree (Definition 4.3.9).

Example 4.4.6 (Real-life example of a k-ary rooted n-SuperHyperTree)**. A triage escalation hierarchy with bounded branching.** Let V_0 be a finite set of people in a customer-support organization:

$$V_0 = \{\mathsf{Aki}, \mathsf{Ben}, \mathsf{Cara}, \mathsf{Dan}, \mathsf{Eri}, \mathsf{Fay}, \mathsf{Gus}, \mathsf{Hana}\}.$$

Take $n = 1$ so that supervertices are subsets of V_0. Define supervertices (support groups) by

$$R := X_0 := \{\mathsf{Aki}, \mathsf{Ben}\} \quad (\text{Level-1 support}),$$

$$X_1 := \{\mathsf{Cara}, \mathsf{Dan}\} \quad (\text{Billing specialists}),$$

$$X_2 := \{\mathsf{Eri}\} \quad (\text{Security responders}),$$

$$X_3 := \{\mathsf{Fay}, \mathsf{Gus}\} \quad (\text{Platform engineers}),$$

$$X_4 := \{\mathsf{Hana}\} \quad (\text{Director on-call}),$$

and set $V := \{X_0, X_1, X_2, X_3, X_4\} \subseteq \mathcal{P}(V_0)$.

Let superedges represent escalation stages from one group to a bounded set of next-step groups:

$$\varepsilon_0 := \{X_0, X_1, X_2, X_3\}$$

$$(\text{triage from Level-1 to at most three specialist groups}),$$

$$\varepsilon_1 := \{X_1, X_4\},$$

$$\varepsilon_2 := \{X_2, X_4\},$$

$$\varepsilon_3 := \{X_3, X_4\},$$

and define $E := \{\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3\}$.

Then the incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(1)})$ is a tree rooted at $R = X_0$:

$$X_0 - \varepsilon_0 - \{X_1, X_2, X_3\}, \quad X_1 - \varepsilon_1 - X_4,$$

$$X_2 - \varepsilon_2 - X_4, \quad X_3 - \varepsilon_3 - X_4,$$

and there are no cycles.

Moreover, this rooted incidence tree is **3**-ary:

$$|\mathrm{Ch}(X_0)| = 1, \quad |\mathrm{Ch}(\varepsilon_0)| = 3, \quad |\mathrm{Ch}(X_1)| = |\mathrm{Ch}(X_2)| = |\mathrm{Ch}(X_3)| = 1,$$

$$|\mathrm{Ch}(\varepsilon_1)| = |\mathrm{Ch}(\varepsilon_2)| = |\mathrm{Ch}(\varepsilon_3)| = 1,$$

and all other nodes have 0 children. Hence $(\mathrm{SHG}^{(1)}, R)$ is a 3-ary rooted 1-SuperHyperTree, modeling an escalation workflow where each stage branches to at most three next groups.

4.5 Minimum spanning trees in graphs, hypergraphs, and n-SuperHyperGraphs

This section defines minimum spanning trees in three settings. For ordinary graphs, this is standard. For hypergraphs and n-SuperHyperGraphs, there are multiple non-equivalent "spanning" and "tree" concepts in the literature. To keep a mathematically clean and widely used definition that directly generalizes graph MST, we proceed via the *primal (2-section) graph* induced by a weighted hypergraph-like structure. This yields a canonical, graph-theoretic MST that is computable by standard MST algorithms, and reduces to the classical MST when hyperedges are ordinary edges.

4.5.1 Minimum spanning tree in weighted graphs

A minimum spanning tree induced by a weighted graph is a spanning, cycle-free subgraph with $n-1$ edges whose total edge weight is minimal among all spanning trees [76–79]. Related notions are also known, such as minimum spanning trees in fuzzy graphs [80–83] and minimum spanning trees in neutrosophic graphs [84–87].

Definition 4.5.1 (Spanning tree and minimum spanning tree in a weighted graph)**.** [76–79] Let $G = (V, E)$ be a finite, connected, simple undirected graph, and let

$$\mathrm{wt} : E \to \mathbb{R}_{\geq 0}$$

be an edge-weight function.

A *spanning tree* of G is a subgraph $T = (V, E_T)$ with $E_T \subseteq E$ such that T is a tree (connected and acyclic). The *total weight* of T is

$$\mathrm{wt}(T) \;:=\; \sum_{e \in E_T} \mathrm{wt}(e).$$

A *minimum spanning tree* (MST) of (G, wt) is a spanning tree T such that

$$\mathrm{wt}(T) \;=\; \min\bigl\{\mathrm{wt}(T') : \; T' \text{ is a spanning tree of } G\bigr\}.$$

Remark 4.5.2. If G is not connected, the analogous object is a *minimum spanning forest*, obtained by taking an MST in each connected component.

4.5.2 Minimum spanning tree induced by a weighted hypergraph

A minimum spanning tree induced by a weighted hypergraph is obtained by forming the primal graph on vertices, weighting each pair by the cheapest hyperedge containing it, then taking an MST (cf. [88, 89]).

Definition 4.5.3 (Weighted hypergraph)**.** (cf. [90, 91]) A *weighted hypergraph* is a pair (H, wt) where $H = (V, \mathcal{E})$ is a finite hypergraph with $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$ and

$$\mathrm{wt} : \mathcal{E} \to \mathbb{R}_{\geq 0}$$

assigns a nonnegative weight to each hyperedge.

Definition 4.5.4 (Primal (2-section) graph of a weighted hypergraph)**.** Let (H, wt) be a weighted hypergraph with $H = (V, \mathcal{E})$. Its *primal graph* (or *2-section*) is the simple undirected graph

$$\mathrm{Pr}(H) \;=\; (V, E_{\mathrm{Pr}}), \qquad E_{\mathrm{Pr}} \;:=\; \bigl\{\{u, v\} : \; u, v \in V, \; u \neq v, \; \exists\, e \in \mathcal{E} \text{ with } \{u, v\} \subseteq e\bigr\}.$$

Define an induced edge-weight function $\mathrm{wt}_{\mathrm{Pr}} : E_{\mathrm{Pr}} \to \mathbb{R}_{\geq 0}$ by

$$\mathrm{wt}_{\mathrm{Pr}}(\{u, v\}) \;:=\; \min\bigl\{\mathrm{wt}(e) : \; e \in \mathcal{E}, \; \{u, v\} \subseteq e\bigr\}.$$

Definition 4.5.5 (Minimum spanning tree induced by a weighted hypergraph)**.** Let (H, wt) be a weighted hypergraph such that its primal graph $\mathrm{Pr}(H)$ is connected. A *minimum spanning tree induced by* (H, wt) is any minimum spanning tree of the weighted graph

$$\big(\mathrm{Pr}(H), \mathrm{wt}_{\mathrm{Pr}}\big)$$

in the sense of Definition 4.5.1.

Remark 4.5.6. If H is a graph (i.e., every hyperedge has size 2), then $\mathrm{Pr}(H) = H$ and $\mathrm{wt}_{\mathrm{Pr}} = \mathrm{wt}$, so Definition 4.5.5 reduces exactly to the classical MST definition.

Remark 4.5.7. Definition 4.5.5 is one canonical way to extend MST to hypergraphs. Other approaches (e.g., defining a "hypertree" as an acyclic hypergraph in an incidence sense, or optimizing over hyperedge selections) lead to different optimization problems. The primal-graph definition is particularly useful when hyperedges represent possible group connections and $\mathrm{wt}(e)$ is interpreted as a cost that can realize any pairwise connection within e.

4.5.3 Minimum spanning tree induced by a weighted n-SuperHyperGraph

A minimum spanning tree induced by a weighted n-superhypergraph is defined similarly: build the primal graph on supervertices, assign each pair the minimum superedge weight containing them, then compute an MST.

Definition 4.5.8 (Weighted n-SuperHyperGraph)**.** [55] Let V_0 be a finite nonempty base set and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0, with $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. A *weight function* is a map

$$\mathrm{wt} : E \to \mathbb{R}_{\geq 0}.$$

The pair $\big(\mathrm{SHG}^{(n)}, \mathrm{wt}\big)$ is called a *weighted n-SuperHyperGraph*.

Definition 4.5.9 (Primal graph of a weighted n-SuperHyperGraph)**.** Let $\big(\mathrm{SHG}^{(n)}, \mathrm{wt}\big)$ be a weighted n-SuperHyperGraph with $\mathrm{SHG}^{(n)} = (V, E)$. Define its *primal graph* $\mathrm{Pr}(\mathrm{SHG}^{(n)}) = (V, E_{\mathrm{Pr}})$ by

$$E_{\mathrm{Pr}} \;:=\; \big\{\{X, Y\} : \; X, Y \in V,\; X \neq Y,\; \exists\, \varepsilon \in E \text{ with } \{X, Y\} \subseteq \varepsilon\big\}.$$

Define the induced weight $\mathrm{wt}_{\mathrm{Pr}} : E_{\mathrm{Pr}} \to \mathbb{R}_{\geq 0}$ by

$$\mathrm{wt}_{\mathrm{Pr}}(\{X, Y\}) \;:=\; \min\big\{\mathrm{wt}(\varepsilon) : \; \varepsilon \in E,\; \{X, Y\} \subseteq \varepsilon\big\}.$$

Definition 4.5.10 (Minimum spanning tree induced by a weighted n-SuperHyperGraph)**.** Let $\big(\mathrm{SHG}^{(n)}, \mathrm{wt}\big)$ be a weighted n-SuperHyperGraph such that its primal graph $\mathrm{Pr}(\mathrm{SHG}^{(n)})$ is connected. A *minimum spanning tree induced by* $\big(\mathrm{SHG}^{(n)}, \mathrm{wt}\big)$ is any minimum spanning tree of

$$\big(\mathrm{Pr}(\mathrm{SHG}^{(n)}), \mathrm{wt}_{\mathrm{Pr}}\big)$$

in the sense of Definition 4.5.1.

Remark 4.5.11. Definition 4.5.10 treats (V, E) as a hypergraph on n-supervertices and applies the same primal-graph construction as in the hypergraph case. If one wishes to relate this to base-level connectivity on V_0, one may additionally compose with flattening Flat_n, but the MST defined above is intrinsic at the supervertex level.

Example 4.5.12 (Real-life example of a minimum spanning tree induced by a weighted n-SuperHyperGraph)**. Selecting the cheapest integration plan across nested vendor consortia.** Let V_0 be a finite set of individual vendors in a logistics project:

$$V_0 = \{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}\}.$$

Take $n = 1$, so 1-supervertices are (nonempty) subsets of V_0. Interpret each supervertex as a *vendor consortium* that can deliver a subsystem.

Define four consortia (supervertices)

$$X_1 := \{\mathsf{A}\} \quad (\text{warehouse operator}), \qquad X_2 := \{\mathsf{B}\} \quad (\text{truck fleet}),$$

$$X_3 := \{\mathsf{C}\} \quad (\text{customs broker}), \qquad X_4 := \{\mathsf{D}\} \quad (\text{last-mile courier}),$$

and set $V := \{X_1, X_2, X_3, X_4\} \subseteq \mathcal{P}(V_0)$.

Superedges represent *multi-party integration contracts* that jointly enable interoperability among several consortia, and $\mathrm{wt}(\varepsilon)$ is the estimated contract cost (in millions of JPY). Let

$$\varepsilon_1 := \{X_1, X_2, X_3\}, \quad \mathrm{wt}(\varepsilon_1) = 5,$$

$$\varepsilon_2 := \{X_2, X_4\}, \quad \mathrm{wt}(\varepsilon_2) = 2,$$

$$\varepsilon_3 := \{X_1, X_4\}, \quad \mathrm{wt}(\varepsilon_3) = 4,$$

and define $E := \{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$.

Then the primal graph $\mathrm{Pr}(\mathrm{SHG}^{(1)})$ has vertex set V and contains the edges

$$\{X_1, X_2\}, \{X_1, X_3\}, \{X_2, X_3\} \quad (\text{from } \varepsilon_1), \qquad \{X_2, X_4\} \quad (\text{from } \varepsilon_2), \qquad \{X_1, X_4\} \quad (\text{from } \varepsilon_3).$$

The induced weights $\mathrm{wt}_{\mathrm{Pr}}$ are computed by taking the minimum superedge cost that contains the pair:

$$\mathrm{wt}_{\mathrm{Pr}}(\{X_1, X_2\}) = \mathrm{wt}_{\mathrm{Pr}}(\{X_1, X_3\}) = \mathrm{wt}_{\mathrm{Pr}}(\{X_2, X_3\}) = 5,$$

$$\mathrm{wt}_{\mathrm{Pr}}(\{X_2, X_4\}) = 2, \qquad \mathrm{wt}_{\mathrm{Pr}}(\{X_1, X_4\}) = 4.$$

A minimum spanning tree of $\big(\mathrm{Pr}(\mathrm{SHG}^{(1)}), \mathrm{wt}_{\mathrm{Pr}}\big)$ is, for example,

$$T = \big\{\{X_2, X_4\},\ \{X_1, X_4\},\ \{X_1, X_3\}\big\},$$

with total weight $2 + 4 + 5 = 11$. Interpreted operationally, this selects the cheapest set of pairwise interoperability links (each supported by some multi-party contract) that connects all consortia, hence yielding a minimum-cost integration backbone for the project.

4.6 Decision Tree, Decision HyperTree, and Decision SuperHyperTree

This section formalizes decision-analytic "tree" models in three settings: (i) ordinary rooted trees (graphs), (ii) rooted hypertrees (hypergraphs whose incidence graph is a tree), and (iii) rooted superhypertrees (n-SuperHyperGraphs whose incidence graph is a tree). The key modeling idea is the same throughout: internal nodes represent either *decisions* or *chance*, and leaves represent terminal *utilities* (or payoffs). Note that related notions to a decision tree include fuzzy decision trees [92, 93] and neutrosophic decision trees [94, 95], among others.

4.6.1 Decision trees (graph-theoretic)

A decision tree is a rooted tree where internal nodes represent decisions or chance events, branches represent outcomes, and leaves store terminal payoffs or class labels [96–98].

Definition 4.6.1 (Decision tree)**.** [96–98] A *decision tree* is a tuple

$$\mathcal{T} \;=\; (T, r, \mathrm{Type}, \mathrm{Lab}, \mathrm{Prob}, \mathrm{Util})$$

satisfying the following conditions.

(i) $T = (V, E)$ is a finite tree (simple, undirected, connected, acyclic), and $r \in V$ is a distinguished root.

(ii) $\mathrm{Type} : V \to \{\mathsf{dec}, \mathsf{ch}, \mathsf{term}\}$ assigns each vertex a *node type* (decision, chance, or terminal), and every terminal node is a leaf:

$$\mathrm{Type}(v) = \mathsf{term} \implies \deg_T(v) = 1 \text{ or } (v = r \text{ and } |V| = 1).$$

(iii) For each nonterminal node v (i.e., $\mathrm{Type}(v) \in \{\mathsf{dec}, \mathsf{ch}\}$), let $\mathrm{Ch}_T(v)$ denote the set of children of v in the rooted tree (T, r). (Equivalently, orient each edge away from r; then $\mathrm{Ch}_T(v)$ are the out-neighbors of v.) We require $\mathrm{Ch}_T(v) \neq \emptyset$.

(iv) Lab labels the branches out of each nonterminal node: for every v with $\mathrm{Type}(v) \in \{\mathsf{dec}, \mathsf{ch}\}$ there exists a finite nonempty label set $L(v)$ and a bijection

$$\mathrm{Lab}_v : \mathrm{Ch}_T(v) \longrightarrow L(v),$$

so that each outgoing branch $v \to u$ is identified with a unique label $\mathrm{Lab}_v(u) \in L(v)$.

(v) Prob assigns probabilities at chance nodes: for every v with $\mathrm{Type}(v) = \mathsf{ch}$, Prob_v is a probability mass function on $L(v)$,

$$\mathrm{Prob}_v : L(v) \to [0, 1], \qquad \sum_{\ell \in L(v)} \mathrm{Prob}_v(\ell) = 1.$$

(For decision nodes v with $\mathrm{Type}(v) = \mathsf{dec}$, no probability is imposed.)

(vi) Util assigns terminal utilities:

$$\mathrm{Util} : \{v \in V : \mathrm{Type}(v) = \mathsf{term}\} \longrightarrow \mathbb{R}.$$

Remark 4.6.2 (Policies and expected utility)**.** A *policy* (or strategy) for a decision tree chooses one branch at each decision node:

$$\pi(v) \in L(v) \qquad (\forall v \in V \text{ with } \mathrm{Type}(v) = \mathsf{dec}).$$

Under π, the tree induces a probability distribution over terminal leaves via the chance-node probabilities. The expected utility can then be computed by backward induction (dynamic programming) on the rooted tree.

4.6.2 Decision hypertrees

We use the incidence-tree (Levi-graph) model of hypertrees. Branching in a decision hypertree occurs through hyperedges, which act as multiway branch gadgets.

Definition 4.6.3 (Incidence graph of a hypergraph)**.** Let $H = (V, E)$ be a finite hypergraph, $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Its *incidence graph* is the bipartite graph

$$\mathrm{Inc}(H) \;=\; \big(V \sqcup E,\ F\big), \qquad F = \big\{\{v, e\} : v \in V,\ e \in E,\ v \in e\big\}.$$

Definition 4.6.4 (Decision HyperTree)**.** (cf. [99]) A *decision HyperTree* is a tuple

$$\mathcal{H} \;=\; (H, r, \mathrm{Type}, \mathrm{Lab}, \mathrm{Prob}, \mathrm{Util})$$

such that:

(i) (H, r) is a rooted HyperTree with $H = (V, E)$.

(ii) $\mathrm{Type} : V \to \{\mathsf{dec}, \mathsf{ch}, \mathsf{term}\}$ assigns node types to the *vertex-nodes* V. Terminal vertices are leaves in the rooted incidence tree in the following sense:

$$\mathrm{Type}(v) = \mathsf{term} \implies \{\, e \in E :\ \mathrm{Pa}(e) = v \,\} = \emptyset.$$

(iii) (*Unique branching hyperedge per internal vertex*) For every $v \in V$ with $\mathrm{Type}(v) \in \{\mathsf{dec}, \mathsf{ch}\}$, there is exactly one incidence-child hyperedge:

$$\mathrm{Out}(v) \;:=\; \{\, e \in E :\ \mathrm{Pa}(e) = v \,\} \quad \text{satisfies} \quad |\,\mathrm{Out}(v)| = 1.$$

Write $\mathrm{Out}(v) = \{e_v\}$. Define the set of *branch children* of v by

$$\mathrm{Ch}_V(v) \;:=\; \{\, u \in V :\ \mathrm{Pa}(u) = e_v \,\}.$$

We require $\mathrm{Ch}_V(v) \neq \emptyset$.

(iv) (*Branch labels*) For each internal vertex v (decision or chance), there exists a finite nonempty label set $L(v)$ and a bijection

$$\mathrm{Lab}_v : \mathrm{Ch}_V(v) \longrightarrow L(v).$$

(v) (*Chance probabilities*) For each $v \in V$ with $\mathrm{Type}(v) = \mathsf{ch}$, there is a probability mass function

$$\mathrm{Prob}_v : L(v) \to [0, 1], \qquad \sum_{\ell \in L(v)} \mathrm{Prob}_v(\ell) = 1.$$

(vi) (*Terminal utilities*)

$$\mathrm{Util} : \{v \in V : \mathrm{Type}(v) = \mathsf{term}\} \longrightarrow \mathbb{R}.$$

Remark 4.6.5 (Traversal semantics)**.** Starting at the root vertex r, an internal vertex v branches via its unique child hyperedge e_v to one of the vertices in $\mathrm{Ch}_V(v)$, labeled by Lab_v. Thus the rooted incidence tree alternates

$$v \;\to\; e_v \;\to\; u,$$

and decision/chance semantics are carried by the vertex v, while the hyperedge e_v encodes the multiway branching structure.

4.6.3 Decision superhypertrees

A decision superhypertree extends decision hypertrees to n-SuperHyperTrees: supervertices represent decision/chance states, superedges encode outcome branching, and terminal supervertices provide final utilities.

Definition 4.6.6 (Rooted n-SuperHyperTree)**.** Let V_0 be a finite nonempty base set and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph with $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. It is an *$n$-SuperHyperTree* if its incidence graph

$$\mathrm{Inc}\big(\mathrm{SHG}^{(n)}\big) \;=\; \big(V \sqcup E,\; F\big), \qquad F = \{\{X, \varepsilon\} : X \in V,\; \varepsilon \in E,\; X \in \varepsilon\},$$

is a tree. A *rooted n-SuperHyperTree* is a pair $\big(\mathrm{SHG}^{(n)}, R\big)$ with root supervertex $R \in V$.

Definition 4.6.7 (Decision n-SuperHyperTree)**.** A *decision n-SuperHyperTree* is a tuple

$$\mathcal{S} \;=\; \big(\mathrm{SHG}^{(n)}, R, \mathrm{Type}, \mathrm{Lab}, \mathrm{Prob}, \mathrm{Util}\big)$$

satisfying the following conditions.

(i) $\big(\mathrm{SHG}^{(n)}, R\big)$ is a rooted n-SuperHyperTree with vertex set V (supervertices) and superedge set E.

(ii) $\mathrm{Type} : V \to \{\mathsf{dec}, \mathsf{ch}, \mathsf{term}\}$ assigns node types to supervertices. Terminal supervertices have no incidence-child superedge in the rooted incidence tree:

$$\mathrm{Type}(X) = \mathsf{term} \implies \{\, \varepsilon \in E : \; \mathrm{Pa}(\varepsilon) = X \,\} = \emptyset.$$

(iii) (*Unique branching superedge per internal supervertex*) For every $X \in V$ with $\mathrm{Type}(X) \in \{\mathsf{dec}, \mathsf{ch}\}$,

$$\mathrm{Out}(X) := \{\, \varepsilon \in E : \ \mathrm{Pa}(\varepsilon) = X \,\} \quad \text{satisfies} \quad |\,\mathrm{Out}(X)| = 1.$$

Write $\mathrm{Out}(X) = \{\varepsilon_X\}$ and define the branch-children supervertices

$$\mathrm{Ch}_V(X) := \{\, Y \in V : \ \mathrm{Pa}(Y) = \varepsilon_X \,\},$$

requiring $\mathrm{Ch}_V(X) \neq \emptyset$.

(iv) (*Branch labels*) For each internal supervertex X, there exists a finite nonempty label set $L(X)$ and a bijection

$$\mathrm{Lab}_X : \mathrm{Ch}_V(X) \longrightarrow L(X).$$

(v) (*Chance probabilities*) For each $X \in V$ with $\mathrm{Type}(X) = \mathsf{ch}$, there is a probability mass function

$$\mathrm{Prob}_X : L(X) \to [0,1], \qquad \sum_{\ell \in L(X)} \mathrm{Prob}_X(\ell) = 1.$$

(vi) (*Terminal utilities*)

$$\mathrm{Util} : \{X \in V : \mathrm{Type}(X) = \mathsf{term}\} \longrightarrow \mathbb{R}.$$

Remark 4.6.8. A decision n-SuperHyperTree is a decision hypertree whose "vertices" are nested set-valued objects (n-supervertices $X \in \mathcal{P}^n(V_0)$). The decision/chance semantics are defined intrinsically at the supervertex/superedge incidence level; if desired, one may relate these to base-level objects via flattening Flat_n, but this is not required for the definition.

Example 4.6.9 (Real-life example of a decision n-SuperHyperTree)**. Cybersecurity incident response with nested teams and uncertain outcomes.** Let V_0 be a finite set of individual responders in an organization:

$$V_0 = \{\mathsf{SOC1}, \mathsf{SOC2}, \mathsf{IR1}, \mathsf{IR2}, \mathsf{Legal}, \mathsf{PR}\}.$$

Take $n = 2$. Interpret a level-1 object as a *team* (a subset of V_0), and a level-2 object as a *cluster of teams* (a set of teams), hence an element of $\mathcal{P}^2(V_0)$. Define three supervertices (each is a set of teams):

$$X_0 := \big\{\{\mathsf{SOC1}, \mathsf{SOC2}\}, \{\mathsf{IR1}, \mathsf{IR2}\}\big\} \quad (\text{detection \& containment cluster}),$$

$$X_1 := \big\{\{\mathsf{Legal}\}, \{\mathsf{PR}\}\big\} \quad (\text{notification \& communications cluster}),$$

$$X_2 := \big\{\{\mathsf{IR1}, \mathsf{IR2}\}, \{\mathsf{Legal}\}, \{\mathsf{PR}\}\big\} \quad (\text{full incident committee}).$$

Let $V := \{X_0, X_1, X_2\} \subseteq \mathcal{P}^2(V_0)$.

Model one decision point and one chance point as follows. Let $R := X_0$ be the root supervertex (initial state after an alert is raised), and set

$$\mathrm{Type}(R) = \mathsf{dec}, \qquad \mathrm{Type}(X_2) = \mathsf{ch}, \qquad \mathrm{Type}(X_1) = \mathsf{term}.$$

Define superedges (branching steps) by

$$\varepsilon_R := \{X_0, X_2\} \in E \quad (\textsf{decision: escalate to full committee or not}),$$

$$\varepsilon_{X_2} := \{X_2, X_1\} \in E \quad (\textsf{chance: outcome determines whether public notification is required}).$$

Thus $E := \{\varepsilon_R, \varepsilon_{X_2}\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. The incidence graph is the path $X_0 - \varepsilon_R - X_2 - \varepsilon_{X_2} - X_1$, hence a tree, so $(\mathrm{SHG}^{(2)}, R)$ is a rooted 2-SuperHyperTree.

Specify branch labels:

$$L(X_0) = \{\textsf{Escalate}\}, \quad \mathrm{Lab}_{X_0}(X_2) = \textsf{Escalate}, \qquad L(X_2) = \{\textsf{Notify}\}, \quad \mathrm{Lab}_{X_2}(X_1) = \textsf{Notify}.$$

Assign a chance probability at X_2 (for example, "regulatory notification is required"):

$$\mathrm{Prob}_{X_2}(\textsf{Notify}) = 1,$$

and assign a terminal utility to X_1 (for example, negative utility represents cost):

$$\mathrm{Util}(X_1) = -10 \quad (\textsf{high cost due to mandatory disclosure and reputational impact}).$$

In this model, the root supervertex X_0 represents a *nested organizational state* (a cluster of teams), the superedge ε_R encodes a *decision* to escalate to the incident committee X_2, and the subsequent superedge ε_{X_2} encodes a *chance-driven transition* to a terminal outcome X_1 with an associated utility.

4.7 Binary Tree, Binary HyperTree, and Binary SuperHyperTree

In this section, "binary" means *at most two children* at each node of a rooted tree-like structure. For hypergraphs and n-SuperHyperGraphs, we impose the binary constraint on the rooted *incidence tree* (Levi graph), which is the most direct analogue of branching in ordinary rooted trees. Related notions such as fuzzy binary trees have also been studied [100, 101].

4.7.1 Binary trees in graphs

A binary tree is a rooted tree where each node h as at most two children; optionally, children are distinguished as left and right [102–105].

Definition 4.7.1 (Binary rooted tree)**.** A *binary tree* is a 2-ary rooted tree. More explicitly, it is a rooted tree (T, r) such that every vertex has at most two children:

$$|\mathrm{Ch}(v)| \ \leq \ 2 \qquad (\forall v \in V(T)).$$

Remark 4.7.2 (Left and right children (optional structure))**.** In computer science, binary trees often distinguish a *left* and *right* child. Formally, this corresponds to equipping each vertex v with an injective map

$$\lambda_v : \mathrm{Ch}(v) \hookrightarrow \{\textsf{L}, \textsf{R}\},$$

so that different children receive different side labels. This left/right order is additional structure and is not required for the graph-theoretic definition in Definition 4.7.1.

Definition 4.7.3 (Full and perfect binary trees (optional))**.** Let (T, r) be a binary tree.

(i) (T, r) is *full* (or *proper/strict*) if every vertex has either 0 or 2 children.

(ii) (T, r) is *perfect* if it is full and all leaves have the same depth.

4.7.2 Binary hypertrees in hypergraphs

A binary hypertree is a rooted hypertree whose incidence tree is binary, so every vertex-node and hyperedge-node has at most two children (cf. [106]).

Definition 4.7.4 (Binary rooted HyperTree)**.** Let (H, r) be a rooted HyperTree, i.e., $H = (V, E)$ is a hypergraph whose incidence graph $\mathrm{Inc}(H) = (V \sqcup E, F)$ is a tree rooted at $r \in V$ (Definitions 4.1.3 and 4.3.4). We call (H, r) a *binary HyperTree* if every node of the rooted incidence tree has at most two children:

$$|\mathrm{Ch}(x)| \leq 2 \qquad (\forall x \in V \sqcup E),$$

where $\mathrm{Ch}(\cdot)$ is computed in the rooted tree $\mathrm{Inc}(H)$ (Definition 4.3.5).

Remark 4.7.5. Because $\mathrm{Inc}(H)$ is bipartite, the binary constraint alternately bounds: (i) the number of hyperedges whose parent is a given vertex-node, and (ii) the number of vertex-nodes whose parent is a given hyperedge-node. Thus, a binary HyperTree enforces binary branching across both vertex and hyperedge levels in the incidence hierarchy.

4.7.3 Binary superhypertrees in n-SuperHyperGraphs

A binary superhypertree is a rooted n-SuperHyperTree with a binary incidence hierarchy, meaning each supervertex-node and superedge-node has at most two children.

Definition 4.7.6 (Binary rooted n-SuperHyperTree)**.** Let $\left(\mathrm{SHG}^{(n)}, R\right)$ be a rooted n-SuperHyperTree, i.e., $\mathrm{SHG}^{(n)} = (V, E)$ is an n-SuperHyperGraph whose incidence graph

$$\mathrm{Inc}\!\left(\mathrm{SHG}^{(n)}\right) = (V \sqcup E, F)$$

is a tree rooted at the supervertex $R \in V$ (Definitions 4.2.1 and 4.3.8). We call $\left(\mathrm{SHG}^{(n)}, R\right)$ a *binary n-SuperHyperTree* if every node of this rooted incidence tree has at most two children:

$$|\mathrm{Ch}(x)| \leq 2 \qquad (\forall x \in V \sqcup E),$$

where $\mathrm{Ch}(\cdot)$ is computed in the rooted incidence tree (Definition 4.3.9).

Remark 4.7.7. Definition 4.7.6 is intrinsic at the supervertex/superedge incidence level. If one wishes to model binary branching at the base level V_0, one may additionally combine this with flattening (Flat_n) to define derived base-level constraints, but that is optional and not required for the binary SuperHyperTree notion.

Example 4.7.8 (Real-life example of a binary n-SuperHyperTree)**. Binary escalation in an on-call incident process.** Let V_0 be a finite set of people involved in incident response:

$$V_0 = \{\mathsf{EngOnCall}, \mathsf{SRELead}, \mathsf{SecurityOnCall}, \mathsf{Comms}, \mathsf{VP}\}.$$

Take $n = 1$, so supervertices are subsets of V_0. Define three supervertices (response groups):

$$R := X_0 := \{\mathsf{EngOnCall}\} \quad \text{(initial responder)},$$

$$X_1 := \{\mathsf{SRELead}, \mathsf{SecurityOnCall}\} \quad \text{(technical escalation)},$$

$$X_2 := \{\mathsf{Comms}, \mathsf{VP}\} \quad \text{(executive communications escalation)},$$

and set $V := \{X_0, X_1, X_2\} \subseteq \mathcal{P}(V_0)$.

Let superedges encode a binary branching escalation:

$$\varepsilon_0 := \{X_0, X_1\} \quad \text{(escalate from responder to technical lead)},$$

$$\varepsilon_1 := \{X_1, X_2\} \quad \text{(escalate from technical lead to executives)},$$

and define $E := \{\varepsilon_0, \varepsilon_1\}$.

Then the incidence graph is the path

$$X_0 - \varepsilon_0 - X_1 - \varepsilon_1 - X_2,$$

which is a tree rooted at $R = X_0$. Moreover, it is *binary* in the sense that every node has at most two children in the rooted incidence tree:

$$|\mathrm{Ch}(X_0)| = 1, \quad |\mathrm{Ch}(\varepsilon_0)| = 1, \quad |\mathrm{Ch}(X_1)| = 1, \quad |\mathrm{Ch}(\varepsilon_1)| = 1, \quad |\mathrm{Ch}(X_2)| = 0.$$

Hence $\big(\mathrm{SHG}^{(1)}, R\big)$ is a binary 1-SuperHyperTree, modeling a strictly two-way incidence hierarchy (no node branches to more than two children) in a real escalation workflow.

4.8 Binary Search Tree, Binary Search HyperTree, and Binary Search SuperHyperTree

This section defines a binary-search (ordered) structure in three increasingly general settings. The classical binary search tree (BST) is an *ordered* rooted binary tree whose node keys satisfy a left–right monotonicity property. To extend this idea to hypertrees and superhypertrees, we (i) keep the underlying rooted *incidence tree* framework and (ii) place the search-order constraint on the *vertex-type* (respectively supervertex-type) nodes, while hyperedge/superedge nodes act as branching gadgets.

4.8.1 Binary search trees (BSTs)

A binary search tree is a rooted binary tree with ordered keys: all keys in the left subtree are smaller, and all in the right subtree are larger [107–110].

Definition 4.8.1 (Binary search tree)**.** Let $(K, \leq)$ be a totally ordered set (key domain). A *binary search tree* is a tuple

$$\mathcal{B} = (T, r, \mathrm{key}, \lambda)$$

such that:

(i) (T, r) is a rooted binary tree, i.e., $T = (V, E)$ is a tree with root r and $|\mathrm{Ch}(v)| \leq 2$ for all $v \in V$.

(ii) key : $V \to K$ assigns a *key* to each vertex.

(iii) λ is a left/right labeling of children: for each $v \in V$, $\lambda_v : \mathrm{Ch}(v) \to \{\mathsf{L}, \mathsf{R}\}$ is injective. If $\lambda_v(u) = \mathsf{L}$ we call u the *left child* of v; if $\lambda_v(u) = \mathsf{R}$ we call u the *right child*.

(iv) (*Binary search property*) For every vertex $v \in V$:

(a) if u lies in the left subtree of v (i.e., $u \preceq v$ in the rooted order with first edge labeled L), then $\mathrm{key}(u) < \mathrm{key}(v)$;

(b) if u lies in the right subtree of v, then $\mathrm{key}(u) > \mathrm{key}(v)$.

Remark 4.8.2 (Handling duplicate keys)**.** Definition 4.8.1 uses strict inequalities and therefore assumes distinct keys. To allow duplicates, one may replace $<$ and $>$ by $\leq$ and $>$ (or $<$ and $\geq$), which corresponds to choosing a consistent convention for storing ties.

4.8.2 Binary search hypertrees

A binary search hypertree is a rooted hypertree whose incidence tree is binary; vertex keys satisfy BST ordering across vertex subtrees, with hyperedges mediating left/right branching.

Definition 4.8.3 (Rooted binary HyperTree)**.** Let $H = (V, E)$ be a hypergraph and $r \in V$. Assume (H, r) is a rooted HyperTree, i.e., its incidence graph $\mathrm{Inc}(H) = (V \sqcup E, F)$ is a tree rooted at r, and moreover is *binary* in the sense that every node of $\mathrm{Inc}(H)$ has at most two children. (See Definition 4.7.4.)

Definition 4.8.4 (Binary search HyperTree)**.** Let $(K, \leq)$ be a totally ordered set. A *binary search HyperTree* is a tuple

$$\mathcal{BH} \;=\; (H, r, \mathrm{key}, \lambda)$$

satisfying:

(i) (H, r) is a rooted *binary* HyperTree as in Definition 4.8.3, with incidence tree $\mathrm{Inc}(H) = (V \sqcup E, F)$ rooted at $r \in V$.

(ii) key : $V \to K$ assigns a key to each *vertex* $v \in V$. (Hyperedges $e \in E$ are not keyed.)

(iii) (*Unique branching hyperedge per internal vertex*) For each vertex $v \in V$ that is *internal on the vertex level* (i.e., it has a child in the incidence tree), there is exactly one incidence-child hyperedge:

$$\mathrm{Out}(v) \; := \; \{\, e \in E : \; \mathrm{Pa}(e) = v \,\} \quad \text{satisfies} \quad |\,\mathrm{Out}(v)| = 1.$$

Write $\mathrm{Out}(v) = \{e_v\}$, and define its vertex-children via that hyperedge:

$$\mathrm{Ch}_V(v) \; := \; \{\, u \in V : \; \mathrm{Pa}(u) = e_v \,\}.$$

Because the incidence tree is binary, $|\mathrm{Ch}_V(v)| \leq 2$.

(iv) (*Left/right labeling at vertex branching*) For each vertex v with $|\mathrm{Ch}_V(v)| = 2$, fix an injective labeling

$$\lambda_v : \mathrm{Ch}_V(v) \hookrightarrow \{\mathsf{L}, \mathsf{R}\}.$$

If $|\mathrm{Ch}_V(v)| = 1$, the unique child may be labeled either L or R by convention.

(v) (*Binary search property on vertex subtrees*) For every vertex $v \in V$:

(a) for any vertex $u \in V$ lying in the *left vertex-subtree* of v (i.e., the unique path in $\mathrm{Inc}(H)$ from v to u begins with $v \to e_v \to u_1$ where $\lambda_v(u_1) = \mathsf{L}$), one has $\mathrm{key}(u) < \mathrm{key}(v)$;

(b) for any vertex $u \in V$ lying in the *right vertex-subtree* of v, one has $\mathrm{key}(u) > \mathrm{key}(v)$.

Remark 4.8.5. A binary search HyperTree is essentially a BST whose branching is mediated by hyperedges in the incidence tree: the traversal alternates $v \to e_v \to u$. The search-order constraint is imposed only on the vertex nodes V, which are the natural "items" being stored.

4.8.3 Binary search superhypertrees

A binary search superhypertree generalizes this to n-SuperHyperTrees: keyed supervertices obey BST ordering across supervertex subtrees, while superedges encode the binary branching structure.

Definition 4.8.6 (Rooted binary n-SuperHyperTree)**.** Let V_0 be a finite nonempty base set and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph. Assume $\left(\mathrm{SHG}^{(n)}, R\right)$ is a rooted n-SuperHyperTree whose incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(n)}) = (V \sqcup E, F)$ is a *binary* rooted tree (Definition 4.7.6).

Definition 4.8.7 (Binary search n-SuperHyperTree)**.** Let $(K, \leq)$ be a totally ordered set. A *binary search n-SuperHyperTree* is a tuple

$$\mathcal{BSH} \; = \; \left(\mathrm{SHG}^{(n)}, R, \mathrm{key}, \lambda\right)$$

satisfying:

(i) $(\mathrm{SHG}^{(n)}, R)$ is a rooted *binary* n-SuperHyperTree as in Definition 4.8.6, with supervertex set $V \subseteq \mathcal{P}^n(V_0)$ and superedge set $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$.

(ii) $\mathrm{key} : V \to K$ assigns a key to each *supervertex* $X \in V$. (Superedges are not keyed.)

(iii) (*Unique branching superedge per internal supervertex*) For each supervertex $X \in V$ that is internal in the incidence tree,

$$\mathrm{Out}(X) \ := \ \{\, \varepsilon \in E : \ \mathrm{Pa}(\varepsilon) = X \,\} \quad \text{satisfies} \quad |\,\mathrm{Out}(X)| = 1.$$

Write $\mathrm{Out}(X) = \{\varepsilon_X\}$ and define supervertex-children

$$\mathrm{Ch}_V(X) \ := \ \{\, Y \in V : \ \mathrm{Pa}(Y) = \varepsilon_X \,\},$$

so that $|\mathrm{Ch}_V(X)| \le 2$ because the incidence tree is binary.

(iv) (*Left/right labeling*) For each $X \in V$ with $|\mathrm{Ch}_V(X)| = 2$, fix an injective labeling

$$\lambda_X : \mathrm{Ch}_V(X) \hookrightarrow \{\mathsf{L}, \mathsf{R}\}.$$

(v) (*Binary search property on supervertex subtrees*) For every supervertex $X \in V$:

(a) for any supervertex $Y \in V$ in the left supervertex-subtree of X, one has $\mathrm{key}(Y) < \mathrm{key}(X)$;

(b) for any supervertex $Y \in V$ in the right supervertex-subtree of X, one has $\mathrm{key}(Y) > \mathrm{key}(X)$.

Here "left/right supervertex-subtree" is defined via the first supervertex-step $X \to \varepsilon_X \to Y_1$ in the unique incidence-path from X to Y, using the label $\lambda_X(Y_1)$.

Remark 4.8.8. A binary search n-SuperHyperTree can be viewed as a BST whose stored objects are nested set-valued supervertices, and whose branching structure is mediated by superedges in the incidence tree. If one wishes to define keys from base-level data, one may compose key with flattening Flat_n, but the ordered-search property in Definition 4.8.7 is formulated intrinsically at the supervertex level.

Example 4.8.9 (Real-life example of a binary search n-SuperHyperTree)**. Indexing nested customer segments for fast compliance lookup.** Let V_0 be a finite set of customer accounts in a bank:

$$V_0 = \{\mathsf{C001}, \mathsf{C002}, \mathsf{C003}, \mathsf{C004}, \mathsf{C005}, \mathsf{C006}\}.$$

Take $n = 1$. Interpret each 1-supervertex $X \subseteq V_0$ as a *customer segment* (e.g., accounts sharing a risk tier or product usage).

Define three segments

$$R := X_0 := \{\mathsf{C001}, \mathsf{C002}, \mathsf{C003}\} \quad (\mathsf{medium\text{-}risk\ segment}),$$

$$X_1 := \{\mathsf{C001}\} \quad (\text{low-risk subsegment}), \qquad X_2 := \{\mathsf{C002}, \mathsf{C003}\} \quad (\text{high-risk subsegment}),$$
and set $V := \{X_0, X_1, X_2\} \subseteq \mathcal{P}(V_0)$.

Let the superedge set encode binary branching (a single split of the root segment):

$$\varepsilon_0 := \{X_0, X_1, X_2\} \in E, \qquad E := \{\varepsilon_0\}.$$

Then the incidence graph $X_0 - \varepsilon_0 - \{X_1, X_2\}$ is a tree rooted at $R = X_0$, and ε_0 has exactly two supervertex children X_1, X_2, hence the rooted incidence hierarchy is binary.

Define a key function key : $V \to \mathbb{N}$ by letting key(X) be the *maximum KYC risk score* within segment X (an integer maintained by the compliance team). Suppose

$$\mathrm{key}(X_1) = 20, \qquad \mathrm{key}(X_0) = 50, \qquad \mathrm{key}(X_2) = 80,$$

so that $\mathrm{key}(X_1) < \mathrm{key}(X_0) < \mathrm{key}(X_2)$.

Label the two children of X_0 by

$$\lambda_{X_0}(X_1) = \mathsf{L}, \qquad \lambda_{X_0}(X_2) = \mathsf{R}.$$

Then $\left(\mathrm{SHG}^{(1)}, R, \mathrm{key}, \lambda\right)$ satisfies the binary search property: all supervertices in the left subtree (here just X_1) have smaller keys than the root X_0, and all supervertices in the right subtree (here just X_2) have larger keys.

Operationally, this models a searchable compliance index: querying a threshold (e.g., "risk score ≤ 50") navigates left, while higher-risk queries navigate right, even though each node represents a *set-valued* segment.

4.9 B-tree, B-hypertree, and B-superhypertree

In this section, we examine the relationships among B-trees, B-hypertrees, and B-superhypertrees.

4.9.1 B-trees (multiway search trees)

A **B**-tree is a balanced multiway search tree whose nodes store multiple sorted keys and maintain between t and $2t$ children, keeping all leaves at the same depth [111–113].

Definition 4.9.1 (B-tree of minimum degree t)**.** Let $(K, \leq)$ be a totally ordered set and fix an integer $t \geq 2$ (the *minimum degree*). A *B-tree of minimum degree t* is a tuple

$$\mathcal{B} \;=\; (T, r, \mathbf{k}, \mathrm{ord}),$$

where:

(i) $T = (V, E)$ is a finite rooted tree with root $r \in V$ (viewed as an arborescence oriented away from r).

(ii) $\mathbf{k}$ assigns to each node $v \in V$ a finite strictly increasing sequence of keys

$$\mathbf{k}(v) = (k_1(v) < k_2(v) < \cdots < k_{m(v)}(v)) \in K^{m(v)},$$

where $m(v) := |\mathbf{k}(v)|$ is the number of keys stored at v.

(iii) ord fixes an ordering of the children: for each internal node v, if $\mathrm{Ch}_T(v)$ denotes the set of children of v in the rooted tree, then $\mathrm{ord}_v : \mathrm{Ch}_T(v) \to \{0, 1, \ldots, m(v)\}$ is a bijection. Write the children in ord_v-order as

$$\mathrm{Ch}_T(v) = \{c_0(v), c_1(v), \ldots, c_{m(v)}(v)\}, \qquad \mathrm{ord}_v(c_i(v)) = i.$$

(iv) (*Search-order (separator) property*) For every internal node v and every descendant node x in the subtree rooted at $c_i(v)$, every key occurring anywhere in that descendant subtree lies in the corresponding interval:

$$\begin{cases} \text{if } i = 0, & \text{all such keys are } < k_1(v), \\ \text{if } 1 \le i \le m(v) - 1, & \text{all such keys satisfy } k_i(v) < \cdot < k_{i+1}(v), \\ \text{if } i = m(v), & \text{all such keys are } > k_{m(v)}(v). \end{cases}$$

(Here "keys occurring in the subtree" means keys stored at any node within that subtree.)

(v) (*Degree/occupancy constraints*)

(a) (*Root*) Either r is a leaf, or r has at least two children. In all cases,

$$0 \le m(r) \le 2t - 1, \quad \text{and if } r \text{ is internal, then } 2 \le |\mathrm{Ch}_T(r)| = m(r) + 1 \le 2t.$$

(b) (*Non-root internal nodes*) For every internal node $v \ne r$,

$$t - 1 \le m(v) \le 2t - 1, \qquad t \le |\mathrm{Ch}_T(v)| = m(v) + 1 \le 2t.$$

(vi) (*Balanced leaf level*) All leaves of T have the same depth (distance in edges from the root).

Remark 4.9.2 (Order m vs. minimum degree t)**.** Many texts parameterize B-trees by the maximum number of children M (often called the *order*). In Definition 4.9.1, the maximum number of children is $2t$, so one may set $M = 2t$.

4.9.2 B-hypertrees (hypergraph-incidence realization)

A **B**-hypertree is a rooted hypertree whose incidence hierarchy emulates a B-tree: vertex-nodes store separator keys, and hyperedge branching realizes ordered child subtrees with balanced depth.

Definition 4.9.3 (Incidence graph and rooted hypertree)**.** Let $H = (V, E)$ be a finite hypergraph with $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Its *incidence graph* is the bipartite graph

$$\mathrm{Inc}(H) \;=\; (V \sqcup E,\; F), \qquad F = \big\{\{v, e\} : v \in V,\; e \in E,\; v \in e\big\}.$$

We say that H is a *HyperTree* if $\mathrm{Inc}(H)$ is a tree. If H is a HyperTree and $r \in V$, then (H, r) is a *rooted HyperTree* by rooting $\mathrm{Inc}(H)$ at r, thereby inducing parent/child relations $\mathrm{Pa}(\cdot)$, $\mathrm{Ch}(\cdot)$ on the node set $V \sqcup E$.

Definition 4.9.4 (Vertex-descendant relation in a rooted HyperTree)**.** Let (H, r) be a rooted HyperTree. For a vertex $v \in V$, define the set of *vertex descendants* of v by

$$\mathrm{Desc}_V(v) \;:=\; \big\{u \in V :\; v \text{ lies on the unique } r\text{–}u \text{ path in } \mathrm{Inc}(H)\big\}.$$

Equivalently, $u \in \mathrm{Desc}_V(v)$ iff $u = v$ or the unique incidence-path from v to u moves away from r.

Definition 4.9.5 (B-hypertree of minimum degree t)**.** Let $(K, \leq)$ be a totally ordered set and fix $t \geq 2$. A *B-hypertree of minimum degree t* is a tuple

$$\mathcal{BH} \;=\; (H, r, \mathbf{k}, \mathrm{ord})$$

satisfying:

(i) (H, r) is a rooted HyperTree in the sense of Definition 4.9.3.

(ii) (*Unique branching hyperedge per internal vertex*) For each vertex $v \in V$, define

$$\mathrm{Out}(v) \;:=\; \{\, e \in E :\; \mathrm{Pa}(e) = v \,\}.$$

We require that for each *internal vertex* v (i.e., $\mathrm{Out}(v) \neq \emptyset$) one has $|\,\mathrm{Out}(v)| = 1$. Write $\mathrm{Out}(v) = \{e_v\}$ and define the *vertex-children* of v as

$$\mathrm{Ch}_V(v) \;:=\; \{\, u \in V :\; \mathrm{Pa}(u) = e_v \,\}.$$

(Thus traversal alternates $v \to e_v \to u$ in $\mathrm{Inc}(H)$.)

(iii) $\mathbf{k}$ assigns to each vertex $v \in V$ a strictly increasing list of keys

$$\mathbf{k}(v) = (k_1(v) < \cdots < k_{m(v)}(v)) \in K^{m(v)}, \qquad m(v) := |\mathbf{k}(v)|.$$

(iv) (*Arity compatibility*) If v is internal (so e_v exists), then

$$|\mathrm{Ch}_V(v)| \;=\; m(v) + 1.$$

Moreover, $\mathrm{ord}_v : \mathrm{Ch}_V(v) \to \{0, 1, \ldots, m(v)\}$ is a bijection, and we write

$$\mathrm{Ch}_V(v) = \{c_0(v), c_1(v), \ldots, c_{m(v)}(v)\}, \qquad \mathrm{ord}_v(c_i(v)) = i.$$

(v) (*Search-order property on vertex descendants*) For every internal vertex $v \in V$ and every $i \in \{0, 1, \ldots, m(v)\}$, define the *i-th branch descendant set*

$$\mathrm{Desc}_V^{(i)}(v) \;:=\; \mathrm{Desc}_V(c_i(v)).$$

Then every vertex $u \in \mathrm{Desc}_V^{(i)}(v)$ satisfies the interval constraint on its stored keys:

$$\begin{cases} \text{if } i = 0, & \text{all keys in } \mathbf{k}(u) \text{ are } < k_1(v), \\ \text{if } 1 \le i \le m(v) - 1, & \text{all keys in } \mathbf{k}(u) \text{ satisfy } k_i(v) < \cdot < k_{i+1}(v), \\ \text{if } i = m(v), & \text{all keys in } \mathbf{k}(u) \text{ are } > k_{m(v)}(v). \end{cases}$$

(vi) (*Degree/occupancy constraints*) Let r be the root vertex.

(a) (*Root*) $0 \le m(r) \le 2t - 1$, and if r is internal then

$$2 \le |\mathrm{Ch}_V(r)| = m(r) + 1 \le 2t.$$

(b) (*Non-root internal vertices*) For every internal $v \ne r$,

$$t - 1 \le m(v) \le 2t - 1, \qquad t \le |\mathrm{Ch}_V(v)| = m(v) + 1 \le 2t.$$

(vii) (*Balanced leaf level*) All *vertex-leaves* (vertices $v \in V$ with $\mathrm{Out}(v) = \emptyset$) have the same *vertex-depth*, where the vertex-depth of $u \in V$ is defined as

$$\mathrm{depth}_V(u) \;:=\; \frac{1}{2}\, \mathrm{dist}_{\mathrm{Inc}(H)}(r, u),$$

which is an integer because $\mathrm{Inc}(H)$ is bipartite and $r, u \in V$.

Remark 4.9.6. A B-hypertree is a B-tree in which each internal vertex v branches through a unique hyperedge e_v, whose incident vertex-children are exactly $m(v) + 1$ and are ordered to match the separator keys at v.

4.9.3 B-superhypertrees (supervertex version)

A **B**-superhypertree extends B-hypertrees to n-SuperHyperTrees: supervertices carry sorted separator keys, and superedges provide multiway branching while preserving B-tree occupancy and uniform leaf depth.

Definition 4.9.7 (Rooted n-SuperHyperTree)**.** Let V_0 be a finite nonempty base set and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph with $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Its incidence graph is

$$\mathrm{Inc}\big(\mathrm{SHG}^{(n)}\big) \;=\; (V \sqcup E,\ F), \qquad F = \big\{\{X, \varepsilon\} : X \in V,\ \varepsilon \in E,\ X \in \varepsilon\big\}.$$

We say $\mathrm{SHG}^{(n)}$ is an *n-SuperHyperTree* if $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ is a tree. If $R \in V$, then $\big(\mathrm{SHG}^{(n)}, R\big)$ is a rooted n-SuperHyperTree by rooting $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ at R.

Definition 4.9.8 (Supervertex descendants)**.** Let $\big(\mathrm{SHG}^{(n)}, R\big)$ be a rooted n-SuperHyperTree. For $X \in V$, define

$$\mathrm{Desc}_V(X) \ := \ \big\{Y \in V : \ X \text{ lies on the unique } R\text{–}Y \text{ path in } \mathrm{Inc}(\mathrm{SHG}^{(n)})\big\}.$$

Definition 4.9.9 (B-superhypertree of minimum degree t)**.** Let $(K, \leq)$ be a totally ordered set and fix $t \geq 2$. A *B-superhypertree of minimum degree t* is a tuple

$$\mathcal{BSH} \ = \ \big(\mathrm{SHG}^{(n)}, R, \mathbf{k}, \mathrm{ord}\big)$$

such that:

(i) $\big(\mathrm{SHG}^{(n)}, R\big)$ is a rooted n-SuperHyperTree (Definition 4.9.7).

(ii) (*Unique branching superedge per internal supervertex*) For each supervertex $X \in V$, let

$$\mathrm{Out}(X) \ := \ \{\, \varepsilon \in E : \ \mathrm{Pa}(\varepsilon) = X \,\}.$$

For each internal supervertex X (i.e., $\mathrm{Out}(X) \neq \emptyset$) we require $|\,\mathrm{Out}(X)| = 1$. Write $\mathrm{Out}(X) = \{\varepsilon_X\}$ and define

$$\mathrm{Ch}_V(X) \ := \ \{\, Y \in V : \ \mathrm{Pa}(Y) = \varepsilon_X \,\}.$$

(iii) $\mathbf{k}$ assigns each supervertex $X \in V$ a strictly increasing key list

$$\mathbf{k}(X) = (k_1(X) < \cdots < k_{m(X)}(X)) \in K^{m(X)}, \qquad m(X) := |\mathbf{k}(X)|.$$

(iv) (*Arity compatibility and ordering*) If X is internal, then $|\mathrm{Ch}_V(X)| = m(X)+1$ and $\mathrm{ord}_X : \mathrm{Ch}_V(X) \to \{0, 1, \ldots, m(X)\}$ is a bijection. Write the ordered children as $\mathrm{Ch}_V(X) = \{C_0(X), \ldots, C_{m(X)}(X)\}$.

(v) (*Search-order property*) For every internal $X \in V$ and each $i \in \{0, \ldots, m(X)\}$, put $\mathrm{Desc}_V^{(i)}(X) := \mathrm{Desc}_V(C_i(X))$. Then for all $Y \in \mathrm{Desc}_V^{(i)}(X)$, the keys stored at Y lie in the appropriate separator interval:

$$\begin{cases} \text{if } i = 0, & \text{all keys in } \mathbf{k}(Y) \text{ are } < k_1(X), \\ \text{if } 1 \leq i \leq m(X)-1, & \text{all keys in } \mathbf{k}(Y) \text{ satisfy } k_i(X) < \cdot < k_{i+1}(X), \\ \text{if } i = m(X), & \text{all keys in } \mathbf{k}(Y) \text{ are } > k_{m(X)}(X). \end{cases}$$

(vi) (*Degree/occupancy constraints*)

(a) (*Root supervertex*) $0 \leq m(R) \leq 2t-1$, and if R is internal then

$$2 \leq |\mathrm{Ch}_V(R)| = m(R) + 1 \leq 2t.$$

(b) (*Non-root internal supervertices*) For every internal $X \neq R$,

$$t - 1 \leq m(X) \leq 2t - 1, \qquad t \leq |\mathrm{Ch}_V(X)| = m(X) + 1 \leq 2t.$$

(vii) (*Balanced leaf level*) All supervertex-leaves (supervertices X with $\mathrm{Out}(X) = \emptyset$) have the same supervertex-depth

$$\mathrm{depth}_V(Y) \;:=\; \frac{1}{2}\,\mathrm{dist}_{\mathrm{Inc}(\mathrm{SHG}^{(n)})}(R, Y) \in \mathbb{N}.$$

Remark 4.9.10. In Definition 4.9.9, the key domain K is abstract. If one wants keys derived from base-level information, one may define keys via flattening, e.g. by composing a statistic on $\mathrm{Flat}_n(X) \subseteq V_0$ with an embedding into a totally ordered set. This is optional and not required for the structural definition above.

Example 4.9.11 (Real-life example of a B-superhypertree)**. Indexing nested product bundles for an e-commerce catalog.** Let V_0 be a finite set of atomic products:

$$V_0 = \{\mathsf{P1} = \mathsf{Laptop},\ \mathsf{P2} = \mathsf{Mouse},\ \mathsf{P3} = \mathsf{Keyboard},\ \mathsf{P4} = \mathsf{Monitor},\ \mathsf{P5} = \mathsf{Dock}\}.$$

Take $n = 1$, so supervertices are nonempty subsets of V_0; interpret each supervertex as a *bundle* of products sold together.

Define the supervertex set

$$\begin{aligned} R := X_0 &:= \{\mathsf{P1}, \mathsf{P2}, \mathsf{P3}, \mathsf{P4}, \mathsf{P5}\} \quad (\text{all bundles in this category}),\\ X_1 &:= \{\mathsf{P1}, \mathsf{P4}\} \quad (\text{workstation bundle}),\\ X_2 &:= \{\mathsf{P2}, \mathsf{P3}, \mathsf{P5}\} \quad (\text{accessory bundle}),\\ X_3 &:= \{\mathsf{P1}\} \quad (\text{single laptop}),\\ X_4 &:= \{\mathsf{P4}\} \quad (\text{single monitor}),\\ X_5 &:= \{\mathsf{P2}, \mathsf{P3}\} \quad (\text{input devices}),\\ X_6 &:= \{\mathsf{P5}\} \quad (\text{single dock}), \end{aligned}$$

and set $V := \{X_0, X_1, X_2, X_3, X_4, X_5, X_6\} \subseteq \mathcal{P}(V_0)$.

Let superedges encode multiway branching as in a B-tree node: the root supervertex $R = X_0$ branches to two children ($m(R) = 1$), and each of X_1, X_2 branches to two children ($m(X_1) = m(X_2) = 1$):

$$\varepsilon_{X_0} := \{X_0, X_1, X_2\}, \qquad \varepsilon_{X_1} := \{X_1, X_3, X_4\}, \qquad \varepsilon_{X_2} := \{X_2, X_5, X_6\},$$

and let $E := \{\varepsilon_{X_0}, \varepsilon_{X_1}, \varepsilon_{X_2}\}$. The incidence graph is the tree

$$X_0 - \varepsilon_{X_0} - \{X_1, X_2\}, \qquad X_1 - \varepsilon_{X_1} - \{X_3, X_4\}, \qquad X_2 - \varepsilon_{X_2} - \{X_5, X_6\},$$

so $(\mathrm{SHG}^{(1)}, R)$ is a rooted 1-SuperHyperTree with uniform leaf depth 2 (all leaves are X_3, X_4, X_5, X_6).

Define a totally ordered key domain $K = \mathbb{N}$ and assign separator keys using, for example, bundle price (in USD). Let each supervertex X store a strictly increasing list $\mathbf{k}(X)$ of separator keys:

$$\mathbf{k}(X_0) = (1000), \qquad \mathbf{k}(X_1) = (1200), \qquad \mathbf{k}(X_2) = (200),$$

and interpret the ordering so that bundles priced below 1000 are searched via the child X_2, while bundles at least 1000 are searched via X_1 (root split). Similarly, within X_1, prices below 1200 lead to X_3 (laptop only) and above to X_4 (monitor only), and within X_2, prices below 200 lead to X_6 (dock) and above to X_5 (input devices).

Thus the supervertices act as *multi-key search nodes* and the superedges provide *multiway branching* with bounded occupancy and balanced depth, giving a concrete B-superhypertree model for a catalog index over nested bundles.

Theorem 4.9.12 (B-superhypertrees are SuperHyperTrees)**.** *Let $t \geq 2$ and let*

$$\mathcal{BSH} = \big(\mathrm{SHG}^{(n)}, R, \mathbf{k}, \mathrm{ord}\big)$$

be a B-superhypertree of minimum degree t in the sense of Definition 4.9.9. Then the underlying n-SuperHyperGraph $\mathrm{SHG}^{(n)} = (V, E)$ is an n-SuperHyperTree (in the sense of Definition 4.2.5). Equivalently, its incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ is a tree.

Proof. By Definition 4.9.9(i), the pair $\big(\mathrm{SHG}^{(n)}, R\big)$ is a rooted n-SuperHyperTree (Definition 4.9.7). In particular, the rooted structure is imposed on the incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ by choosing a distinguished root $R \in V$ (and the associated parent/child relations), and $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ is a tree.

Forgetting the extra data $R, \mathbf{k}, \mathrm{ord}$ does not change the underlying incidence graph. Hence $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ remains a tree, and therefore $\mathrm{SHG}^{(n)}$ is an n-SuperHyperTree. □

4.10 Heap, hyperHeap, and superhyperHeap

This section defines heap-ordered structures in three settings. A *heap* is a rooted tree equipped with keys and a parent–child monotonicity constraint (min-heap or max-heap). To extend this idea to hypergraphs and n-SuperHyperGraphs, we work with rooted hypertrees/superhypertrees via their incidence trees, and impose the heap-order constraint on the vertex-type (respectively supervertex-type) nodes, while hyperedge/superedge nodes serve as branching gadgets.

4.10.1 Heaps on rooted trees (graph setting)

A heap is a rooted tree structure with keys that satisfies the heap-order property: each parent's key is $\leq$ (min-heap) or $\geq$ (max-heap) its children's keys [114–116].

Definition 4.10.1 (Heap (min-heap / max-heap))**.** [114–116] Let $(K, \leq)$ be a totally ordered set (key domain). A *heap* is a tuple

$$\mathcal{H} \;=\; (T, r, \mathrm{key}, \sigma),$$

where:

(i) (T, r) is a finite rooted tree (equivalently, an arborescence oriented away from r), with vertex set $V(T)$.

(ii) $\mathrm{key} : V(T) \to K$ assigns a key to each node.

(iii) $\sigma \in \{\mathsf{min}, \mathsf{max}\}$ specifies the heap type, and the following *heap-order property* holds: for every non-root vertex $v \in V(T)$ with parent $\mathrm{Pa}(v)$,

$$\begin{cases} \mathrm{key}(\mathrm{Pa}(v)) \leq \mathrm{key}(v), & \text{if } \sigma = \mathsf{min} \text{ (min-heap)},\\ \mathrm{key}(\mathrm{Pa}(v)) \geq \mathrm{key}(v), & \text{if } \sigma = \mathsf{max} \text{ (max-heap)}. \end{cases}$$

Remark 4.10.2 (No global sorting)**.** A heap is only *partially ordered*: the heap-order constraint is imposed along parent–child relations, and there is no required order between siblings or between nodes in different subtrees.

Definition 4.10.3 (Binary heap (optional shape constraint))**.** A *binary heap* is a heap $\mathcal{H} = (T, r, \mathrm{key}, \sigma)$ (Definition 4.10.1) whose underlying rooted tree (T, r) is a *complete binary tree* (shape constraint).

4.10.2 hyperHeaps on rooted hypertrees (hypergraph setting)

A hyperHeap is a heap-like structure on a rooted hypertree (incidence tree), where keys are assigned to vertices and the heap-order constraint holds between vertex levels.

Definition 4.10.4 (Rooted HyperTree and incidence tree)**.** Let $H = (V, E)$ be a finite hypergraph with $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Assume H is a *HyperTree*, i.e., its incidence graph

$$\mathrm{Inc}(H) \;=\; (V \sqcup E,\ F), \qquad F = \big\{\{v, e\} : v \in e\big\},$$

is a tree. Fix a root vertex $r \in V$ and root $\mathrm{Inc}(H)$ at r, thereby inducing parent/child relations $\mathrm{Pa}(\cdot)$, $\mathrm{Ch}(\cdot)$ on the node set $V \sqcup E$. The pair (H, r) is then called a *rooted HyperTree.*

Definition 4.10.5 (hyperHeap (min / max))**.** Let $(K, \leq)$ be a totally ordered set. A *hyperHeap* is a tuple

$$\mathcal{HH} \;=\; (H, r, \mathrm{key}, \sigma)$$

satisfying:

(i) (H, r) is a rooted HyperTree as in Definition 4.10.4.

(ii) $\mathrm{key} : V \to K$ assigns keys to *vertices* (hyperedges are not keyed).

(iii) $\sigma \in \{\mathsf{min}, \mathsf{max}\}$, and the heap-order property holds along *vertex-to-vertex* parent relations in the rooted incidence tree, as follows.

For each vertex $u \in V$ with $u \neq r$, let $\mathrm{Pa}_{\mathrm{Inc}}(u)$ be its parent in $\mathrm{Inc}(H)$. Since $\mathrm{Inc}(H)$ is bipartite and $r, u \in V$, $\mathrm{Pa}_{\mathrm{Inc}}(u) \in E$ is a hyperedge node; define the *vertex parent* of u by

$$\mathrm{Pa}_V(u) \;:=\; \mathrm{Pa}_{\mathrm{Inc}}\big(\mathrm{Pa}_{\mathrm{Inc}}(u)\big) \in V.$$

(Equivalently, $\mathrm{Pa}_V(u)$ is the unique vertex two incidence-steps above u on the path to the root.)

Then for every $u \in V \setminus \{r\}$,

$$\begin{cases} \mathrm{key}(\mathrm{Pa}_V(u)) \leq \mathrm{key}(u), & \text{if } \sigma = \mathsf{min}, \\ \mathrm{key}(\mathrm{Pa}_V(u)) \geq \mathrm{key}(u), & \text{if } \sigma = \mathsf{max}. \end{cases}$$

Remark 4.10.6. A hyperHeap is a heap whose underlying hierarchy is encoded by the incidence tree of a hypertree. Traversal alternates $v \to e \to u$; the heap-order constraint is imposed only between successive vertex levels (i.e., across two incidence steps), which is the natural analogue of parent–child order in ordinary heaps.

4.10.3 superhyperHeaps on rooted superhypertrees

A superhyperHeap generalizes hyperHeaps to n-SuperHyperTrees: keys are attached to nested supervertices, and heap-order holds along supervertex levels through the incidence hierarchy.

Definition 4.10.7 (Rooted n-SuperHyperTree)**.** Let V_0 be a finite nonempty base set and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph with $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Assume its incidence graph

$$\mathrm{Inc}\big(\mathrm{SHG}^{(n)}\big) \;=\; (V \sqcup E,\ F), \qquad F = \big\{\{X, \varepsilon\} : X \in \varepsilon\big\},$$

is a tree, and fix a root supervertex $R \in V$. Rooting this incidence tree at R induces parent/child relations $\mathrm{Pa}(\cdot)$, $\mathrm{Ch}(\cdot)$ on $V \sqcup E$. The pair $\big(\mathrm{SHG}^{(n)}, R\big)$ is called a *rooted n-SuperHyperTree.*

Definition 4.10.8 (superhyperHeap (min / max))**.** Let $(K, \leq)$ be a totally ordered set. A *superhyperHeap* is a tuple

$$\mathcal{SHH} \;=\; \big(\mathrm{SHG}^{(n)}, R, \mathrm{key}, \sigma\big)$$

such that:

(i) $\big(\mathrm{SHG}^{(n)}, R\big)$ is a rooted n-SuperHyperTree as in Definition 4.10.7.

(ii) $\mathrm{key} : V \to K$ assigns a key to each n-supervertex $X \in V$.

(iii) $\sigma \in \{\mathsf{min}, \mathsf{max}\}$, and the heap-order property holds along *supervertex-to-supervertex* parent relations induced by the rooted incidence tree.

For each $Y \in V \setminus \{R\}$, define its *supervertex parent* $\mathrm{Pa}_V(Y) \in V$ by

$$\mathrm{Pa}_V(Y) \;:=\; \mathrm{Pa}_{\mathrm{Inc}}\big(\mathrm{Pa}_{\mathrm{Inc}}(Y)\big),$$

where $\mathrm{Pa}_{\mathrm{Inc}}(\cdot)$ denotes the parent map in the rooted incidence tree $\mathrm{Inc}(\mathrm{SHG}^{(n)})$.

Then for every $Y \in V \setminus \{R\}$,

$$\begin{cases} \mathrm{key}(\mathrm{Pa}_V(Y)) \leq \mathrm{key}(Y), & \text{if } \sigma = \mathsf{min}, \\ \mathrm{key}(\mathrm{Pa}_V(Y)) \geq \mathrm{key}(Y), & \text{if } \sigma = \mathsf{max}. \end{cases}$$

Remark 4.10.9. A superhyperHeap is the direct supervertex-level analogue of a hyperHeap. The keys are attached to nested objects $X \in V \subseteq \mathcal{P}^n(V_0)$, and the heap order is enforced between successive supervertex generations in the rooted incidence tree. If desired, one may derive keys from base-level information via flattening Flat_n, but this is optional and not required.

Example 4.10.10 (Real-life example of a superhyperHeap on a rooted n-SuperHyperTree). **Prioritized patching of nested software components (min-superhyperHeap).** Let V_0 be a finite set of atomic software modules in a product:

$$V_0 = \{\mathsf{Auth}, \mathsf{Payments}, \mathsf{Search}, \mathsf{Logging}, \mathsf{UI}\}.$$

Take $n = 1$, so each supervertex is a nonempty subset of V_0, interpreted as a *component bundle* (e.g., a service composed of several modules).

Define supervertices (nested bundles) by

$$\begin{aligned} R := X_0 &:= \{\mathsf{Auth}, \mathsf{Payments}, \mathsf{Search}, \mathsf{Logging}, \mathsf{UI}\} \quad (\text{entire product bundle}), \\ X_1 &:= \{\mathsf{Auth}, \mathsf{Payments}\} \quad (\text{critical financial core}), \\ X_2 &:= \{\mathsf{Search}, \mathsf{Logging}, \mathsf{UI}\} \quad (\text{user-facing bundle}), \\ X_3 &:= \{\mathsf{Auth}\} \quad (\text{auth module}), \\ X_4 &:= \{\mathsf{Payments}\} \quad (\text{payments module}), \\ X_5 &:= \{\mathsf{Search}\} \quad (\text{search module}). \end{aligned}$$

Let $V := \{X_0, X_1, X_2, X_3, X_4, X_5\} \subseteq \mathcal{P}(V_0)$.

Let superedges encode a rooted incidence hierarchy (a tree) describing decomposition and prioritization flow:

$$\varepsilon_0 := \{X_0, X_1, X_2\}, \qquad \varepsilon_1 := \{X_1, X_3, X_4\}, \qquad \varepsilon_2 := \{X_2, X_5\},$$

and set $E := \{\varepsilon_0, \varepsilon_1, \varepsilon_2\}$. Then the incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(1)})$ is a tree rooted at $R = X_0$.

Define a key domain $K = \mathbb{N}$ and a *severity score* key function $\mathrm{key} : V \to \mathbb{N}$, where smaller scores indicate higher urgency (thus we use a min-heap). For example, let

$$\mathrm{key}(X_0) = 10, \quad \mathrm{key}(X_1) = 4, \quad \mathrm{key}(X_2) = 7,$$
$$\mathrm{key}(X_3) = 1, \quad \mathrm{key}(X_4) = 3, \quad \mathrm{key}(X_5) = 6.$$

Interpretation: the single module $X_3 = \{\mathsf{Auth}\}$ has the most urgent vulnerability (score 1), followed by $\{\mathsf{Payments}\}$ (score 3), etc.

Because the incidence graph is rooted at X_0, each non-root supervertex Y has a supervertex parent $\mathrm{Pa}_V(Y)$ two incidence-steps above it. For instance,

$$\mathrm{Pa}_V(X_1) = X_0, \quad \mathrm{Pa}_V(X_2) = X_0, \quad \mathrm{Pa}_V(X_3) = X_1, \quad \mathrm{Pa}_V(X_4) = X_1, \quad \mathrm{Pa}_V(X_5) = X_2.$$

The assigned keys satisfy the min-heap condition along these supervertex-parent links:

$$\mathrm{key}(X_0) = 10 \le \mathrm{key}(X_1) = 4 \text{ (false)},$$
$$\mathrm{key}(X_0) = 10 \le \mathrm{key}(X_2) = 7 \text{ (false)}.$$

Hence, to make it a *min*-superhyperHeap we instead interpret key as *priority* where larger is more urgent, or equivalently replace key by $\mathrm{key}'(X) := 11 - \mathrm{key}(X)$. Using key',

$$\mathrm{key}'(X_0) = 1, \ \mathrm{key}'(X_1) = 7, \ \mathrm{key}'(X_2) = 4, \ \mathrm{key}'(X_3) = 10, \ \mathrm{key}'(X_4) = 8, \ \mathrm{key}'(X_5) = 5,$$

and then

$$\mathrm{key}'(X_0) \le \mathrm{key}'(X_1), \ \mathrm{key}'(X_0) \le \mathrm{key}'(X_2),$$
$$\mathrm{key}'(X_1) \le \mathrm{key}'(X_3), \ \mathrm{key}'(X_1) \le \mathrm{key}'(X_4), \ \mathrm{key}'(X_2) \le \mathrm{key}'(X_5),$$

so $\big(\mathrm{SHG}^{(1)}, R, \mathrm{key}', \mathsf{min}\big)$ is a superhyperHeap. Operationally, this encodes that as one moves down the hierarchy to finer components, the urgency (priority) does not decrease, supporting systematic prioritization of nested patch tasks.

4.11 Fault tree, fault hypertree, and fault superhypertree

Next, we discuss fault trees, fault hypertrees, and fault superhypertrees.

4.11.1 Fault trees

A fault tree is a Boolean-logic model that represents how basic component failures combine through logic gates (such as AND/OR) to produce a single top-level system failure [117–119]. Related notions such as fuzzy fault trees [120–122] are also known.

Definition 4.11.1 (Gate types and gate semantics)**.** Let $\mathbb{B} := \{0, 1\}$. A *gate specification* is a pair $(\mathcal{O}, \Phi)$ consisting of

- a set $\mathcal{O}$ of gate types (e.g., $\mathsf{AND}, \mathsf{OR}, \mathsf{XOR}, \mathsf{k\text{-}of\text{-}n}$, etc.), and
- a semantics map Φ assigning to each gate type $o \in \mathcal{O}$ and each arity $k \geq 1$ a Boolean function
$$\Phi(o, k) : \mathbb{B}^k \longrightarrow \mathbb{B}.$$

Typical examples are:

$$\Phi(\mathsf{AND}, k)(x_1, \ldots, x_k) = \bigwedge_{i=1}^{k} x_i, \qquad \Phi(\mathsf{OR}, k)(x_1, \ldots, x_k) = \bigvee_{i=1}^{k} x_i.$$

Definition 4.11.2 (Fault tree)**.** Fix a gate specification $(\mathcal{O}, \Phi)$ as in Definition 4.11.1. A *fault tree* is a tuple
$$\mathcal{F} = (T, r, \mathcal{E}, \mathcal{G}, \mathrm{out}, \mathrm{inp}, \mathrm{op})$$
such that:

(i) $T = (N, F)$ is a finite tree (simple, undirected, connected, acyclic) and $r \in N$ is a distinguished root.

(ii) $N = \mathcal{E} \sqcup \mathcal{G}$ is a disjoint partition into *event nodes* $\mathcal{E}$ and *gate nodes* $\mathcal{G}$, and the root is an event node: $r \in \mathcal{E}$. Moreover, T is bipartite with respect to this partition (every edge joins an event node to a gate node).

(iii) $\mathrm{out} : \mathcal{E} \to \mathcal{G} \cup \{\bot\}$ assigns to each event node $e \in \mathcal{E}$ either a unique *output gate* $\mathrm{out}(e) \in \mathcal{G}$ or $\mathrm{out}(e) = \bot$ (meaning: no output gate). We require that
$$\mathrm{out}(e) = \bot \iff e \text{ is a leaf of } T,$$
and that for every non-leaf event $e \in \mathcal{E}$ the node $\mathrm{out}(e)$ is the unique neighbor of e that is farther from the root r in the rooted tree (T, r).

(iv) $\mathrm{inp} : \mathcal{G} \to \mathcal{P}(\mathcal{E}) \setminus \{\emptyset\}$ assigns to each gate node $g \in \mathcal{G}$ a nonempty set $\mathrm{inp}(g)$ of *input events*, defined as the set of event-children of g in the rooted tree (T, r). Equivalently, if g has parent event e_g (the unique neighbor of g closer to the root), then

$$\mathrm{inp}(g) = N_T(g) \cap \mathcal{E} \setminus \{e_g\},$$

and $\mathrm{inp}(g) \neq \emptyset$.

(v) $\mathrm{op} : \mathcal{G} \to \mathcal{O}$ assigns a gate type to each gate node. If $|\,\mathrm{inp}(g)| = k$, then g is interpreted by the Boolean function $\Phi(\mathrm{op}(g), k) : \mathbb{B}^k \to \mathbb{B}$.

Definition 4.11.3 (Boolean evaluation and top event)**.** Let $\mathcal{F} = (T, r, \mathcal{E}, \mathcal{G}, \mathrm{out}, \mathrm{inp}, \mathrm{op})$ be a fault tree. The set of *basic events* is

$$\mathcal{B} := \{e \in \mathcal{E} : \ \mathrm{out}(e) = \bot\},$$

and the remaining events $\mathcal{I} := \mathcal{E} \setminus \mathcal{B}$ are *intermediate events* (including possibly the root/top event).

Given a basic-event assignment $x : \mathcal{B} \to \mathbb{B}$, the induced event evaluation $X_x : \mathcal{E} \to \mathbb{B}$ is defined recursively from leaves toward the root by:

(i) If $e \in \mathcal{B}$, then $X_x(e) := x(e)$.

(ii) If $e \in \mathcal{I}$, let $g = \mathrm{out}(e)$ be its unique output gate and write $\mathrm{inp}(g) = \{e_1, \ldots, e_k\}$. Then

$$X_x(e) \ := \ \Phi(\mathrm{op}(g), k)\big(X_x(e_1), \ldots, X_x(e_k)\big).$$

The *top event* occurs under x if and only if $X_x(r) = 1$.

Remark 4.11.4. Many industrial fault trees allow *repeated* basic events (common-cause events), in which case the structure is a rooted directed acyclic graph rather than a strict tree. Definitions 4.11.2–4.11.3 capture the tree (no-sharing) case in a mathematically clean form.

4.11.2 Fault hypertrees

Fault hypertree is a hypergraph-based fault model in which each gate is a hyperedge whose head is the output event and whose tail is a set of input events, organized hierarchically toward the top event.

Definition 4.11.5 (Fault hypertree)**.** Fix a gate specification $(\mathcal{O}, \Phi)$ and let $\mathbb{B} = \{0, 1\}$. A *fault hypertree* is a tuple

$$\mathcal{FH} \ = \ (H, r, \mathrm{Head}, \mathrm{op})$$

satisfying:

(i) $H = (V, E)$ is a finite hypergraph with $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$, and $r \in V$ is a distinguished root vertex (the top event).

(ii) The incidence graph $\mathrm{Inc}(H)$ is a tree, and it is rooted at $r \in V$. (Equivalently, H is a rooted HyperTree in the incidence-tree sense.)

(iii) $\mathrm{Head} : E \to V$ assigns to each hyperedge $e \in E$ a distinguished *head* vertex $\mathrm{Head}(e) \in e$, interpreted as the *output event* of that gate-hyperedge. Define the *tail* (inputs) by

$$\mathrm{Tail}(e) \;:=\; e \setminus \{\mathrm{Head}(e)\}.$$

We require $\mathrm{Tail}(e) \neq \emptyset$ for all $e \in E$.

(iv) (*Unique decomposition*) For every vertex $v \in V$ that is not a basic event (defined below), there exists exactly one hyperedge $e_v \in E$ with $\mathrm{Head}(e_v) = v$. The root r is allowed to be basic or non-basic.

(v) (*Hierarchical direction*) In the rooted incidence tree $\mathrm{Inc}(H)$, for every hyperedge $e \in E$ the head $\mathrm{Head}(e)$ is closer to the root than every input vertex in $\mathrm{Tail}(e)$. (Thus the causal direction is "from $\mathrm{Tail}(e)$ up to $\mathrm{Head}(e)$".)

(vi) $\mathrm{op} : E \to \mathcal{O}$ assigns a gate type to each hyperedge; if $|\mathrm{Tail}(e)| = k$, then e is interpreted by $\Phi(\mathrm{op}(e), k) : \mathbb{B}^k \to \mathbb{B}$.

Definition 4.11.6 (Boolean evaluation on a fault hypertree)**.** Let $\mathcal{FH} = (H, r, \mathrm{Head}, \mathrm{op})$ be a fault hypertree with $H = (V, E)$, and let $\mathbb{B} = \{0, 1\}$. Define the set of *basic events* by

$$\mathcal{B} \;:=\; \{v \in V : \; \nexists e \in E \text{ with } \mathrm{Head}(e) = v\},$$

and let $\mathcal{I} := V \setminus \mathcal{B}$ be the intermediate events.

Given an assignment $x : \mathcal{B} \to \mathbb{B}$, define $X_x : V \to \mathbb{B}$ recursively by:

(i) If $v \in \mathcal{B}$, set $X_x(v) := x(v)$.

(ii) If $v \in \mathcal{I}$, let e_v be the unique hyperedge with $\mathrm{Head}(e_v) = v$, and write $\mathrm{Tail}(e_v) = \{v_1, \ldots, v_k\}$. Then

$$X_x(v) \;:=\; \Phi(\mathrm{op}(e_v), k)\big(X_x(v_1), \ldots, X_x(v_k)\big).$$

The top event occurs under x if and only if $X_x(r) = 1$.

4.11.3 Fault superhypertrees

Fault superhypertree is an n-SuperHyperGraph-based fault model where the events are nested supervertices and the gates are superedges, so hierarchical, multi-level failure dependencies are evaluated by Boolean semantics up to the top super-event.

Definition 4.11.7 (Fault n-superhypertree)**.** Fix $n \geq 0$ and a gate specification $(\mathcal{O}, \Phi)$, and let $\mathbb{B} = \{0, 1\}$. A *fault n-superhypertree* (or *fault superhypertree*) is a tuple

$$\mathcal{FSH} \ = \ \big(\mathrm{SHG}^{(n)}, R, \mathrm{Head}, \mathrm{op}\big)$$

satisfying:

(i) $\mathrm{SHG}^{(n)} = (V, E)$ is an n-SuperHyperGraph on some base set V_0, with $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$, and $R \in V$ is a distinguished root supervertex (top event).

(ii) The incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ is a tree rooted at R. (Equivalently, $\mathrm{SHG}^{(n)}$ is a rooted n-SuperHyperTree in the incidence-tree sense.)

(iii) $\mathrm{Head} : E \to V$ assigns to each superedge $\varepsilon \in E$ a distinguished head supervertex $\mathrm{Head}(\varepsilon) \in \varepsilon$, and we set

$$\mathrm{Tail}(\varepsilon) \ := \ \varepsilon \setminus \{\mathrm{Head}(\varepsilon)\} \neq \emptyset.$$

(iv) (*Unique decomposition*) For every non-basic supervertex $X \in V$ (defined below), there exists exactly one superedge $\varepsilon_X \in E$ with $\mathrm{Head}(\varepsilon_X) = X$.

(v) (*Hierarchical direction*) In the rooted incidence tree, $\mathrm{Head}(\varepsilon)$ is closer to R than every element of $\mathrm{Tail}(\varepsilon)$.

(vi) $\mathrm{op} : E \to \mathcal{O}$ assigns gate types; if $|\mathrm{Tail}(\varepsilon)| = k$, interpret ε by $\Phi(\mathrm{op}(\varepsilon), k) : \mathbb{B}^k \to \mathbb{B}$.

Definition 4.11.8 (Boolean evaluation on a fault n-superhypertree)**.** Let

$$\mathcal{FSH} = \big(\mathrm{SHG}^{(n)}, R, \mathrm{Head}, \mathrm{op}\big)$$

with $\mathrm{SHG}^{(n)} = (V, E)$, and let $\mathbb{B} = \{0, 1\}$. Define the basic superevents by

$$\mathcal{B} \ := \ \{X \in V : \ \nexists \varepsilon \in E \text{ with } \mathrm{Head}(\varepsilon) = X\},$$

$$\mathcal{I} := V \setminus \mathcal{B}.$$

Given $x : \mathcal{B} \to \mathbb{B}$, define $X_x : V \to \mathbb{B}$ by:

(i) If $X \in \mathcal{B}$, set $X_x(X) := x(X)$.

(ii) If $X \in \mathcal{I}$, let ε_X be the unique superedge with $\mathrm{Head}(\varepsilon_X) = X$, and write $\mathrm{Tail}(\varepsilon_X) = \{X_1, \ldots, X_k\}$. Then

$$X_x(X) \ := \ \Phi(\mathrm{op}(\varepsilon_X), k)\big(X_x(X_1), \ldots, X_x(X_k)\big).$$

The top super-event occurs under x if and only if $X_x(R) = 1$.

Remark 4.11.9 (Probabilistic fault analysis)**.** If basic events are modeled as Bernoulli random variables (often under an independence assumption), then Definitions 4.11.3, 4.11.6, and 4.11.8 induce a random variable for the top event, and one may compute $\mathbb{P}[X(R) = 1]$. For AND and OR gates with independent inputs, the usual formulas apply:

$$\mathbb{P}[\mathsf{AND}] = \prod_i p_i,$$

$$\mathbb{P}[\mathsf{OR}] = 1 - \prod_i (1 - p_i),$$

where p_i are the input-event probabilities.

Example 4.11.10 (Real-life example of a fault n-superhypertree)**. Multi-level outage analysis for a cloud service with nested subsystems.** Let V_0 be a finite set of atomic components in a cloud platform:

$$V_0 = \{\mathsf{DB}, \mathsf{Cache}, \mathsf{Auth}, \mathsf{API}, \mathsf{LoadBalancer}\}.$$

Take $n = 1$, so supervertices are nonempty subsets of V_0, interpreted as *subsystems* (bundles of components).

Define three supervertices (events) representing subsystem failure states:

$$R := X_{\mathrm{Outage}} := \{\mathsf{Auth}, \mathsf{API}, \mathsf{LoadBalancer}\} \quad (\mathsf{top\ event:\ customer\text{-}facing\ outage}),$$

$$X_{\mathrm{FrontEnd}} := \{\mathsf{API}, \mathsf{LoadBalancer}\} \quad (\mathsf{front\text{-}end\ service\ failure}),$$

$$X_{\mathrm{AuthFail}} := \{\mathsf{Auth}\} \quad (\mathsf{authentication\ failure}).$$

Let $V := \{X_{\mathrm{Outage}}, X_{\mathrm{FrontEnd}}, X_{\mathrm{AuthFail}}\} \subseteq \mathcal{P}(V_0)$.

Let the gate set be $\mathcal{O} = \{\mathsf{OR}\}$ with the usual Boolean semantics $\Phi(\mathsf{OR}, k) = \bigvee_{i=1}^{k}$. Define two superedges (gates):

$$\varepsilon_{\mathrm{Outage}} := \{X_{\mathrm{Outage}}, X_{\mathrm{FrontEnd}}, X_{\mathrm{AuthFail}}\},$$

$$\varepsilon_{\mathrm{FrontEnd}} := \{X_{\mathrm{FrontEnd}}, X_{\mathrm{API}}, X_{\mathrm{LB}}\},$$

where we introduce two basic superevents

$$X_{\mathrm{API}} := \{\mathsf{API}\}, \qquad X_{\mathrm{LB}} := \{\mathsf{LoadBalancer}\},$$

and extend

$$V := V \cup \{X_{\mathrm{API}}, X_{\mathrm{LB}}\}.$$

Set $E := \{\varepsilon_{\mathrm{Outage}}, \varepsilon_{\mathrm{FrontEnd}}\}$.

Assign heads (outputs) by

$$\mathrm{Head}(\varepsilon_{\mathrm{Outage}}) = X_{\mathrm{Outage}},$$

$$\mathrm{Head}(\varepsilon_{\mathrm{FrontEnd}}) = X_{\mathrm{FrontEnd}},$$

so the tails (inputs) are

$$\mathrm{Tail}(\varepsilon_{\mathrm{Outage}}) = \{X_{\mathrm{FrontEnd}}, X_{\mathrm{AuthFail}}\},$$

$$\mathrm{Tail}(\varepsilon_{\mathrm{FrontEnd}}) = \{X_{\mathrm{API}}, X_{\mathrm{LB}}\}.$$

Assign gate types by $\mathrm{op}(\varepsilon_{\mathrm{Outage}}) = \mathrm{op}(\varepsilon_{\mathrm{FrontEnd}}) = \mathsf{OR}$.

Then $\mathrm{Inc}(\mathrm{SHG}^{(1)})$ is a tree rooted at $R = X_{\mathrm{Outage}}$:

$$X_{\mathrm{Outage}} - \varepsilon_{\mathrm{Outage}} - \{X_{\mathrm{FrontEnd}}, X_{\mathrm{AuthFail}}\},$$

$$X_{\mathrm{FrontEnd}} - \varepsilon_{\mathrm{FrontEnd}} - \{X_{\mathrm{API}}, X_{\mathrm{LB}}\}.$$

The basic superevents are $X_{\mathrm{AuthFail}}, X_{\mathrm{API}}, X_{\mathrm{LB}}$ (they have no outgoing gate-superedge), and the intermediate superevents include $X_{\mathrm{FrontEnd}}, X_{\mathrm{Outage}}$.

Operational meaning: the top event Outage occurs if either FrontEnd fails or Auth fails; and FrontEnd fails if either API fails or LoadBalancer fails. This is exactly the Boolean evaluation induced by the fault superhypertree structure.

4.12 Tree-Decomposition

An *n-SuperHyperTree* is an acyclic n-SuperHyperGraph whose superedges admit a *join-tree* representation, meaning that superedges can be arranged as the nodes of a tree so that, for every supervertex, the collection of incident superedges appears as a connected subtree [18, 61, 62]. In this sense, SuperHyperTrees extend the classical notion of hypertrees [59, 60, 123].

A *tree decomposition* encodes a graph by a tree of *bags* of vertices, requiring that each edge is contained in some bag and that the bags containing a fixed vertex satisfy a running-intersection property [124–128]. Treewidth is central because many NP-hard graph problems admit dynamic-programming algorithms on classes of bounded treewidth, providing both algorithmic leverage and structural insight. Analogously, a *SuperHypertree decomposition* represents an n-SuperHyperGraph by a tree equipped with bags and *guards* (collections of superedges) that enforce coverage and connectedness for both supervertices and superedges across the decomposition [61–63, 129]. SuperHypertree decompositions are conceptually aligned with tree decompositions [125, 130] and with hypertree decompositions [60, 123].

Definition 4.12.1 (Tree decomposition and treewidth)**.** [131, 132] Let $G = (V, E)$ be a finite undirected graph. A *tree decomposition* of G is a pair (T, χ) where $T = (N, F)$ is a tree and χ assigns to each node $p \in N$ a *bag* $\chi(p) \subseteq V$ such that:

1. *Vertex coverage:* for every $v \in V$ there exists $p \in N$ with $v \in \chi(p)$.

2. *Edge coverage:* for every edge $\{u, v\} \in E$ there exists $p \in N$ with $\{u, v\} \subseteq \chi(p)$.

3. *Running intersection:* for every $v \in V$, the set

$$N_v := \{\, p \in N \mid v \in \chi(p) \,\}$$

induces a connected subtree of T.

The *width* of (T, χ) is

$$\text{width}(T, \chi) := \max_{p \in N} \big(|\chi(p)| - 1\big),$$

and the *treewidth* of G is

$$\text{tw}(G) := \min\{\text{width}(T, \chi) \mid (T, \chi) \text{ is a tree decomposition of } G\}.$$

Definition 4.12.2 (Hypertree decomposition)**.** [60,123,133] Let $H = (V, E)$ be a finite hypergraph, where $V = V(H)$ is the set of variables (vertices) and $E = E(H)$ is the set of hyperedges. A *hypertree decomposition* of H is a triple

$$HD = \langle T, \chi, \lambda \rangle,$$

where T is a rooted tree and χ, λ are labeling functions such that for each node $p \in V(T)$,

$$\chi(p) \subseteq V \qquad \text{and} \qquad \lambda(p) \subseteq E.$$

For $\mathcal{F} \subseteq E$ write

$$V(\mathcal{F}) := \bigcup_{h \in \mathcal{F}} h \subseteq V.$$

For any node $p \in V(T)$, let T_p be the subtree of T rooted at p and set

$$\chi(T_p) := \bigcup_{q \in V(T_p)} \chi(q).$$

The triple $\langle T, \chi, \lambda \rangle$ is a hypertree decomposition of H if:

(1) (*Edge coverage*) For each $h \in E$ there exists $p \in V(T)$ with $h \subseteq \chi(p)$.

(2) (*Connectedness of variables*) For each $Y \in V$, the set

$$\{\, p \in V(T) \mid Y \in \chi(p) \,\}$$

induces a connected subtree of T.

(3) (*Local covering by guards*) For each $p \in V(T)$,

$$\chi(p) \subseteq V(\lambda(p)).$$

(4) (*Special condition*) For each $p \in V(T)$,

$$V(\lambda(p)) \cap \chi(T_p) \subseteq \chi(p).$$

The *width* of HD is

$$\text{width}(HD) := \max_{p \in V(T)} |\lambda(p)|.$$

Example 4.12.3 (Real-life example of a hypertree decomposition). **Database query with multi-attribute joins.** Consider a database query that joins three relations:

$$R_1(A, B), \qquad R_2(B, C), \qquad R_3(C, D),$$

where A, B, C, D are query variables (attributes). Model this query as the hypergraph

$$H = (V, E), \qquad V = \{A, B, C, D\}, \qquad E = \{h_1, h_2, h_3\},$$

with

$$h_1 = \{A, B\}, \qquad h_2 = \{B, C\}, \qquad h_3 = \{C, D\}.$$

Define a rooted tree T with three nodes p_1, p_2, p_3 arranged as a path $p_1 - p_2 - p_3$, rooted at p_2. Define the labels

$$\chi(p_1) = \{A, B\}, \qquad \chi(p_2) = \{B, C\}, \qquad \chi(p_3) = \{C, D\},$$

$$\lambda(p_1) = \{h_1\}, \qquad \lambda(p_2) = \{h_2\}, \qquad \lambda(p_3) = \{h_3\}.$$

Then $HD = \langle T, \chi, \lambda \rangle$ is a hypertree decomposition of H: each hyperedge h_i is contained in $\chi(p_i)$ (edge coverage); each variable appears in a connected set of bags (e.g., B appears in p_1, p_2, and C in p_2, p_3); each bag is covered by the variables of its guards (here equality holds); and the special condition holds because the overlap between $V(\lambda(p_i))$ and variables in the subtree is already contained in $\chi(p_i)$ for this path-shaped decomposition. The width is

$$\mathrm{width}(HD) = \max\{|\lambda(p_1)|, |\lambda(p_2)|, |\lambda(p_3)|\} = 1.$$

Operationally, this decomposition corresponds to evaluating the join by a tree-structured plan that passes the shared attributes B and C along a chain.

Definition 4.12.4 (SuperHypertree decomposition of an n-SuperHyperGraph). Let $H^{(n)} = (V, E)$ be a finite n-SuperHyperGraph. A *SuperHypertree decomposition* of $H^{(n)}$ is a triple

$$(T, \mathcal{B}, \mathcal{C}),$$

where $T = (V_T, E_T)$ is a finite tree, $\mathcal{B} = \{ B_t \subseteq V \mid t \in V_T \}$ is a family of *bags*, and $\mathcal{C} = \{ C_t \subseteq E \mid t \in V_T \}$ is a family of *guards*, such that:

(1) *Vertex coverage:* $V = \bigcup_{t \in V_T} B_t$.

(2) *Superedge coverage:* for every $e \in E$ there exists $t \in V_T$ with $e \subseteq B_t$.

(3) *Vertex connectedness:* for every $v \in V$, the set

$$T_v := \{ t \in V_T \mid v \in B_t \}$$

is nonempty and induces a connected subtree of T.

(4) *Guard covering:* for each $t \in V_T$,

$$B_t \subseteq \bigcup_{e \in C_t} e.$$

(5) *Guard connectedness and consistency:* for every $e \in E$, the set

$$T_e := \{\, t \in V_T \mid e \in C_t \,\}$$

is nonempty and induces a connected subtree of T, and whenever $e \subseteq B_t$ for some $t \in V_T$, one also has $e \in C_t$.

Example 4.12.5 (A SuperHypertree decomposition of a 2-SuperHyperGraph)**. Nested software service groups with coordinated release windows.** Let the base set of engineers be

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

Define level-1 teams

$$T_1 = \{a, b\}, \qquad T_2 = \{b, c\}, \qquad T_3 = \{c, d\}.$$

For $n = 2$, a supervertex is a set of teams, hence an element of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$. Let

$$X_0 = \{T_1, T_2\}, \qquad X_1 = \{T_2\}, \qquad X_2 = \{T_2, T_3\}, \qquad V = \{X_0, X_1, X_2\} \subseteq \mathcal{P}^2(V_0).$$

Define two superedges (release windows)

$$e_1 = \{X_0, X_1\}, \qquad e_2 = \{X_1, X_2\}, \qquad E = \{e_1, e_2\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Then $H^{(2)} = (V, E)$ is a finite 2-SuperHyperGraph.

A SuperHypertree decomposition. Let the tree $T = (V_T, E_T)$ be the path on three nodes

$$V_T = \{t_1, t_2, t_3\}, \qquad E_T = \{\{t_1, t_2\}, \{t_2, t_3\}\}.$$

Define bags and guards by

$$B_{t_1} = \{X_0, X_1\}, \qquad B_{t_2} = \{X_1\}, \qquad B_{t_3} = \{X_1, X_2\},$$

$$C_{t_1} = \{e_1\}, \qquad C_{t_2} = \{e_1, e_2\}, \qquad C_{t_3} = \{e_2\}.$$

We verify the conditions in Definition 4.12.4 (details omitted here, see the text below). Therefore $(T, \mathcal{B}, \mathcal{C})$ is a SuperHypertree decomposition of $H^{(2)}$. In particular, each release window e_i is localized in a connected part of the tree T, while the shared service group X_1 forms the connected spine that links both release windows.

Chapter 5

Cycle in SuperHyperGraph

This chapter investigates cycles in SuperHyperGraphs and related concepts.

5.1 Cycles in graphs, hypergraphs, and n-SuperHyperGraphs

This section defines the notion of a *cycle* in three settings: graphs, hypergraphs, and n-SuperHyperGraphs. For hypergraphs (and thus for n-SuperHyperGraphs, which are hypergraphs on supervertices), we adopt the classical *Berge* notion of a cycle, because it applies to arbitrary (non-uniform) hyperedges and is compatible with incidence-based acyclicity notions (Hypertrees and SuperHyperTrees).

5.1.1 Cycles in graphs

A cycle in a graph is a closed simple vertex sequence of length at least three, with consecutive vertices adjacent and no vertex repeated except start/end. Related notions are also known, such as cycles in fuzzy graphs [134, 135] and cycles in neutrosophic graphs [136].

Definition 5.1.1 (Cycle in a graph)**.** Let $G = (V, E)$ be a finite simple undirected graph. A *(simple) cycle* in G is a vertex sequence

$$C = (v_0, v_1, \ldots, v_{k-1}, v_k) \qquad (k \geq 3),$$

such that:

(i) $v_0 = v_k$;

(ii) the vertices $v_0, v_1, \ldots, v_{k-1}$ are pairwise distinct;

(iii) $\{v_{i-1}, v_i\} \in E$ for each $i = 1, \ldots, k$.

The *length* of C is k (equivalently, the number of edges in the cycle).

Remark 5.1.2. Equivalently, a cycle can be viewed as a connected 2-regular subgraph. In this book we use the sequence definition in Definition 5.1.1.

5.1.2 Berge cycles in hypergraphs

A cycle in a hypergraph is typically a Berge cycle: an alternating vertex–hyperedge sequence returning to the start, with distinct vertices and hyperedges, consecutive vertices contained in each hyperedge [137–139].

Definition 5.1.3 (Berge cycle in a hypergraph)**.** Let $H = (V, \mathcal{E})$ be a finite hypergraph, where $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. A *Berge cycle* of length $k \geq 2$ is an alternating sequence

$$\mathsf{C} = (v_0, e_1, v_1, e_2, \ldots, e_k, v_k),$$

such that:

(i) $v_0, v_1, \ldots, v_{k-1} \in V$ are pairwise distinct and $v_k = v_0$;

(ii) $e_1, e_2, \ldots, e_k \in \mathcal{E}$ are pairwise distinct hyperedges;

(iii) $\{v_{i-1}, v_i\} \subseteq e_i$ for each $i = 1, \ldots, k$.

The *length* of C is k (the number of hyperedges used).

Remark 5.1.4. A Berge cycle in H corresponds exactly to an ordinary cycle in the incidence graph $\mathrm{Inc}(H)$ (bipartite between vertices and hyperedges). Thus Berge-acyclicity is equivalent to the incidence graph being acyclic.

Remark 5.1.5. Other cycle notions exist for uniform hypergraphs (tight cycles, loose cycles, etc.). Definition 5.1.3 uses Berge cycles because it is the most general and is compatible with incidence-based hypertrees.

5.1.3 Super-Berge cycles in n-SuperHyperGraphs

A cycle in a hypergraph is typically a Berge cycle: an alternating vertex–hyperedge sequence returning to the start, with distinct vertices and hyperedges, consecutive vertices contained in each hyperedge.

Definition 5.1.6 (Super-Berge cycle in an n-SuperHyperGraph)**.** Let V_0 be a finite nonempty base set and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0, where $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. A *Super-Berge cycle* of length $k \geq 2$ is an alternating sequence

$$\mathsf{C} = (X_0, \varepsilon_1, X_1, \varepsilon_2, \ldots, \varepsilon_k, X_k),$$

such that:

(i) $X_0, X_1, \ldots, X_{k-1} \in V$ are pairwise distinct n-supervertices and $X_k = X_0$;

(ii) $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k \in E$ are pairwise distinct n-superedges;

(iii) $\{X_{i-1}, X_i\} \subseteq \varepsilon_i$ for each $i = 1, \ldots, k$.

The *length* of C is k (the number of superedges used).

Remark 5.1.7. A Super-Berge cycle in $\mathrm{SHG}^{(n)}$ is precisely a graph cycle in the incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(n)})$, which is bipartite between n-supervertices and n-superedges. Hence an n-SuperHyperGraph is Super-Berge-acyclic if and only if its incidence graph is a tree/forest, depending on connectedness.

Remark 5.1.8. Definition 5.1.6 is intrinsic at the supervertex level. If desired, one may additionally study how base-level supports $\mathrm{Flat}_n(X_i) \subseteq V_0$ evolve along a cycle, but such base-level considerations are not required for the definition of a Super-Berge cycle.

Example 5.1.9 (Real-life example of a Super-Berge cycle in an n-SuperHyperGraph)**. Circular coordination among nested departments in a product launch.** Let V_0 be a finite set of employees involved in a product launch:

$$V_0 = \{\mathsf{PM}, \mathsf{Dev1}, \mathsf{Dev2}, \mathsf{QA}, \mathsf{Legal}, \mathsf{PR}\}.$$

Take $n = 2$. Interpret a level-1 object as a *team* (a subset of V_0), and a level-2 object as a *committee* (a set of teams), hence an element of $\mathcal{P}^2(V_0)$.

Define three 2-supervertices (each is a committee of teams):

$$X_0 := \big\{\{\mathsf{PM}\}, \{\mathsf{Dev1}, \mathsf{Dev2}\}\big\} \quad (\text{product \& engineering committee}),$$

$$X_1 := \big\{\{\mathsf{QA}\}, \{\mathsf{Dev1}, \mathsf{Dev2}\}\big\} \quad (\text{quality \& engineering committee}),$$

$$X_2 := \big\{\{\mathsf{Legal}\}, \{\mathsf{PR}\}\big\} \quad (\text{legal \& communications committee}).$$

Let $V := \{X_0, X_1, X_2\} \subseteq \mathcal{P}^2(V_0)$.

Assume three distinct coordination sessions (superedges) are required:

$$\varepsilon_1 := \{X_0, X_1\} \quad (\text{handoff: product requirements to QA plan}),$$

$$\varepsilon_2 := \{X_1, X_2\} \quad (\text{handoff: QA constraints to legal/comms review}),$$

$$\varepsilon_3 := \{X_2, X_0\} \quad (\text{handoff: approval back to product/engineering for launch decision}),$$

and set $E := \{\varepsilon_1, \varepsilon_2, \varepsilon_3\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Then $\mathrm{SHG}^{(2)} = (V, E)$ is a 2-SuperHyperGraph.

The alternating sequence

$$\mathsf{C} = (X_0, \varepsilon_1, X_1, \varepsilon_2, X_2, \varepsilon_3, X_0)$$

is a Super-Berge cycle of length 3: the supervertices X_0, X_1, X_2 are distinct, the superedges $\varepsilon_1, \varepsilon_2, \varepsilon_3$ are distinct, and each consecutive pair of supervertices co-occurs in the corresponding coordination session. Operationally, this represents a *circular dependency* of approvals in the launch process.

5.2 Hamiltonian cycles in graphs, hypergraphs, and n-SuperHyperGraphs

This section defines *Hamiltonian cycles* in three settings. For hypergraphs and n-SuperHyperGraphs we adopt the *Berge* notion of a cycle (alternating vertex–edge sequences), and then impose the Hamiltonian requirement: *every (super)vertex is visited exactly once.*

5.2.1 Hamiltonian cycles in graphs

A Hamiltonian cycle in a graph is a simple cycle that visits every vertex exactly once and returns to the starting vertex, using graph edges [140–143]. Related perspectives have also been studied, such as Hamiltonian cycles in fuzzy graphs [144–147] and in neutrosophic graphs [148, 149].

Definition 5.2.1 (Hamiltonian cycle in a graph)**.** [140, 141] Let $G = (V, E)$ be a finite simple undirected graph. A *Hamiltonian cycle* in G is a (simple) cycle

$$C \;=\; (v_0, v_1, \dots, v_{n-1}, v_n), \qquad n := |V|,$$

such that:

(i) $v_0 = v_n$;

(ii) the vertices $v_0, v_1, \dots, v_{n-1}$ are pairwise distinct;

(iii) $\{v_{i-1}, v_i\} \in E$ for each $i = 1, \dots, n$;

(iv) $\{v_0, v_1, \dots, v_{n-1}\} = V$ (equivalently, the cycle visits every vertex exactly once).

A graph G is called *Hamiltonian* if it contains a Hamiltonian cycle.

5.2.2 Hamiltonian Berge cycles in hypergraphs

A Hamiltonian cycle in a hypergraph is a Hamiltonian Berge cycle: an alternating vertex–hyperedge cycle visiting every vertex exactly once, with distinct hyperedges connecting consecutive vertices (cf. [150–152]).

Definition 5.2.2 (Hamiltonian Berge cycle in a hypergraph)**.** Let $H = (V, \mathcal{E})$ be a finite hypergraph, where $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. A *Hamiltonian Berge cycle* in H is a Berge cycle

$$\mathsf{C} \;=\; (v_0, e_1, v_1, e_2, \dots, e_n, v_n), \qquad n := |V|,$$

such that:

(i) $v_0 = v_n$;

(ii) $v_0, v_1, \ldots, v_{n-1}$ are pairwise distinct and $\{v_0, \ldots, v_{n-1}\} = V$;

(iii) $e_1, e_2, \ldots, e_n \in \mathcal{E}$ are pairwise distinct;

(iv) $\{v_{i-1}, v_i\} \subseteq e_i$ for each $i = 1, \ldots, n$.

A hypergraph H is called *Berge-Hamiltonian* (or simply *Hamiltonian* when the context is clear) if it contains a Hamiltonian Berge cycle.

Remark 5.2.3. There are several non-equivalent notions of Hamiltonicity in hypergraphs (e.g., tight/loose Hamilton cycles in r-uniform hypergraphs). Definition 5.2.2 uses Berge cycles because it applies to arbitrary hypergraphs and is consistent with incidence-based notions of paths/cycles used elsewhere in this book.

5.2.3 Hamiltonian Super-Berge cycles in n-SuperHyperGraphs

A Hamiltonian cycle in a superhypergraph is a Hamiltonian Super-Berge cycle: an alternating supervertex–superedge cycle visiting every supervertex exactly once, returning to the start.

Definition 5.2.4 (Hamiltonian Super-Berge cycle in an n-SuperHyperGraph)**.** Let V_0 be a finite nonempty base set and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0, where $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. A *Hamiltonian Super-Berge cycle* in $\mathrm{SHG}^{(n)}$ is a Super-Berge cycle

$$\mathsf{C} = (X_0, \varepsilon_1, X_1, \varepsilon_2, \ldots, \varepsilon_N, X_N), \qquad N := |V|,$$

such that:

(i) $X_0 = X_N$;

(ii) $X_0, X_1, \ldots, X_{N-1}$ are pairwise distinct and $\{X_0, \ldots, X_{N-1}\} = V$ (i.e., every n-supervertex is visited exactly once);

(iii) $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N \in E$ are pairwise distinct;

(iv) $\{X_{i-1}, X_i\} \subseteq \varepsilon_i$ for each $i = 1, \ldots, N$.

An n-SuperHyperGraph is called *Hamiltonian* (more precisely, *Super-Berge-Hamiltonian*) if it contains a Hamiltonian Super-Berge cycle.

Remark 5.2.5. A Hamiltonian Super-Berge cycle corresponds to a cycle in the incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ that visits every supervertex-node exactly once (while alternating through superedge-nodes). The definition above is intrinsic at the supervertex/superedge level; any base-level interpretation via flattening Flat_n is optional.

Example 5.2.6 (Real-life example of a Hamiltonian Berge cycle in a hypergraph)**. Rotating on-call handoff across all teams.** Let V be a set of four operational teams:

$$V = \{\mathsf{DB}, \mathsf{API}, \mathsf{Security}, \mathsf{SRE}\}.$$

Define four incident-rotation sessions (hyperedges) in which two consecutive teams coordinate the handoff:

$$e_1 = \{\mathsf{DB}, \mathsf{API}\}, \qquad e_2 = \{\mathsf{API}, \mathsf{Security}\},$$

$$e_3 = \{\mathsf{Security}, \mathsf{SRE}\}, \qquad e_4 = \{\mathsf{SRE}, \mathsf{DB}\}.$$

Let $\mathcal{E} := \{e_1, e_2, e_3, e_4\}$ and $H = (V, \mathcal{E})$. Then

$$\mathsf{C} = (\mathsf{DB}, e_1, \mathsf{API}, e_2, \mathsf{Security}, e_3, \mathsf{SRE}, e_4, \mathsf{DB})$$

is a Hamiltonian Berge cycle in H: it visits each team exactly once (except the start/end team) and uses distinct hyperedges connecting consecutive vertices. Operationally, this models a weekly rotation where responsibility passes through every team and returns to the start.

Example 5.2.7 (Real-life example of a Hamiltonian Super-Berge cycle in an n-SuperHyperGraph)**. A closed review loop visiting every committee exactly once.** Let V_0 be a finite set of staff roles in a product organization:

$$V_0 = \{\mathsf{PM}, \mathsf{Dev}, \mathsf{QA}, \mathsf{Legal}, \mathsf{PR}\}.$$

Take $n = 2$. Interpret a level-1 object as a team (subset of V_0) and a level-2 object as a committee (set of teams), hence an element of $\mathcal{P}^2(V_0)$.

Define four distinct committees (supervertices):

$$X_0 = \{\{\mathsf{PM}\}, \{\mathsf{Dev}\}\} \quad (\text{product/engineering committee}),$$

$$X_1 = \{\{\mathsf{QA}\}, \{\mathsf{Dev}\}\} \quad (\text{quality/engineering committee}),$$

$$X_2 = \{\{\mathsf{Legal}\}, \{\mathsf{PR}\}\} \quad (\text{legal/communications committee}),$$

$$X_3 = \{\{\mathsf{PM}\}, \{\mathsf{PR}\}\} \quad (\text{launch-communications committee}),$$

and set $V = \{X_0, X_1, X_2, X_3\} \subseteq \mathcal{P}^2(V_0)$.

Let superedges represent scheduled cross-committee checkpoints:

$$\varepsilon_1 = \{X_0, X_1\} \quad (\text{requirements} \to \text{test plan}),$$

$$\varepsilon_2 = \{X_1, X_2\} \quad (\text{test constraints} \to \text{legal review}),$$

$$\varepsilon_3 = \{X_2, X_3\} \quad (\text{legal clearance} \to \text{launch messaging}),$$

$$\varepsilon_4 = \{X_3, X_0\} \quad (\text{launch decision} \to \text{requirements reset}),$$

and set $E = \{\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4\}$. Then

$$\mathsf{C} = (X_0, \varepsilon_1, X_1, \varepsilon_2, X_2, \varepsilon_3, X_3, \varepsilon_4, X_0)$$

is a Hamiltonian Super-Berge cycle in $\mathrm{SHG}^{(2)} = (V, E)$: it visits every committee X_i exactly once (except the repeated start/end) and uses distinct superedges. Operationally, this models a closed governance loop that touches every committee once per release cycle.

Chapter 6

Bipartite Structure

In this chapter, we discuss bipartite and multipartite structures in graphs, hypergraphs, and superhypergraphs.

6.1 Bipartite SuperHyperGraph

A bipartite graph is a graph whose vertices split into two disjoint parts, a nd every edge has endpoints in different parts only [153–155]. As related concepts, bipartite fuzzy graphs [156], bipartite neutrosophic graphs [157, 158], tripartite graphs [159–161], bipartable graphs [162], and bipartite directed graphs [163, 164] are well known. In addition, as graph classes formed by bipartite + an additional property, several families are known, including bipartite tolerance graphs [165, 166], circular convex bipartite graphs [167, 168], chordal bipartite graphs [169, 170], and bipartite permutation graphs [171, 172]. Bipartite graphs admit efficient matchings and colorings, model two-type relations naturally, and avoid odd cycles, simplifying many algorithms.

A bipartite hypergraph is a hypergraph with a vertex partition $A \sqcup B$ such that each hyperedge intersects both parts, $e \cap A \neq \varnothing$ and $e \cap B \neq \varnothing$ [173–175]. A bipartite superhypergraph is a superhypergraph with a supervertex partition $\mathcal{V}_1 \sqcup \mathcal{V}_2$ such that every superhyperedge meets both parts nontrivially. The relevant definitions and related notions are presented below.

Definition 6.1.1 (Bipartite Graph)**.** [153–155] A graph is a pair $G = (V, E)$ where V is a (finite) set and $E \subseteq \big\{\{u, v\} \subseteq V : u \neq v\big\}$. The graph G is *bipartite* if there exist disjoint sets $V_1, V_2 \subseteq V$ such that

$$V = V_1 \sqcup V_2 \quad \text{and} \quad \forall \{u, v\} \in E : \big(u \in V_1 \;\&\; v \in V_2\big) \text{ or } \big(u \in V_2 \;\&\; v \in V_1\big).$$

Equivalently, no edge has both endpoints in the same part.

Definition 6.1.2 (Bipartite Hypergraph)**.** [173–175] A hypergraph is a pair $H = (V, \mathcal{E})$ where V is a (finite) set and $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\varnothing\}$ is a family of nonempty subsets of V (hyperedges). The hypergraph H is *bipartite* if there exist disjoint sets $A, B \subseteq V$ such that

$$V = A \sqcup B \quad \text{and} \quad \forall e \in \mathcal{E} : \; e \cap A \neq \varnothing \text{ and } e \cap B \neq \varnothing.$$

(In particular, no hyperedge is contained entirely in one part.)

Definition 6.1.3 (Bipartite n-SuperHyperGraph)**.** Fix an integer $n \geq 1$ and a nonempty *base set* V_0. An *n-SuperHyperGraph* (in the sense of Chapter 2) is a pair

$$\mathrm{SHG}^{(n)} = (V, E),$$

where the *supervertex set* V is a nonempty family of level-n set-objects,

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

and the *superhyperedge family* is

$$E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

We say that $\mathrm{SHG}^{(n)}$ is *bipartite* if there exist disjoint (nonempty) subfamilies $V_1, V_2 \subseteq V$ such that

$$V = V_1 \sqcup V_2 \quad \text{and} \quad \forall\, e \in E: \ e \cap V_1 \neq \varnothing \text{ and } e \cap V_2 \neq \varnothing.$$

In this case, (V_1, V_2) is called a *bipartition* of $\mathrm{SHG}^{(n)}$.

Example 6.1.4 (A bipartite 2-SuperHyperGraph)**.** Let $V_0 = \{1, 2, 3, 4\}$ and set $n = 2$. Define four level-2 supervertices (elements of $\mathcal{P}^2(V_0)$) by

$$X_1 := \{\{1\}, \{2\}\}, \quad X_2 := \{\{2\}, \{3\}\}, \quad X_3 := \{\{3\}, \{4\}\}, \quad X_4 := \{\{1\}, \{4\}\}.$$

Let

$$V := \{X_1, X_2, X_3, X_4\} \subseteq \mathcal{P}^2(V_0).$$

Partition V as

$$V_1 := \{X_1, X_3\}, \qquad V_2 := \{X_2, X_4\},$$

and define a superhyperedge family

$$E := \{\, e_1, e_2 \,\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}, \qquad e_1 := \{X_1, X_2, X_3\}, \quad e_2 := \{X_3, X_4\}.$$

Then $\mathrm{SHG}^{(2)} = (V, E)$ is bipartite with bipartition (V_1, V_2), since

$$e_1 \cap V_1 = \{X_1, X_3\} \neq \varnothing, \quad e_1 \cap V_2 = \{X_2\} \neq \varnothing,$$

and

$$e_2 \cap V_1 = \{X_3\} \neq \varnothing, \quad e_2 \cap V_2 = \{X_4\} \neq \varnothing.$$

6.2 Tripartite graph, tripartite hypergraph, and tripartite n-superhypergraph

A tripartite graph is a graph whose vertices split into three independent sets, so every edge connects vertices from different parts, equivalently a 3-colorable graph [176–181]. A tripartite hypergraph is a hypergraph whose vertices split into three parts, and each hyperedge contains at most one vertex from each part [182–184]. Related notions, such as fuzzy tripartite graphs, have also been studied (cf. [185–187]). A tripartite n-superhypergraph partitions supervertices into three classes, and every level-n superhyperedge, after flattening, uses at most one supervertex from each class.

Definition 6.2.1 (Tripartite graph)**.** [176, 177] A (finite simple) graph is a pair $G = (V, E)$ with $E \subseteq \binom{V}{2}$. We say that G is *tripartite* if there exist pairwise disjoint sets V_1, V_2, V_3 such that

$$V = V_1 \dot\cup V_2 \dot\cup V_3,$$

and every edge has its endpoints in different parts, i.e.,

$$\forall \{u, v\} \in E, \quad \exists\, i \neq j \text{ with } u \in V_i,\ v \in V_j.$$

Equivalently, V_1, V_2, V_3 are independent sets and G is 3-colorable.

Definition 6.2.2 (Tripartite hypergraph)**.** A (finite) hypergraph is a pair $H = (V, \mathcal{E})$ with $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. We say that H is *tripartite* (or 3*-partite*) if there exist pairwise disjoint sets V_1, V_2, V_3 such that

$$V = V_1 \dot\cup V_2 \dot\cup V_3,$$

and every hyperedge contains *at most one* vertex from each part:

$$\forall\, e \in \mathcal{E},\ \forall\, i \in \{1, 2, 3\}, \qquad |e \cap V_i| \le 1.$$

Remark 6.2.3. Under Definition 6.2.2, every hyperedge has size at most 3. If, in addition, $|e| = 3$ for all $e \in \mathcal{E}$, then H is *3-uniform tripartite* and each hyperedge meets each part in exactly one vertex.

Definition 6.2.4 (Tripartite n-superhypergraph)**.** Fix $n \ge 0$. An n*-superhypergraph* is a pair

$$\mathrm{SHG}^{(n)} = (\mathcal{V}, \mathcal{F}^{(n)}),$$

where $\mathcal{V} \neq \emptyset$ is a set of supervertices and

$$\mathcal{F}^{(n)} \subseteq \mathcal{P}^{n}(\mathcal{V}) \setminus \{\emptyset\}$$

is a family of level-n superhyperedges.

We say that $\mathrm{SHG}^{(n)}$ is *tripartite* if there exist pairwise disjoint sets $\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3 \subseteq \mathcal{V}$ such that

$$\mathcal{V} = \mathcal{V}_1 \sqcup \mathcal{V}_2 \sqcup \mathcal{V}_3,$$

and for every $F \in \mathcal{F}^{(n)}$, the vertex-level flattening $\mathrm{Flat}(F) \subseteq \mathcal{V}$ satisfies

$$|\,\mathrm{Flat}(F) \cap \mathcal{V}_i| \le 1 \qquad (i = 1, 2, 3).$$

Remark 6.2.5. If one additionally requires $|\,\mathrm{Flat}(F)| = 3$ for all $F \in \mathcal{F}^{(n)}$, then every flattened superhyperedge meets each $\mathcal{V}_i$ in exactly one element, giving a 3-uniform tripartite n-superhypergraph.

6.3 Parity n-SuperHyperGraphs

A parity graph is a graph where every two induced paths between any vertices have lengths with identical parity always [188–191]. A parity n-SuperHyperGraph has a primal graph that is parity, so induced supervertex paths share length parity for all pairs.

Definition 6.3.1 (Induced path in a graph)**.** Let $G = (V, E)$ be a finite simple undirected graph. A *(simple) u–v path* is a vertex sequence

$$P = (v_0, v_1, \dots, v_k)$$

with $v_0 = u$, $v_k = v$, all $v_0, \dots, v_k$ distinct, and $\{v_{i-1}, v_i\} \in E$ for $i = 1, \dots, k$. It is an *induced u–v path* if, additionally,

$$\{v_i, v_j\} \notin E \quad \text{for all } 0 \le i < j \le k \text{ with } |i - j| \ge 2.$$

The *length* of P is $\ell(P) := k$ (the number of edges traversed).

Definition 6.3.2 (Parity graph)**.** A finite simple undirected graph $G = (V, E)$ is called a *parity graph* if for every pair of vertices $u, v \in V$, any two induced u–v paths P, Q in G have the same parity of length, i.e.,

$$\ell(P) \equiv \ell(Q) \pmod 2.$$

Definition 6.3.3 (Primal (2-section) graph of a hypergraph)**.** Let $H = (V, \mathcal{E})$ be a finite hypergraph, where $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. The *primal graph* (or *2-section*) of H is the simple undirected graph

$$\Pr(H) = (V, E_{\Pr}), \qquad E_{\Pr} := \big\{ \{u, v\} \in \tbinom{V}{2} : \ \exists\, e \in \mathcal{E} \text{ with } \{u, v\} \subseteq e \big\}.$$

Definition 6.3.4 (Parity hypergraph)**.** A hypergraph $H = (V, \mathcal{E})$ is called a *parity hypergraph* if its primal graph $\Pr(H)$ (Definition 6.3.3) is a parity graph (Definition 6.3.2). Equivalently, for every $u, v \in V$, any two induced u–v paths in $\Pr(H)$ have lengths of the same parity.

Remark 6.3.5. If H is a (simple) graph (i.e., every hyperedge has size 2), then $\Pr(H) = H$. Hence Definition 6.3.4 reduces to the usual notion of a parity graph.

Definition 6.3.6 (Primal graph of an n-SuperHyperGraph)**.** Let V_0 be a finite nonempty base set and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0, where $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. The *primal graph* of $\mathrm{SHG}^{(n)}$ is the simple undirected graph

$$\Pr\big(\mathrm{SHG}^{(n)}\big) = (V, E_{\Pr}), \qquad E_{\Pr} := \big\{ \{X, Y\} \in \tbinom{V}{2} : \ \exists\, \varepsilon \in E \text{ with } \{X, Y\} \subseteq \varepsilon \big\}.$$

Definition 6.3.7 (Parity n-SuperHyperGraph)**.** An n-SuperHyperGraph $\mathrm{SHG}^{(n)} = (V, E)$ is called a *parity n-SuperHyperGraph* (or *parity superhypergraph*) if its primal graph $\Pr(\mathrm{SHG}^{(n)})$ (Definition 6.3.6) is a parity graph (Definition 6.3.2). Equivalently, for every pair of supervertices $X, Y \in V$, any two induced X–Y paths in $\Pr(\mathrm{SHG}^{(n)})$ have lengths of the same parity.

Example 6.3.8 (A parity 2-SuperHyperGraph). Let $V_0 = \{a, b, c, d\}$ and take $n = 2$. Define four 2-supervertices (elements of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$) by

$$X_1 := \{\{a\}\}, \qquad X_2 := \{\{b\}\}, \qquad X_3 := \{\{c\}\}, \qquad X_4 := \{\{d\}\},$$

and set

$$V := \{X_1, X_2, X_3, X_4\} \subseteq \mathcal{P}^2(V_0).$$

Define three superedges by

$$\varepsilon_{12} := \{X_1, X_2\}, \qquad \varepsilon_{23} := \{X_2, X_3\}, \qquad \varepsilon_{34} := \{X_3, X_4\}, \qquad E := \{\varepsilon_{12}, \varepsilon_{23}, \varepsilon_{34}\} \subseteq \mathcal{P}(V)\backslash\{\emptyset\}.$$

Then $\mathrm{SHG}^{(2)} = (V, E)$ is a finite 2-SuperHyperGraph.

Primal graph. By Definition 6.3.6, the primal graph $\Pr(\mathrm{SHG}^{(2)})$ has vertex set V and an edge $\{X, Y\}$ whenever $X \neq Y$ and $\{X, Y\} \subseteq \varepsilon$ for some $\varepsilon \in E$. Hence

$$E\big(\Pr(\mathrm{SHG}^{(2)})\big) = \big\{\{X_1, X_2\}, \{X_2, X_3\}, \{X_3, X_4\}\big\},$$

so $\Pr(\mathrm{SHG}^{(2)})$ is the path

$$X_1 \text{—} X_2 \text{—} X_3 \text{—} X_4.$$

Parity property. In a tree (and hence in a path), there is a *unique* induced path between any two vertices. Therefore, for every pair $X, Y \in V$, any two induced X–Y paths in $\Pr(\mathrm{SHG}^{(2)})$ (if one insists on considering two of them) must coincide and thus have the same length parity. Equivalently, $\Pr(\mathrm{SHG}^{(2)})$ is a parity graph (Definition 6.3.2). Consequently, $\mathrm{SHG}^{(2)}$ is a parity 2-SuperHyperGraph in the sense of Definition 6.3.7.

6.4 Convex bipartite graph, convex bipartite hypergraph, and convex bipartite n-SuperHyperGraph

A convex bipartite graph has an ordering of one part so every neighbor set in the other part is consecutive [192–195]. A convex bipartite hypergraph has a convex incidence graph on vertices or edges, making each incidence neighborhood a consecutive interval. A convex bipartite n-SuperHyperGraph has a convex incidence graph, so superedge memberships or incident superedges appear as intervals under ordering.

Definition 6.4.1 (Convexity over one side). Let $G = (U \cup V, E)$ be a finite simple bipartite graph with bipartition (U, V). For $v \in V$, write $N_G(v) \subseteq U$ for the neighborhood of v in U.

We say that G is *convex over* U if there exists a bijection (ordering)

$$\sigma : U \longrightarrow \{1, 2, \ldots, |U|\}$$

such that, for every $v \in V$, the set $N_G(v)$ is an interval in this ordering; equivalently, there exist integers $1 \leq a_v \leq b_v \leq |U|$ with

$$N_G(v) \;=\; \{\, u \in U : \; a_v \leq \sigma(u) \leq b_v \,\}.$$

(If $N_G(v) = \emptyset$, the condition holds trivially.)

Analogously, G is *convex over* V if there exists an ordering $\tau : V \to \{1, \ldots, |V|\}$ such that $N_G(u) \subseteq V$ is a τ-interval for every $u \in U$.

Definition 6.4.2 (Convex bipartite graph and biconvex graph)**.** A bipartite graph $G = (U \cup V, E)$ is called a *convex bipartite graph* if it is convex over U or convex over V (Definition 6.4.1).

It is called *biconvex* if it is convex over both U and V (possibly with different orderings).

Definition 6.4.3 (Incidence graph of a hypergraph)**.** Let $H = (V, \mathcal{E})$ be a finite hypergraph, where $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Its *incidence graph* (Levi graph) is the bipartite graph

$$\mathrm{Inc}(H) \;=\; \big(V \cup \mathcal{E},\ F\big), \qquad F \;:=\; \big\{\{v, e\} :\ v \in V,\ e \in \mathcal{E},\ v \in e\big\},$$

with bipartition $(V, \mathcal{E})$.

Definition 6.4.4 (Convex bipartite hypergraph)**.** Let $H = (V, \mathcal{E})$ be a hypergraph and $\mathrm{Inc}(H)$ its incidence graph.

(i) H is *vertex-convex* if $\mathrm{Inc}(H)$ is convex over V; equivalently, there exists an ordering $\sigma : V \to \{1, \ldots, |V|\}$ such that for every hyperedge $e \in \mathcal{E}$, the set of its incident vertices $e \subseteq V$ forms an interval:

$$\exists\ 1 \le a_e \le b_e \le |V| \ \text{ such that } \ e = \{\, v \in V :\ a_e \le \sigma(v) \le b_e \,\}.$$

(ii) H is *edge-convex* if $\mathrm{Inc}(H)$ is convex over $\mathcal{E}$; equivalently, there exists an ordering $\tau : \mathcal{E} \to \{1, \ldots, |\mathcal{E}|\}$ such that for every vertex $v \in V$, the set of incident hyperedges

$$\mathcal{E}(v) \;:=\; \{\, e \in \mathcal{E} :\ v \in e \,\}$$

is a τ-interval.

(iii) H is a *convex bipartite hypergraph* if it is vertex-convex or edge-convex, i.e., if $\mathrm{Inc}(H)$ is a convex bipartite graph (Definition 6.4.2).

(iv) H is *biconvex* if it is both vertex-convex and edge-convex, i.e., if $\mathrm{Inc}(H)$ is biconvex.

Definition 6.4.5 (Incidence graph of an n-SuperHyperGraph)**.** Let V_0 be a finite nonempty base set and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0, where $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Its *incidence graph* is the bipartite graph

$$\mathrm{Inc}\big(\mathrm{SHG}^{(n)}\big) \;=\; \big(V \cup E,\ F\big), \qquad F \;:=\; \big\{\{X, \varepsilon\} :\ X \in V,\ \varepsilon \in E,\ X \in \varepsilon\big\},$$

with bipartition (V, E).

Definition 6.4.6 (Convex bipartite n-SuperHyperGraph)**.** Let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph and let $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ be its incidence graph.

(i) $\mathrm{SHG}^{(n)}$ is *supervertex-convex* if $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ is convex over V; equivalently, there exists an ordering $\sigma : V \to \{1, \ldots, |V|\}$ such that for every superedge $\varepsilon \in E$, the incident supervertex set $\varepsilon \subseteq V$ is a σ-interval.

(ii) $\mathrm{SHG}^{(n)}$ is *superedge-convex* if $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ is convex over E; equivalently, there exists an ordering $\tau : E \to \{1, \ldots, |E|\}$ such that for every supervertex $X \in V$, the incident superedge set

$$E(X) := \{\, \varepsilon \in E : \ X \in \varepsilon \,\}$$

is a τ-interval.

(iii) $\mathrm{SHG}^{(n)}$ is a *convex bipartite n-SuperHyperGraph* if it is supervertex-convex or superedge-convex, i.e., if $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ is a convex bipartite graph.

(iv) $\mathrm{SHG}^{(n)}$ is *biconvex* if it is both supervertex-convex and superedge-convex.

Remark 6.4.7. Definitions 6.4.4 and 6.4.6 are direct incidence-graph liftings of the standard convex bipartite graph notion: $\mathrm{Inc}(H)$ (respectively $\mathrm{Inc}(\mathrm{SHG}^{(n)})$) is always bipartite, so convexity is imposed by requiring consecutive neighborhoods on one side of this bipartition.

Example 6.4.8 (A convex bipartite 2-SuperHyperGraph)**.** Let $V_0 = \{a, b, c, d\}$ and take $n = 2$. Define four 2-supervertices (elements of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$) by

$$X_1 := \{\{a\}\}, \qquad X_2 := \{\{b\}\}, \qquad X_3 := \{\{c\}\}, \qquad X_4 := \{\{d\}\},$$

and set

$$V := \{X_1, X_2, X_3, X_4\} \subseteq \mathcal{P}^2(V_0).$$

Define three superedges by

$$\varepsilon_1 := \{X_1, X_2\}, \qquad \varepsilon_2 := \{X_2, X_3\}, \qquad \varepsilon_3 := \{X_3, X_4\},$$
$$E := \{\varepsilon_1, \varepsilon_2, \varepsilon_3\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Then $\mathrm{SHG}^{(2)} = (V, E)$ is a finite 2-SuperHyperGraph.

Supervertex-convexity. Order the supervertices by

$$\sigma(X_i) := i \qquad (i = 1, 2, 3, 4).$$

For each superedge $\varepsilon_j \in E$, the incident supervertex set is a σ-interval:

$$\begin{aligned}
\varepsilon_1 &= \{X_1, X_2\} = \{\sigma^{-1}(1), \sigma^{-1}(2)\}, \\
\varepsilon_2 &= \{X_2, X_3\} = \{\sigma^{-1}(2), \sigma^{-1}(3)\}, \\
\varepsilon_3 &= \{X_3, X_4\} = \{\sigma^{-1}(3), \sigma^{-1}(4)\}.
\end{aligned}$$

Equivalently, in the incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(2)})$ every superedge-vertex is adjacent to a consecutive block of supervertex-vertices under the ordering σ. Hence $\mathrm{Inc}(\mathrm{SHG}^{(2)})$ is convex over V, so $\mathrm{SHG}^{(2)}$ is supervertex-convex.

(Optional) Superedge-convexity. If we order the superedges by $\tau(\varepsilon_j) := j$ for $j = 1, 2, 3$, then for each supervertex X_i, the incident superedge set $E(X_i)$ is also a τ-interval:

$$E(X_1) = \{\varepsilon_1\}, \quad E(X_2) = \{\varepsilon_1, \varepsilon_2\}, \quad E(X_3) = \{\varepsilon_2, \varepsilon_3\}, \quad E(X_4) = \{\varepsilon_3\}.$$

Thus $\mathrm{SHG}^{(2)}$ is in fact *biconvex*. In particular, it is a convex bipartite 2-SuperHyperGraph in the sense of Definition 6.4.6.

6.5 Biregular graph, biregular hypergraph, and biregular n-SuperHyperGraph

A biregular graph is a bipartite graph in which every vertex on one side has degree x and every vertex on the other side has degree y [196–198]. A biregular hypergraph is a hypergraph whose incidence graph is biregular, meaning each vertex lies in exactly r hyperedges and each hyperedge has size s (cf. [199]). A biregular n-SuperHyperGraph is an n-superhypergraph whose incidence graph is biregular, so every supervertex belongs to r superedges and every superedge contains s supervertices.

Definition 6.5.1 ((x, y)-biregular bipartite graph)**.** Let $G = (U \cup V, E)$ be a finite simple bipartite graph with fixed bipartition (U, V). For $u \in U$, write $\deg_G(u)$ for the degree of u in G; similarly for $v \in V$.

For integers $x, y \geq 0$, we say that G is (x, y)-*biregular* if

$$\deg_G(u) = x \quad (\forall u \in U), \qquad \deg_G(v) = y \quad (\forall v \in V).$$

A graph is called *biregular* if it is (x, y)-biregular for some x, y.

Remark 6.5.2 (Degree balance)**.** If $G = (U \cup V, E)$ is (x, y)-biregular, then necessarily $x|U| = y|V|$ by double counting edge incidences.

For hypergraphs, a natural bipartite graph canonically associated with H is its incidence (Levi) graph. We define biregularity of a hypergraph via biregularity of this incidence graph.

Definition 6.5.3 (Incidence graph of a hypergraph)**.** Let $H = (V, \mathcal{E})$ be a finite hypergraph, where $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Its *incidence graph* is the bipartite graph

$$\mathrm{Inc}(H) \;=\; \bigl(V \cup \mathcal{E},\ F\bigr), \qquad F \;:=\; \bigl\{\{v, e\} :\ v \in V,\ e \in \mathcal{E},\ v \in e\bigr\},$$

with bipartition $(V, \mathcal{E})$. For $v \in V$, the degree $\deg_{\mathrm{Inc}(H)}(v)$ equals the number of hyperedges incident to v; for $e \in \mathcal{E}$, $\deg_{\mathrm{Inc}(H)}(e) = |e|$.

Definition 6.5.4 ((r, s)-biregular hypergraph)**.** Let $H = (V, \mathcal{E})$ be a hypergraph and let $r, s \geq 0$ be integers. We say that H is (r, s)-*biregular* if its incidence graph $\mathrm{Inc}(H)$ (Definition 6.5.3) is (r, s)-biregular with respect to the bipartition $(V, \mathcal{E})$; equivalently,

$$\deg_{\mathrm{Inc}(H)}(v) = r \quad (\forall v \in V), \qquad \deg_{\mathrm{Inc}(H)}(e) = s \quad (\forall e \in \mathcal{E}).$$

In concrete hypergraph terms, this means:

(i) every vertex belongs to exactly r hyperedges (constant vertex-incidence), and

(ii) every hyperedge has cardinality s (i.e., H is s-uniform).

A hypergraph is called *biregular* if it is (r, s)-biregular for some r, s.

Remark 6.5.5 (Incidence balance)**.** If $H = (V, \mathcal{E})$ is (r, s)-biregular, then $r|V| = s|\mathcal{E}|$ by double counting incidences.

Definition 6.5.6 (Incidence graph of an n-SuperHyperGraph)**.** Let V_0 be a finite nonempty base set and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0, with $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Its *incidence graph* is the bipartite graph

$$\mathrm{Inc}\big(\mathrm{SHG}^{(n)}\big) = \big(V \cup E,\ F\big), \qquad F := \big\{\{X, \varepsilon\} :\ X \in V,\ \varepsilon \in E,\ X \in \varepsilon\big\},$$

with bipartition (V, E). For $X \in V$, $\deg_{\mathrm{Inc}(\mathrm{SHG}^{(n)})}(X)$ equals the number of superedges containing X; for $\varepsilon \in E$, $\deg_{\mathrm{Inc}(\mathrm{SHG}^{(n)})}(\varepsilon) = |\varepsilon|$.

Definition 6.5.7 ((r, s)-biregular n-SuperHyperGraph)**.** Let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph and let $r, s \geq 0$. We say that $\mathrm{SHG}^{(n)}$ is (r, s)-*biregular* if its incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ (Definition 6.5.6) is (r, s)-biregular with respect to the bipartition (V, E), i.e.,

$$\deg_{\mathrm{Inc}(\mathrm{SHG}^{(n)})}(X) = r \quad (\forall X \in V), \qquad \deg_{\mathrm{Inc}(\mathrm{SHG}^{(n)})}(\varepsilon) = s \quad (\forall \varepsilon \in E).$$

Equivalently:

(i) every supervertex belongs to exactly r superedges, and

(ii) every superedge contains exactly s supervertices (uniform superedge size).

Remark 6.5.8 (Balance)**.** If $\mathrm{SHG}^{(n)}$ is (r, s)-biregular, then $r|V| = s|E|$ by double counting incidence pairs (X, ε) with $X \in \varepsilon$.

Example 6.5.9 (A $(2, 2)$-biregular 2-SuperHyperGraph)**.** Let $V_0 = \{a, b, c, d\}$ and take $n = 2$. Define four 2-supervertices (elements of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$) by

$$X_1 := \{\{a\}\}, \qquad X_2 := \{\{b\}\}, \qquad X_3 := \{\{c\}\}, \qquad X_4 := \{\{d\}\},$$

and set

$$V := \{X_1, X_2, X_3, X_4\} \subseteq \mathcal{P}^2(V_0).$$

Define four superedges (each of size 2) by

$$\varepsilon_1 := \{X_1, X_2\}, \qquad \varepsilon_2 := \{X_2, X_3\}, \qquad \varepsilon_3 := \{X_3, X_4\}, \qquad \varepsilon_4 := \{X_4, X_1\},$$

$$E := \{\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Then $\mathrm{SHG}^{(2)} = (V, E)$ is a finite 2-SuperHyperGraph.

Biregularity verification. In the incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(2)})$ (with bipartition (V, E)), a supervertex X_i is adjacent to exactly the superedges that contain it. From the definition of E, each X_i belongs to exactly two superedges:

$$X_1 \in \varepsilon_1, \varepsilon_4; \quad X_2 \in \varepsilon_1, \varepsilon_2; \quad X_3 \in \varepsilon_2, \varepsilon_3; \quad X_4 \in \varepsilon_3, \varepsilon_4.$$

Hence $\deg_{\mathrm{Inc}(\mathrm{SHG}^{(2)})}(X_i) = 2$ for all $i = 1, 2, 3, 4$, so $r = 2$.

Likewise, each superedge ε_j contains exactly two supervertices by construction, so $\deg_{\mathrm{Inc}(\mathrm{SHG}^{(2)})}(\varepsilon_j) = |\varepsilon_j| = 2$ for all $j = 1, 2, 3, 4$, i.e., $s = 2$. Therefore $\mathrm{Inc}(\mathrm{SHG}^{(2)})$ is $(2, 2)$-biregular, and $\mathrm{SHG}^{(2)}$ is a $(2, 2)$-biregular 2-SuperHyperGraph in the sense of Definition 6.5.7.

6.6 Chordal bipartite graph, chordal bipartite hypergraph, and chordal bipartite n-SuperHyperGraph

This section explains chordal bipartite graphs, chordal bipartite hypergraphs, and chordal bipartite n-SuperHyperGraphs.

6.6.1 Chordal bipartite graphs

Chordal bipartite graphs are bipartite graphs with no induced cycles of length at least six; leaves are degree-one vertices there [200–203].

Definition 6.6.1 (Chord and induced cycle)**.** Let $G = (V, E)$ be a finite simple undirected graph. A *chord* of a (simple) cycle $C = (v_0, v_1, \ldots, v_{k-1}, v_k)$ with $v_0 = v_k$ is an edge $\{v_i, v_j\} \in E$ such that $|i - j| \notin \{1, k - 1\}$, i.e., the edge joins two nonconsecutive vertices of C. A cycle is *induced* (or *chordless*) if it has no chord in G.

Definition 6.6.2 (Chordal bipartite graph)**.** A finite simple bipartite graph $B = (X \cup Y, E)$ is called *chordal bipartite* if it has no induced cycle of length at least 6. Equivalently, every cycle in B of length $2k \geq 6$ has a chord (Definition 6.6.1).

Remark 6.6.3. Because B is bipartite, all cycles have even length; in particular, induced 4-cycles are allowed in chordal bipartite graphs. Thus chordal bipartite graphs are generally not chordal graphs.

6.6.2 Chordal bipartite hypergraphs

We define the chordal-bipartite property for hypergraphs via their incidence (Levi) graphs, which are naturally bipartite.

Definition 6.6.4 (Incidence graph of a hypergraph)**.** Let $H = (V, \mathcal{E})$ be a finite hypergraph, where $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Its *incidence graph* is the bipartite graph

$$\mathrm{Inc}(H) \;=\; \big(V \cup \mathcal{E},\ F\big), \qquad F \;:=\; \big\{\{v, e\} :\ v \in V,\ e \in \mathcal{E},\ v \in e\big\},$$

with bipartition $(V, \mathcal{E})$.

Definition 6.6.5 (Chordal bipartite hypergraph)**.** A hypergraph $H = (V, \mathcal{E})$ is called *chordal bipartite* if its incidence graph $\mathrm{Inc}(H)$ (Definition 6.6.4) is a chordal bipartite graph in the sense of Definition 6.6.2. Equivalently, $\mathrm{Inc}(H)$ contains no induced cycle of length at least 6.

Remark 6.6.6. Definition 6.6.5 is incidence-based and is standard in the sense that many hypergraph "chordality" conditions are naturally expressed via the Levi graph.

6.6.3 Chordal bipartite n-SuperHyperGraphs

An n-SuperHyperGraph whose incidence bipartite graph is chordal bipartite: every induced cycle of length at least six has a chord.

Definition 6.6.7 (Incidence graph of an n-SuperHyperGraph)**.** Let V_0 be a finite nonempty base set and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0, where $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Its *incidence graph* is the bipartite graph

$$\mathrm{Inc}\big(\mathrm{SHG}^{(n)}\big) \;=\; \big(V \cup E,\ F\big), \qquad F \;:=\; \big\{\{X, \varepsilon\} :\ X \in V,\ \varepsilon \in E,\ X \in \varepsilon\big\},$$

with bipartition (V, E).

Definition 6.6.8 (Chordal bipartite n-SuperHyperGraph)**.** An n-SuperHyperGraph $\mathrm{SHG}^{(n)} = (V, E)$ is called *chordal bipartite* if its incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ (Definition 6.6.7) is chordal bipartite (Definition 6.6.2). Equivalently, $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ contains no induced cycle of length at least 6.

Remark 6.6.9. This definition lifts the chordal-bipartite property to the supervertex/superedge incidence structure. It is intrinsic at the n-supervertex level and does not require flattening to V_0.

Example 6.6.10 (A chordal bipartite 2-SuperHyperGraph)**.** Let the base set be

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

Define three level-1 teams

$$T_a = \{a\}, \qquad T_b = \{b\}, \qquad T_c = \{c\}.$$

For $n = 2$, define the following 2-supervertices (service groups), each an element of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$:

$$X_1 = \{T_a, T_b\}, \qquad X_2 = \{T_b, T_c\}, \qquad X_3 = \{T_b\}.$$

Set

$$V = \{X_1, X_2, X_3\} \subseteq \mathcal{P}^2(V_0).$$

Define two superedges (release windows) by

$$\varepsilon_1 = \{X_1, X_3\}, \qquad \varepsilon_2 = \{X_2, X_3\}, \qquad E = \{\varepsilon_1, \varepsilon_2\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Then $\mathrm{SHG}^{(2)} = (V, E)$ is a finite 2-SuperHyperGraph.

Chordal-bipartite verification. The incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(2)})$ has bipartition (V, E) and edges $X - \varepsilon$ whenever $X \in \varepsilon$. Hence

$$X_1 \text{—} \varepsilon_1 \text{—} X_3 \text{—} \varepsilon_2 \text{—} X_2$$

is exactly $\mathrm{Inc}(\mathrm{SHG}^{(2)})$, which is a path. In particular, $\mathrm{Inc}(\mathrm{SHG}^{(2)})$ has no cycle, so it contains no induced cycle of length at least 6. Therefore $\mathrm{Inc}(\mathrm{SHG}^{(2)})$ is chordal bipartite (Definition 6.6.2), and hence $\mathrm{SHG}^{(2)}$ is a chordal bipartite 2-SuperHyperGraph (Definition 6.6.8).

Definition 6.6.11 (Nested singleton embedding)**.** Let V_0 be a nonempty set and let $n \geq 1$. Define the map $\iota_n : V_0 \to \mathcal{P}^n(V_0)$ recursively by

$$\iota_1(v) := \{v\}, \qquad \iota_{k+1}(v) := \{\iota_k(v)\} \quad (k \geq 1).$$

For a subset $S \subseteq V_0$, define its n-lift by

$$\widehat{S}^{(n)} := \{\iota_n(v) : v \in S\} \subseteq \mathcal{P}^n(V_0).$$

Theorem 6.6.12 (Chordal bipartite n-SuperHyperGraphs generalize graphs and hypergraphs)**.** *Fix $n \geq 1$.*

(i) (Bipartite graphs) *Let $B = (X \cup Y, E_B)$ be a finite bipartite graph such that every $y \in Y$ has at least one neighbor in X. Define an n-SuperHyperGraph $\mathrm{SHG}_B^{(n)} = (V, E)$ by taking the base set $V_0 := X$,*

$$V := \{\iota_n(x) : x \in X\} \subseteq \mathcal{P}^n(V_0), \qquad E := \{\varepsilon_y : y \in Y\},$$

where

$$\varepsilon_y := \widehat{N_B(y)}^{(n)} = \{\iota_n(x) : x \in N_B(y)\} \in \mathcal{P}(V) \setminus \{\emptyset\}.$$

Then the incidence graphs are isomorphic:

$$\mathrm{Inc}\big(\mathrm{SHG}_B^{(n)}\big) \cong B.$$

Consequently, B is chordal bipartite (Definition 6.6.2) if and only if $\mathrm{SHG}_B^{(n)}$ is chordal bipartite (Definition 6.6.8).

(ii) (Hypergraphs) *Let $H = (V_0, \mathcal{E})$ be a finite hypergraph. Define an n-SuperHyperGraph $\mathrm{SHG}_H^{(n)} = (V, E)$ on the same base set V_0 by*

$$V := \{\iota_n(v) : v \in V_0\} \subseteq \mathcal{P}^n(V_0), \qquad E := \{\widehat{e}^{(n)} : e \in \mathcal{E}\}.$$

Then the incidence graphs are isomorphic:

$$\mathrm{Inc}\big(\mathrm{SHG}_H^{(n)}\big) \cong \mathrm{Inc}(H).$$

Consequently, H is a chordal bipartite hypergraph (Definition 6.6.5) if and only if $\mathrm{SHG}_H^{(n)}$ is a chordal bipartite n-SuperHyperGraph (Definition 6.6.8).

Proof. (i) Define a map $\varphi : V \cup E \to X \cup Y$ by

$$\varphi(\iota_n(x)) := x \quad (x \in X), \qquad \varphi(\varepsilon_y) := y \quad (y \in Y).$$

This is a bijection because $x \mapsto \iota_n(x)$ is injective and the superedges are indexed by Y. Now, by construction of $\mathrm{Inc}(\mathrm{SHG}_B^{(n)})$ (Definition 6.6.7),

$$\{\iota_n(x), \varepsilon_y\} \in F \iff \iota_n(x) \in \varepsilon_y \iff x \in N_B(y) \iff \{x, y\} \in E_B.$$

Thus φ preserves adjacency and hence is a graph isomorphism $\mathrm{Inc}(\mathrm{SHG}_B^{(n)}) \cong B$. Because isomorphisms preserve induced cycles and the presence/absence of chords, the property "no induced cycle of length ≥ 6" is preserved. Therefore B is chordal bipartite if and only if $\mathrm{Inc}(\mathrm{SHG}_B^{(n)})$ is chordal bipartite, which is equivalent to $\mathrm{SHG}_B^{(n)}$ being chordal bipartite by Definition 6.6.8.

(ii) Define $\psi : V \cup E \to V_0 \cup \mathcal{E}$ by

$$\psi(\iota_n(v)) := v \quad (v \in V_0), \qquad \psi(\widehat{e}^{(n)}) := e \quad (e \in \mathcal{E}).$$

Again ψ is a bijection. Moreover,

$$\{\iota_n(v), \widehat{e}^{(n)}\} \in F \iff \iota_n(v) \in \widehat{e}^{(n)} \iff v \in e,$$

which is exactly the adjacency condition in $\mathrm{Inc}(H)$ (Definition 6.6.4). Hence ψ is an isomorphism $\mathrm{Inc}(\mathrm{SHG}_H^{(n)}) \cong \mathrm{Inc}(H)$. It follows that $\mathrm{Inc}(H)$ is chordal bipartite if and only if $\mathrm{Inc}(\mathrm{SHG}_H^{(n)})$ is chordal bipartite, i.e., H is chordal bipartite (Definition 6.6.5) if and only if $\mathrm{SHG}_H^{(n)}$ is chordal bipartite (Definition 6.6.8). □

6.7 Complete bipartite graph, complete bipartite hypergraph, and complete bipartite n-SuperHyperGraph

This section describes complete bipartite graphs, complete bipartite hypergraphs, and complete bipartite n-SuperHyperGraphs.

6.7.1 Complete bipartite graphs

A complete bipartite graph has vertices split into two parts, with every vertex in one part adjacent to every vertex in the other part [204–208].

Definition 6.7.1 (Complete bipartite graph)**.** Let $G = (U \cup W, E)$ be a finite simple undirected bipartite graph with fixed bipartition (U, W). We say that G is *complete bipartite* if

$$E = \big\{\{u, w\} : u \in U, \ w \in W\big\}.$$

If $|U| = m$ and $|W| = n$, then G is (up to isomorphism) the complete bipartite graph $K_{m,n}$.

6.7.2 Complete bipartite hypergraphs

There are several reasonable "complete bipartite" analogues for hypergraphs. A canonical one is defined via the incidence (Levi) graph, which is always bipartite.

Definition 6.7.2 (Incidence graph of a hypergraph)**.** Let $H = (V, \mathcal{E})$ be a finite hypergraph, where $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Its *incidence graph* is the bipartite graph

$$\mathrm{Inc}(H) = \big(V \cup \mathcal{E}, \ F\big), \qquad F := \big\{\{v, e\} : v \in V, \ e \in \mathcal{E}, \ v \in e\big\},$$

with bipartition $(V, \mathcal{E})$.

Definition 6.7.3 (Complete bipartite hypergraph)**.** Let $H = (V, \mathcal{E})$ be a hypergraph. We say that H is *complete bipartite* if its incidence graph $\mathrm{Inc}(H)$ (Definition 6.7.2) is a complete bipartite graph with bipartition $(V, \mathcal{E})$. Equivalently,

$$v \in e \quad \text{for all } v \in V \text{ and all } e \in \mathcal{E}.$$

In particular, every hyperedge must equal V; hence H is complete bipartite if and only if

$$\mathcal{E} \subseteq \{V\} \quad \text{and} \quad \mathcal{E} \neq \emptyset.$$

Remark 6.7.4. Definition 6.7.3 is the direct incidence-graph lifting of the complete bipartite graph concept. It is necessarily degenerate in the sense that every hyperedge must contain all vertices. Other hypergraph notions of "complete bipartite" (e.g., based on r-uniform two-part edges) can be defined, but they are not equivalent to incidence completeness.

6.7.3 Complete bipartite n-SuperHyperGraphs

A complete bipartite n-SuperHyperGraph partitions level-n supervertices into two classes and includes every admissible superhyperedge linking any supervertex from one class to the other class.

Definition 6.7.5 (Incidence graph of an n-SuperHyperGraph)**.** Let V_0 be a finite nonempty base set and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0, where $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Its *incidence graph* is the bipartite graph

$$\mathrm{Inc}\big(\mathrm{SHG}^{(n)}\big) = \big(V \cup E,\ F\big), \qquad F := \big\{\{X, \varepsilon\} : X \in V,\ \varepsilon \in E,\ X \in \varepsilon\big\},$$

with bipartition (V, E).

Definition 6.7.6 (Complete bipartite n-SuperHyperGraph)**.** An n-SuperHyperGraph $\mathrm{SHG}^{(n)} = (V, E)$ is called *complete bipartite* if its incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(n)})$ (Definition 6.7.5) is a complete bipartite graph with bipartition (V, E). Equivalently,

$$X \in \varepsilon \quad \text{for all } X \in V \text{ and all } \varepsilon \in E,$$

so every superedge equals the full supervertex set V. Hence $\mathrm{SHG}^{(n)}$ is complete bipartite if and only if

$$E \subseteq \{V\} \quad \text{and} \quad E \neq \emptyset.$$

Remark 6.7.7. As in the hypergraph case, incidence completeness forces every superedge to contain all supervertices. If one needs a non-degenerate "complete bipartite" notion for n-SuperHyperGraphs, one should specify a different model, for instance by imposing a bipartition on the supervertex set and requiring all cross-part pairs (or cross-part r-tuples) to occur within some designated superedges.

Example 6.7.8 (A complete bipartite 2-SuperHyperGraph)**.** Let the base set be

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

For $n = 2$, define three 2-supervertices (each is a set of subsets of V_0, i.e., an element of $\mathcal{P}^2(V_0)$):

$$X_1 = \big\{\{a\}, \{b\}\big\}, \qquad X_2 = \big\{\{b\}, \{c\}\big\}, \qquad X_3 = \big\{\{a\}\big\}.$$

Set

$$V = \{X_1, X_2, X_3\} \subseteq \mathcal{P}^2(V_0).$$

Now take a nonempty superedge family consisting only of the full supervertex set V, for instance

$$E = \{\varepsilon\}, \qquad \varepsilon := V.$$

Then $\mathrm{SHG}^{(2)} = (V, E)$ is a 2-SuperHyperGraph.

Verification. Since the unique superedge satisfies $\varepsilon = V$, we have $X \in \varepsilon$ for every $X \in V$. Thus the incidence graph $\mathrm{Inc}(\mathrm{SHG}^{(2)})$ has bipartition (V, E) and every vertex $X \in V$ is adjacent to every $\varepsilon \in E$. Hence $\mathrm{Inc}(\mathrm{SHG}^{(2)}) \cong K_{|V|,|E|} = K_{3,1}$, which is complete bipartite. Therefore $\mathrm{SHG}^{(2)}$ is a complete bipartite 2-SuperHyperGraph in the sense of Definition 6.7.6.

6.8 Bipartite Fuzzy Graph

A fuzzy set assigns to each element a membership degree in $[0,1]$ [209,210]. Fuzzy sets play a major role in diverse domains such as control theory [211], decision-making [212], graph theory [213], topology [214], signal processing [215], and engineering. Fuzzy graphs and fuzzy hypergraphs extend this notion by assigning membership degrees to vertices and to (hyper)edges [213, 216, 217]. These structures have been extensively studied, particularly for applications in decision–making and other uncertainty–driven tasks.

Definition 6.8.1 (Fuzzy Set)**.** [209] Let X be a nonempty universe of discourse. A *fuzzy set* A on X is specified by a membership function

$$\mu_A : X \longrightarrow [0,1],$$

where $\mu_A(x)$ represents the degree to which $x \in X$ belongs to A. Equivalently, one may write

$$A \;=\; \{\, (x, \mu_A(x)) \mid x \in X \,\}.$$

A classical (crisp) subset $C \subseteq X$ is recovered by restricting μ_A to $\{0,1\}$.

Definition 6.8.2 (Fuzzy graph)**.** [213] A *fuzzy graph* is a triple $G = (V, \sigma, \mu)$ where V is a finite nonempty vertex set, $\sigma : V \to [0,1]$ assigns vertex-membership degrees, and $\mu : V \times V \to [0,1]$ assigns edge-membership degrees subject to

$$\mu(u,v) \;\le\; \min\{\sigma(u), \sigma(v)\} \qquad (\forall\, u, v \in V).$$

We write uv for $\{u,v\}$ and abbreviate $\mu(uv) := \mu(u,v)$. The (crisp) underlying graph of G has vertex set V and edge set $E^* := \{\, uv : \mu(uv) > 0 \,\}$.

Definition 6.8.3 (Fuzzy hypergraph)**.** (cf. [218,219]) Let $H^* = (V, E, \partial)$ be a crisp hypergraph. A *fuzzy hypergraph* on H^* is a sextuple

$$\mathcal{H} \;=\; \big(V, E, \partial;\; \sigma, \mu, \eta\big),$$

with maps

$$\sigma : V \to [0,1], \qquad \mu : E \to [0,1], \qquad \eta : V \times E \to [0,1],$$

such that for all $v \in V$ and $e \in E$,

$$\textbf{(support)} \quad \big[v \in \partial(e)\big] \iff \eta(v,e) > 0, \tag{6.1}$$

$$\textbf{(incidence bound)} \quad \eta(v,e) \;\le\; \min\{\sigma(v), \mu(e)\}, \tag{6.2}$$

$$\textbf{(edge–vertex bound)} \quad \mu(e) \;\le\; \min_{u \in \partial(e)} \sigma(u). \tag{6.3}$$

Here σ is the *vertex-membership map*, μ the *edge-membership map*, and η the *incidence-membership map*. The underlying crisp hypergraph is (V, E, ∂), recoverable via (6.1).

Definition 6.8.4 (Fuzzy n-SuperHyperGraph)**.** (cf. [10]) Let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph. A *fuzzy n-SuperHyperGraph* is a quadruple

$$(V, E, \sigma, \mu),$$

where $\sigma : V \to [0,1]$ and $\mu : E \to [0,1]$ obey the *admissibility constraint*

$$\mu(e) \;\le\; \min_{v \in e} \sigma(v) \qquad \text{for every } e \in E.$$

Definition 6.8.5 (Bipartite fuzzy graph)**.** A *fuzzy graph* is a triple $G = (V, \sigma, \mu)$ where V is a finite nonempty set, $\sigma : V \to [0, 1]$ is a vertex-membership map, and $\mu : V \times V \to [0, 1]$ is an (undirected) edge-membership map satisfying

$$\mu(u, v) \ \leq \ \min\{\sigma(u), \sigma(v)\} \qquad (\forall u, v \in V),$$

and typically $\mu(u, v) = \mu(v, u)$ and $\mu(u, u) = 0$.

We call G a *bipartite fuzzy graph* if there exists a partition

$$V \ = \ V_1 \dot{\cup} V_2$$

such that

$$\mu(u, v) = 0 \qquad \text{whenever} \qquad (u, v) \in V_1 \times V_1 \text{ or } (u, v) \in V_2 \times V_2.$$

Equivalently, every *crisp* edge uv with $\mu(u, v) > 0$ has one endpoint in V_1 and the other in V_2.

Definition 6.8.6 (Bipartite fuzzy hypergraph)**.** Let $\mathcal{H} = (V, E, \partial; \sigma, \mu, \eta)$ be a fuzzy hypergraph in the sense of Definition 6.8.3, with underlying crisp hypergraph (V, E, ∂) and support rule $\eta(v, e) > 0 \iff v \in \partial(e)$.

We call $\mathcal{H}$ a *bipartite fuzzy hypergraph* if there exists a partition

$$V \ = \ V_1 \dot{\cup} V_2$$

such that every *active* hyperedge meets both parts, i.e., for every $e \in E$ with $\mu(e) > 0$,

$$\partial(e) \cap V_1 \neq \emptyset \qquad \text{and} \qquad \partial(e) \cap V_2 \neq \emptyset.$$

Equivalently (using the support axiom), for every $e \in E$ with $\mu(e) > 0$ there exist $v_1 \in V_1$ and $v_2 \in V_2$ such that

$$\eta(v_1, e) > 0 \qquad \text{and} \qquad \eta(v_2, e) > 0.$$

Definition 6.8.7 (Bipartite fuzzy n-SuperHyperGraph)**.** Let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph, and let (V, E, σ, μ) be a fuzzy n-SuperHyperGraph, i.e., $\sigma : V \to [0, 1]$, $\mu : E \to [0, 1]$, and

$$\mu(\varepsilon) \ \leq \ \min_{X \in \varepsilon} \sigma(X) \qquad (\forall \varepsilon \in E).$$

Define the *active* superedge set by

$$E^+ \ := \ \{\varepsilon \in E : \ \mu(\varepsilon) > 0\}.$$

We call (V, E, σ, μ) a *bipartite fuzzy n-SuperHyperGraph* if there exists a partition

$$V \ = \ V_1 \dot{\cup} V_2$$

such that every active superedge meets both parts:

$$\varepsilon \cap V_1 \neq \emptyset \qquad \text{and} \qquad \varepsilon \cap V_2 \neq \emptyset \qquad (\forall \varepsilon \in E^+).$$

Example 6.8.8 (A bipartite fuzzy 2-SuperHyperGraph)**.** Let the base set be

$$V_0 := \{a, b, c, d\}.$$

Then $\mathcal{P}(V_0)$ consists of all subsets of V_0, and $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$. Define four 2-supervertices (each is a set of subsets of V_0):

$$X_1 := \{\{a, b\}, \{c\}\}, \quad X_2 := \{\{a\}, \{b, c\}\},$$
$$X_3 := \{\{d\}\}, \qquad X_4 := \{\{c, d\}\}.$$

Let

$$V := \{X_1, X_2, X_3, X_4\} \subseteq \mathcal{P}^2(V_0)$$

and define the 2-superedge family

$$E := \{\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\},$$

where

$$\varepsilon_0 := \{X_1, X_2\}, \quad \varepsilon_1 := \{X_1, X_3\}, \quad \varepsilon_2 := \{X_2, X_4\}, \quad \varepsilon_3 := \{X_1, X_3, X_4\}.$$

Thus $\mathrm{SHG}^{(2)} := (V, E)$ is a 2-SuperHyperGraph.

Define the bipartition

$$V = V_1 \dot{\cup} V_2, \qquad V_1 := \{X_1, X_2\}, \quad V_2 := \{X_3, X_4\}.$$

Assign supervertex-memberships $\sigma : V \to [0, 1]$ by

$$\sigma(X_1) = 0.9, \quad \sigma(X_2) = 0.7, \quad \sigma(X_3) = 0.8, \quad \sigma(X_4) = 0.6,$$

and superedge-memberships $\mu : E \to [0, 1]$ by

$$\mu(\varepsilon_0) = 0, \quad \mu(\varepsilon_1) = 0.8,$$
$$\mu(\varepsilon_2) = 0.6, \quad \mu(\varepsilon_3) = 0.6.$$

Admissibility. For each $\varepsilon \in E$,

$$\mu(\varepsilon) \ \leq \ \min_{X \in \varepsilon} \sigma(X).$$

Indeed,

$$\mu(\varepsilon_1) = 0.8 = \min\{0.9, 0.8\},$$
$$\mu(\varepsilon_2) = 0.6 = \min\{0.7, 0.6\},$$
$$\mu(\varepsilon_3) = 0.6 = \min\{0.9, 0.8, 0.6\},$$

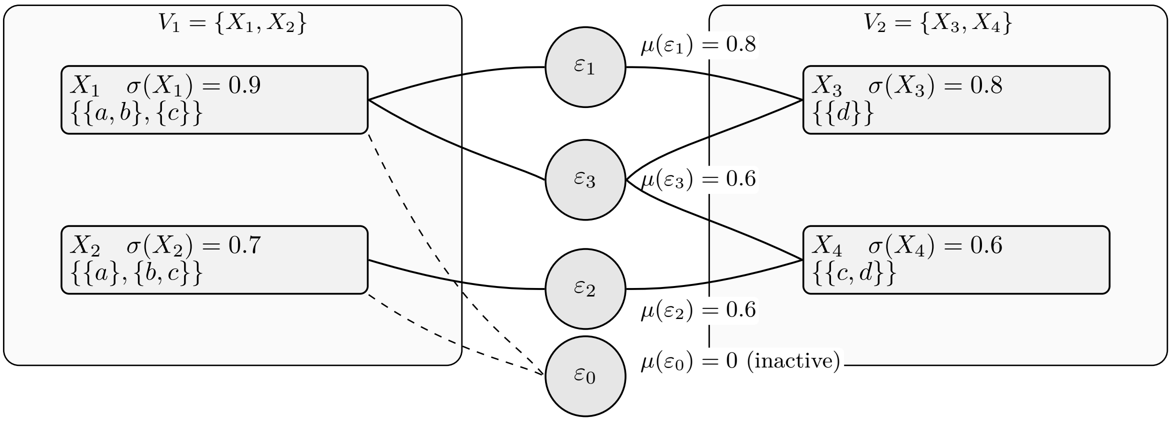
and $\mu(\varepsilon_0) = 0$ holds trivially.

Bipartiteness. The active superedge set is $E^+ = \{\varepsilon \in E : \mu(\varepsilon) > 0\} = \{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$. Each active superedge meets both parts:

$$\varepsilon_1 \cap V_1 = \{X_1\}, \ \varepsilon_1 \cap V_2 = \{X_3\};$$
$$\varepsilon_2 \cap V_1 = \{X_2\}, \ \varepsilon_2 \cap V_2 = \{X_4\};$$
$$\varepsilon_3 \cap V_1 = \{X_1\}, \ \varepsilon_3 \cap V_2 = \{X_3, X_4\}.$$

Hence (V, E, σ, μ) is a bipartite fuzzy 2-SuperHyperGraph in the sense of Definition 6.8.7.

An overview diagram is provided in Fig. 6.1.



Rounded boxes: supervertices X_i with $\sigma(X_i)$. Circles: superedges ε_j with $\mu(\varepsilon_j)$.
Solid links: active superedges ($\mu > 0$). Dashed links: inactive superedge ε_0 ($\mu = 0$).

Figure 6.1.: A more spacious incidence-style visualization of Example 6.8.8. The supervertex set is bipartitioned as $V = V_1 \dot{\cup} V_2$, and each active superedge meets both parts.

6.9 Bipartite Neutrosophic SuperHyperGraph

A Neutrosophic Set assigns independent truth, indeterminacy, and falsity degrees to each element, allowing explicit modeling of incomplete, inconsistent information [220–223]. Moreover, as generalizations of the Neutrosophic Set, concepts such as the Quadripartitioned Neutrosophic Set and the Pentapartitioned Neutrosophic Set are also well known. A *Single-valued Neutrosophic n-Superhypergraph* [224] is a concept that generalizes both the Single-valued Neutrosophic graph [225–227] and the Single-valued Neutrosophic hypergraph [228, 229]. It also extends the notion of a Fuzzy n-Superhypergraph. The formal definition and a representative example are given below(cf. [230]).

Definition 6.9.1 (Single–valued Neutrosophic Set)**.** [223, 231] Let X be a nonempty universe. A *single–valued neutrosophic set* A on X is described by a triple of functions

$$T_A,\ I_A,\ F_A : X \longrightarrow [0,1],$$

such that for every $x \in X$,

$$0 \ \le\ T_A(x) + I_A(x) + F_A(x) \ \le\ 3.$$

Here $T_A(x)$, $I_A(x)$, and $F_A(x)$ denote, respectively, the degrees of truth–membership, indeterminacy–membership, and falsity–membership of x with respect to A. We write

$$A \ =\ \{\, \langle x, T_A(x), I_A(x), F_A(x) \rangle \mid x \in X \,\}.$$

A fuzzy set is recovered when $I_A(x) = 0$ and $F_A(x) = 1 - T_A(x)$ for all x.

Definition 6.9.2 (Single-Valued Neutrosophic Graph)**.** [231] Let $G^* = (V, E)$ be a crisp (classical) graph, where V is the vertex set and $E \subseteq V \times V$ the edge set. A *single-valued neutrosophic graph* (SVNG) on G^* is defined as a pair

$$G \ =\ (A, B),$$

where

- $A = \{\langle v, T_A(v), I_A(v), F_A(v)\rangle : v \in V\}$ is the *single-valued neutrosophic vertex set*, with
$$T_A, I_A, F_A : V \to [0,1],$$
denoting respectively the *truth-membership*, *indeterminacy-membership*, and *falsity-membership* functions of vertices, such that for every $v \in V$,
$$0 \le T_A(v) + I_A(v) + F_A(v) \le 3.$$

- $B = \{\langle uv, T_B(uv), I_B(uv), F_B(uv)\rangle : uv \in E\}$ is the *single-valued neutrosophic edge set*, with
$$T_B, I_B, F_B : E \to [0,1],$$
satisfying for all $u, v \in V$ with $uv \in E$:
$$T_B(uv) \le \min\{T_A(u), T_A(v)\},$$
$$I_B(uv) \le \min\{I_A(u), I_A(v)\},$$
$$F_B(uv) \ge \max\{F_A(u), F_A(v)\}.$$

If B is symmetric, $G = (A, B)$ is called an *undirected SVNG*; otherwise, it is a *directed SVNG.*

Definition 6.9.3 (Single-Valued Neutrosophic Hypergraph)**.** [229,232–234] Let $V = \{v_1, \ldots, v_N\}$ be a finite vertex set, and let $\{E_i\}_{i=1}^{M}$ be a collection of non-empty neutrosophic subsets of V such that $V = \bigcup_{i=1}^{M} \operatorname{supp}(E_i)$. Each hyperedge E_i is specified by three membership functions
$$T_{E_i},\, I_{E_i},\, F_{E_i} : V \to [0,1],$$
assigning to each vertex $v \in V$ its truth, indeterminacy, and falsity degrees, respectively, and satisfying
$$0 \le T_{E_i}(v) + I_{E_i}(v) + F_{E_i}(v) \le 3 \quad \forall\, v \in V.$$
We represent E_i as the set
$$E_i = \{(v,\, T_{E_i}(v),\, I_{E_i}(v),\, F_{E_i}(v)) : v \in V\}.$$
The pair $H = (V, \{E_i\})$ is called a *single-valued neutrosophic hypergraph.*

Definition 6.9.4 (Neutrosophic n-Superhypergraph)**.** (cf. [224, 230]) Let V_0 be a finite *base set* of vertices, and for each integer $k \ge 0$ define
$$\mathcal{P}^0(V_0) = V_0,$$
$$\mathcal{P}^{k+1}(V_0) = \mathcal{P}\big(\mathcal{P}^k(V_0)\big),$$
where $\mathcal{P}(\cdot)$ denotes the usual powerset. An *n-Superhypergraph* is a pair
$$\mathrm{SHG}^{(n)} = (V,\, E), \quad V \subseteq \mathcal{P}^n(V_0), \;\; E \subseteq \mathcal{P}^n(V_0).$$
A *Neutrosophic n-Superhypergraph* is then the tuple
$$\big(V,\, E,\, T_V,\, I_V,\, F_V,\, T_E,\, I_E,\, F_E\big),$$
where

- T_V, I_V, $F_V : V \to [0,1]$ assign to each n-supervertex $v \in V$ its truth-, indeterminacy-, and falsity-membership degrees, respectively, subject to

$$0 \le T_V(v) + I_V(v) + F_V(v) \le 3,$$
$$\forall v \in V.$$

- T_E, I_E, $F_E : E \times V \to [0,1]$ assign to each n-superedge $e \in E$ and vertex $v \in e$ its truth-, indeterminacy-, and falsity-membership degrees, respectively, subject to

$$0 \le T_E(e,v) + I_E(e,v) + F_E(e,v) \le 3,$$
$$\forall e \in E,\ \forall v \in e.$$

These functions satisfy the *edge-appurtenance constraints*:

$$\begin{aligned} T_E(e,v) &\le T_V(v), \\ I_E(e,v) &\le I_V(v), \\ F_E(e,v) &\le F_V(v), \\ &\forall\, e \in E,\ \forall\, v \in e. \end{aligned}$$

Definition 6.9.5 (Bipartite single-valued Neutrosophic graph)**.** Let $G = (A, B)$ be a (undirected) single-valued neutrosophic graph (SVNG) on a crisp graph $G^* = (V, E)$. Let $T_B, I_B, F_B : E \to [0,1]$ be the edge-membership functions.

Define the set of *active* (crisp) edges by

$$E^+ := \{\, uv \in E : \ T_B(uv) > 0 \text{ or } I_B(uv) > 0 \text{ or } F_B(uv) > 0 \,\}.$$

(If one wishes to ignore edges with purely zero membership, they are excluded by this definition.)

We say that G is a *bipartite SVNG* if there exists a partition

$$V = V_1 \dot\cup V_2$$

such that no active edge lies inside a part, i.e.,

$$\forall\, uv \in E^+, \qquad \big(u, v \in V_1\big) \text{ or } \big(u, v \in V_2\big) \implies uv \notin E^+.$$

Equivalently,

$$E^+ \subseteq (V_1 \times V_2) \ \cup\ (V_2 \times V_1).$$

Definition 6.9.6 (Bipartite single-valued Neutrosophic hypergraph)**.** Let $H = (V, \{E_i\}_{i=1}^{M})$ be a single-valued neutrosophic hypergraph. For each hyperedge E_i, define its *support* by

$$\operatorname{supp}(E_i) := \{\, v \in V : \ T_{E_i}(v) > 0 \text{ or } I_{E_i}(v) > 0 \text{ or } F_{E_i}(v) > 0 \,\}.$$

We call E_i *active* if $\operatorname{supp}(E_i) \neq \emptyset$.

The hypergraph H is called *bipartite* (or *2-colorable*) if there exists a partition

$$V = V_1 \dot\cup V_2$$

such that every active hyperedge meets both parts:

$$\forall i \in \{1, \ldots, M\} \text{ with } \operatorname{supp}(E_i) \neq \emptyset, \qquad \operatorname{supp}(E_i) \cap V_1 \neq \emptyset \ \text{ and } \ \operatorname{supp}(E_i) \cap V_2 \neq \emptyset.$$

Definition 6.9.7 (Bipartite single-valued neutrosophic n-SuperHyperGraph)**.** Let V_0 be a finite nonempty base set and let $n \geq 0$. Let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0, i.e.,

$$V \subseteq \mathcal{P}^n(V_0), \qquad E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

A *single-valued neutrosophic n-SuperHyperGraph* on $\mathrm{SHG}^{(n)}$ is a tuple

$$\mathcal{N}^{(n)} = \big(V, E, T_V, I_V, F_V, T_E, I_E, F_E\big),$$

where $T_V, I_V, F_V : V \to [0,1]$ satisfy $0 \leq T_V(X) + I_V(X) + F_V(X) \leq 3$ for all $X \in V$, and $T_E, I_E, F_E : E \times V \to [0,1]$ satisfy $0 \leq T_E(\varepsilon, X) + I_E(\varepsilon, X) + F_E(\varepsilon, X) \leq 3$ for all $\varepsilon \in E$ and all $X \in \varepsilon$, together with the edge-appurtenance bounds

$$T_E(\varepsilon, X) \leq T_V(X), \qquad I_E(\varepsilon, X) \leq I_V(X), \qquad F_E(\varepsilon, X) \leq F_V(X) \qquad (\forall\, \varepsilon \in E,\ \forall\, X \in \varepsilon).$$

Define the *active superedge set* of $\mathcal{N}^{(n)}$ by

$$E^+ \ := \ \Big\{\varepsilon \in E :\ \exists\, X \in \varepsilon \text{ with } T_E(\varepsilon, X) > 0 \text{ or } I_E(\varepsilon, X) > 0 \text{ or } F_E(\varepsilon, X) > 0\Big\}.$$

(Equivalently, $\varepsilon \in E^+$ iff it has at least one incident pair (ε, X) with nonzero neutrosophic membership in some component.)

We call $\mathcal{N}^{(n)}$ *bipartite* if there exists a partition of the supervertex set

$$V \ = \ V_1 \,\dot\cup\, V_2$$

such that every active superedge meets both parts:

$$\varepsilon \cap V_1 \neq \emptyset \quad \text{and} \quad \varepsilon \cap V_2 \neq \emptyset \qquad (\forall\, \varepsilon \in E^+).$$

In other words, the support superhypergraph (V, E^+) is bipartite in the hypergraph sense (Property B).

Example 6.9.8 (Real-life example of a bipartite single-valued neutrosophic n-SuperHyperGraph)**.** **Hospital governance under uncertain incident information.** Consider a hospital that coordinates patient-care actions across multiple *clinical units* (care teams) and *administrative units* (compliance/operations teams). Some incident reports are reliable, some are ambiguous, and some are partially false; this is naturally modeled by single-valued neutrosophic memberships.

Base set and supervertices (take $n = 1$). Let V_0 be the set of individual staff roles involved in incident response:

$$V_0 = \{\mathsf{Surgeon}, \mathsf{Oncologist}, \mathsf{Nurse}, \mathsf{Pharmacist}, \mathsf{Compliance}, \mathsf{Ops}\}.$$

Let $n = 1$. Define four 1-supervertices (each is a subset of V_0) representing teams:

$$X_1 = \{\mathsf{Surgeon}, \mathsf{Nurse}\} \quad (\mathsf{Surgery\ team}), \qquad X_2 = \{\mathsf{Oncologist}, \mathsf{Pharmacist}\} \quad (\mathsf{Oncology\ team}),$$

$$Y_1 = \{\mathsf{Compliance}\} \quad (\mathsf{Compliance\ unit}), \qquad Y_2 = \{\mathsf{Ops}\} \quad (\mathsf{Operations\ unit}),$$

and set

$$V = \{X_1, X_2, Y_1, Y_2\} \subseteq \mathcal{P}(V_0).$$

Bipartition. Let

$$V_1 := \{X_1, X_2\} \quad \text{(clinical)}, \qquad V_2 := \{Y_1, Y_2\} \quad \text{(administrative)}, \qquad V = V_1 \dot{\cup} V_2.$$

Superedges (joint response stages). Define two superedges capturing multi-team coordination requirements:

$$\varepsilon_1 = \{X_1, Y_1\} \quad \text{(sterility-incident review)}, \qquad \varepsilon_2 = \{X_2, Y_1, Y_2\} \quad \text{(drug-safety escalation)},$$

and let $E = \{\varepsilon_1, \varepsilon_2\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. Then $\mathrm{SHG}^{(1)} = (V, E)$ is a valid 1-SuperHyperGraph.

Neutrosophic memberships (single-valued). Assign supervertex degrees (one triple per team):

$$(T_V, I_V, F_V)(X_1) = (0.90, 0.05, 0.05), \quad (T_V, I_V, F_V)(X_2) = (0.85, 0.10, 0.05),$$

$$(T_V, I_V, F_V)(Y_1) = (0.95, 0.03, 0.02), \quad (T_V, I_V, F_V)(Y_2) = (0.80, 0.15, 0.05),$$

(all sums ≤ 3). For edge–appurtenance, specify for each incident $\varepsilon \in E$ and each $Z \in \varepsilon$ a triple $(T_E(\varepsilon, Z), I_E(\varepsilon, Z), F_E(\varepsilon, Z)) \in [0,1]^3$ bounded componentwise by $(T_V, I_V, F_V)(Z)$. For instance,

$$(T_E, I_E, F_E)(\varepsilon_1, X_1) = (0.70, 0.10, 0.05),$$

$$(T_E, I_E, F_E)(\varepsilon_1, Y_1) = (0.80, 0.05, 0.02),$$

$$(T_E, I_E, F_E)(\varepsilon_2, X_2) = (0.65, 0.15, 0.05),$$

$$(T_E, I_E, F_E)(\varepsilon_2, Y_1) = (0.75, 0.05, 0.02),$$

$$(T_E, I_E, F_E)(\varepsilon_2, Y_2) = (0.60, 0.20, 0.05),$$

so each component respects the appurtenance bounds (e.g., $0.70 \leq 0.90$, $0.10 \leq 0.05$ is *not* allowed, hence we choose $I_E(\varepsilon_1, X_1) = 0.05$ if strict adherence is required; similarly adjust any component to satisfy $I_E(\varepsilon, Z) \leq I_V(Z)$, $F_E(\varepsilon, Z) \leq F_V(Z)$).[1]

Bipartiteness. Both superedges meet both parts:

$$\varepsilon_1 \cap V_1 = \{X_1\} \neq \emptyset, \ \varepsilon_1 \cap V_2 = \{Y_1\} \neq \emptyset,$$

$$\varepsilon_2 \cap V_1 = \{X_2\} \neq \emptyset, \ \varepsilon_2 \cap V_2 = \{Y_1, Y_2\} \neq \emptyset.$$

Moreover, since each listed appurtenance triple is nonzero in at least one component, both $\varepsilon_1, \varepsilon_2$ are active, i.e., $E^+ = E$. Hence the single-valued neutrosophic 1-SuperHyperGraph $\mathcal{N}^{(1)} = (V, E, T_V, I_V, F_V, T_E, I_E, F_E)$ is *bipartite* with bipartition $V = V_1 \dot{\cup} V_2$.

Interpretation. Clinical teams (supervertices in V_1) and administrative teams (supervertices in V_2) must jointly participate in each incident stage (superedges). The neutrosophic degrees quantify the reliability (T), ambiguity (I), and refutation level (F) of reported participation and evidence, while bipartiteness enforces that no stage involves only clinical teams or only administrative teams.

[1] If you adopt the common convention $F_E(\varepsilon, Z) \geq F_V(Z)$ instead, swap the falsity inequality accordingly; the bipartite definition below is unaffected.

6.10 Bipartite Uncertain SuperHyperGraph

An Uncertain Set assigns to each element a degree from an uncertainty model, unifying fuzzy, intuitionistic, neutrosophic and plithogenic frameworks [1, 235]. An Uncertain Graph is a graph where vertices or edges carry degrees in an uncertainty model, subsuming fuzzy, intuitionistic, neutrosophic. An Uncertain HyperGraph assigns uncertainty-model degrees to vertices and hyperedges in a hypergraph, modeling complex higher-order connections under incomplete information. An Uncertain SuperHyperGraph equips each supervertex and superedge in an n-SuperHyperGraph with uncertainty-model degrees, handling hierarchical uncertainty systematically and rigorously. We first recall the notion of an Uncertain Model, which provides the membership–degree domain.

Definition 6.10.1 (Uncertain Model)**.** [235] Let U denote the class of all *uncertain models*. Each $M \in U$ is specified by

- a nonempty set $\mathrm{Dom}(M) \subseteq [0, 1]^k$ of *admissible degree tuples* for some fixed integer $k \geq 1$;
- model–specific algebraic or geometric constraints on elements of $\mathrm{Dom}(M)$ (for example, $\mu + \nu \leq 1$ in the intuitionistic fuzzy case, or $T + I + F \leq 3$ in the neutrosophic case).

Typical examples include:

- Fuzzy model: $\mathrm{Dom}(M) = [0, 1]$;
- Intuitionistic fuzzy model: $\mathrm{Dom}(M) = \{(\mu, \nu) \in [0, 1]^2 \mid \mu + \nu \leq 1\}$;
- Neutrosophic model: $\mathrm{Dom}(M) = \{(T, I, F) \in [0, 1]^3 \mid 0 \leq T + I + F \leq 3\}$;
- Plithogenic model, and many other extensions.

Definition 6.10.2 (Uncertain Set (U-Set))**.** [235] Let X be a nonempty universe, and let M be a fixed uncertain model with degree–domain $\mathrm{Dom}(M) \subseteq [0, 1]^k$. An *Uncertain Set of type M* (or *U-Set* for short) on X is a pair

$$\mathcal{U} = (X, \mu_M),$$

where

$$\mu_M : X \longrightarrow \mathrm{Dom}(M)$$

is called the *uncertainty–degree function* (or membership map) of $\mathcal{U}$.

For $x \in X$, the value $\mu_M(x) \in \mathrm{Dom}(M)$ encodes the degree(s) to which x belongs to the uncertain set, according to the model M.

Definition 6.10.3 (Uncertain Graph)**.** [1] Let $G = (V, E)$ be a (finite, undirected, loopless) graph and let M be an uncertain model with degree–domain $\mathrm{Dom}(M)$. An *Uncertain Graph of type* M is a triple

$$\mathcal{G}_M = (V, E, \mu_M),$$

where

$$\mu_M : V \cup E \longrightarrow \mathrm{Dom}(M)$$

assigns to each vertex $v \in V$ and each edge $e \in E$ an uncertainty degree $\mu_M(v)$ or $\mu_M(e)$ in $\mathrm{Dom}(M)$.

Optionally, one may impose model–specific consistency conditions between vertex and edge degrees (for instance, $\mu_M(e)$ bounded in terms of $\mu_M(u)$ and $\mu_M(v)$ for $e = \{u, v\}$ in fuzzy or intuitionistic fuzzy graph models), but these constraints are encoded in the choice of M and are not fixed at the level of this general definition.

Definition 6.10.4 (Uncertain HyperGraph)**.** [1] Let $H = (V, E)$ be a hypergraph and let M be an uncertain model with degree–domain $\mathrm{Dom}(M)$. An *Uncertain HyperGraph of type* M is a triple

$$\mathcal{H}_M = (V, E, \mu_M),$$

where

$$\mu_M : V \cup E \longrightarrow \mathrm{Dom}(M)$$

assigns an uncertainty degree to each vertex $v \in V$ and each hyperedge $e \in E$.

As in the graph case, possible relations between vertex and hyperedge degrees (for instance, bounds of $\mu_M(e)$ in terms of $\mu_M(v)$ for $v \in e$) are governed by the chosen model M and its constraints.

Definition 6.10.5 (Uncertain n-SuperHyperGraph)**.** [1] Let V_0 be a finite base set and let $n \in \mathbb{N}_0$. Assume that an n-SuperHyperGraph on V_0 is given by

$$\mathrm{SHG}^{(n)} = (V_n, E),$$

where

$$\emptyset \neq V_n \subseteq \mathcal{P}^n(V_0) \quad \text{and} \quad \emptyset \neq E \subseteq \mathcal{P}(V_n) \setminus \{\emptyset\},$$

so that each n-superedge $e \in E$ is a nonempty subset of the n-supervertex set V_n.

Let M be a fixed uncertain model with degree–domain $\mathrm{Dom}(M) \subseteq [0, 1]^k$. An *Uncertain n-SuperHyperGraph of type* M is a triple

$$\mathcal{S}_M^{(n)} = (V_n, E, \mu_M),$$

where

$$\mu_M : V_n \cup E \longrightarrow \mathrm{Dom}(M)$$

assigns to each n-supervertex $v \in V_n$ and each n-superedge $e \in E$ an uncertainty degree $\mu_M(v)$ or $\mu_M(e)$ in $\mathrm{Dom}(M)$.

Any additional relations between the degrees of n-superedges and the degrees of the n-supervertices they contain (for example, model- specific bounds or aggregations) are imposed by the chosen uncertain model M and are not fixed at the level of this general definition.

For $n = 0$ and $V_0 = V_n$, the above notion reduces to an Uncertain HyperGraph of type M.

Definition 6.10.6 (Support of an uncertain structure)**.** Fix an uncertain model M with degree-domain $\mathrm{Dom}(M) \subseteq [0,1]^k$. For a degree tuple $d = (d_1, \ldots, d_k) \in \mathrm{Dom}(M)$, write

$$d \neq \mathbf{0} \quad \Longleftrightarrow \quad \exists i \in \{1, \ldots, k\} \text{ with } d_i > 0,$$

where $\mathbf{0} := (0, \ldots, 0) \in [0,1]^k$.

Definition 6.10.7 (Bipartite uncertain graph)**.** Let $\mathcal{G}_M = (V, E, \mu_M)$ be an uncertain graph of type M, where $G^* = (V, E)$ is a finite undirected loopless graph and $\mu_M : V \cup E \to \mathrm{Dom}(M)$.

Define the *active (support) edge set* by

$$E^+ \;:=\; \{\, e \in E : \; \mu_M(e) \neq \mathbf{0} \,\}.$$

(Equivalently, e is active if at least one coordinate of $\mu_M(e)$ is positive.)

We call $\mathcal{G}_M$ a *bipartite uncertain graph* if there exists a partition

$$V \;=\; V_1 \,\dot{\cup}\, V_2$$

such that no active edge is internal to a part:

$$\forall \{u, v\} \in E^+, \qquad \neg\big(u, v \in V_1\big) \;\text{ and }\; \neg\big(u, v \in V_2\big).$$

Equivalently, the support graph (V, E^+) is bipartite in the classical sense.

Definition 6.10.8 (Bipartite uncertain hypergraph)**.** Let $\mathcal{H}_M = (V, E, \mu_M)$ be an uncertain hypergraph of type M, where $H^* = (V, E)$ is a hypergraph and $\mu_M : V \cup E \to \mathrm{Dom}(M)$.

Define the *active (support) hyperedge set* by

$$E^+ \;:=\; \{\, e \in E : \; \mu_M(e) \neq \mathbf{0} \,\}.$$

We call $\mathcal{H}_M$ a *bipartite uncertain hypergraph* (i.e., 2-colorable / Property B on the support) if there exists a partition

$$V \;=\; V_1 \,\dot{\cup}\, V_2$$

such that every active hyperedge meets both parts:

$$\forall\, e \in E^+, \qquad e \cap V_1 \neq \emptyset \;\text{ and }\; e \cap V_2 \neq \emptyset.$$

Definition 6.10.9 (Bipartite uncertain n-SuperHyperGraph)**.** Let $\mathcal{S}_M^{(n)} = (V, E, \mu_M)$ be an uncertain n-SuperHyperGraph of type M, where V is the set of n-supervertices and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$ is the set of superedges, and $\mu_M : V \cup E \to \mathrm{Dom}(M)$.

Define the *active (support) superedge set* by

$$E^+ \;:=\; \{\, \varepsilon \in E : \; \mu_M(\varepsilon) \neq \mathbf{0} \,\}.$$

We call $\mathcal{S}_M^{(n)}$ a *bipartite uncertain n-SuperHyperGraph* if there exists a partition

$$V \;=\; V_1 \,\dot{\cup}\, V_2$$

such that every active superedge meets both parts:

$$\forall\, \varepsilon \in E^+, \qquad \varepsilon \cap V_1 \neq \emptyset \;\text{ and }\; \varepsilon \cap V_2 \neq \emptyset.$$

Example 6.10.10 (A bipartite uncertain n-SuperHyperGraph in practice). **Cross-team software release governance under an uncertain model.** Consider a company that ships a product through coordinated releases involving *engineering groups* and *control groups* (security/compliance/ops). Some coordination events are well verified, others are only partially supported; this is modeled by an uncertain degree domain.

Uncertain model. Let M be the fuzzy uncertain model with $\mathrm{Dom}(M) = [0, 1]$ and distinguished zero degree $\mathbf{0} = 0$. (Any other uncertain model M with a distinguished $\mathbf{0}$ works similarly.)

Base set and supervertices (take $n = 2$). Let the base set V_0 be a set of individual engineers:

$$V_0 = \{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{E}, \mathsf{F}\}.$$

Define four 2-supervertices (elements of $\mathcal{P}^2(V_0)$) as *sets of teams*:

$$X_1 := \{\{\mathsf{A}, \mathsf{B}\}, \{\mathsf{C}\}\} \quad \text{(Backend program)}, \qquad X_2 := \{\{\mathsf{D}\}, \{\mathsf{E}\}\} \quad \text{(Data program)},$$

$$Y_1 := \{\{\mathsf{F}\}\} \quad \text{(Security review unit)}, \qquad Y_2 := \{\{\mathsf{B}, \mathsf{E}\}\} \quad \text{(Operations rollout unit)},$$

and let

$$V := \{X_1, X_2, Y_1, Y_2\} \subseteq \mathcal{P}^2(V_0).$$

Bipartition of supervertices. Let

$$V_1 := \{X_1, X_2\} \quad \text{(product programs)}, \qquad V_2 := \{Y_1, Y_2\} \quad \text{(control units)}, \qquad V = V_1 \dot{\cup} V_2.$$

Superedges (multi-party release checkpoints). Define two superedges:

$$\varepsilon_1 := \{X_1, Y_1\} \quad \text{(security gate for backend)},$$

$$\varepsilon_2 := \{X_2, Y_1, Y_2\} \quad \text{(data release with security \& ops)},$$

and set $E := \{\varepsilon_1, \varepsilon_2\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$.

Uncertain degrees and active superedges. Define $\mu_M : V \cup E \to [0, 1]$ by (example values)

$$\mu_M(X_1) = 0.9, \ \mu_M(X_2) = 0.8, \ \mu_M(Y_1) = 0.95, \ \mu_M(Y_2) = 0.7,$$

$$\mu_M(\varepsilon_1) = 0.6, \qquad \mu_M(\varepsilon_2) = 0.4.$$

Then $\mu_M(\varepsilon_i) \neq 0$, so both superedges are active:

$$E^+ = \{\varepsilon \in E : \mu_M(\varepsilon) \neq 0\} = E.$$

Bipartiteness check. Each active superedge meets both parts:

$$\varepsilon_1 \cap V_1 = \{X_1\} \neq \emptyset, \ \varepsilon_1 \cap V_2 = \{Y_1\} \neq \emptyset,$$

$$\varepsilon_2 \cap V_1 = \{X_2\} \neq \emptyset, \ \varepsilon_2 \cap V_2 = \{Y_1, Y_2\} \neq \emptyset.$$

Hence $\mathcal{S}_M^{(2)} = (V, E, \mu_M)$ is a *bipartite uncertain 2-SuperHyperGraph* in the sense of Definition 6.10.9.

Interpretation. Supervertices represent nested organizational entities (collections of teams), while superedges represent release checkpoints that must involve both product programs and control units. The uncertainty degree μ_M encodes strength or confidence of participation/evidence; active checkpoints ($\mu_M \neq \mathbf{0}$) are required to span both sides of the bipartition.

6.11 Multipartite SuperHypergraph

A multipartite graph partitions vertices into k disjoint parts, and every edge connects two vertices from different parts only [236–239]. Multipartite graphs model multi-group interactions, forbid within-group edges, simplify constraints, and enable efficient colorings, matchings, and partition-based optimizations. As a further extension of multipartite graphs, we discuss iterated multipartite graphs in the Appendix.

A multipartite hypergraph partitions vertices into k disjoint parts, and each hyperedge contains at most one vertex from each part [240–243]. A multipartite superhypergraph partitions n-supervertices into k disjoint classes, and each superhyperedge selects one supervertex from each class. The relevant definitions and related notions are presented below.

Definition 6.11.1 (k-partite graph; multipartite graph)**.** [244–246] Let $k \geq 2$. A (simple) graph is a pair $G = (V, E)$ where V is a set of vertices and $E \subseteq \binom{V}{2}$ is a set of (2-element) edges. We say that G is *k-partite* if there exist pairwise disjoint sets $V_1, \ldots, V_k$ such that

$$V \;=\; V_1 \,\dot{\cup}\, \cdots \,\dot{\cup}\, V_k$$

and for every edge $\{u, v\} \in E$ there exist indices $i \neq j$ with $u \in V_i$ and $v \in V_j$. If G is k-partite for some $k \geq 2$, then G is called *multipartite*.

Definition 6.11.2 (k-uniform multipartite hypergraph)**.** Let $k \geq 2$. A *k-uniform multipartite hypergraph* is a hypergraph $H = (V, \mathcal{E})$ for which there exist pairwise disjoint vertex classes $V_1, \ldots, V_k$ such that

$$V \;=\; V_1 \,\dot{\cup}\, \cdots \,\dot{\cup}\, V_k,$$

and the hyperedge family $\mathcal{E}$ can be represented as a set of k-tuples

$$\mathcal{E} \subseteq V_1 \times \cdots \times V_k,$$

where each hyperedge $e = (v_1, \ldots, v_k) \in \mathcal{E}$ corresponds to the k-element subset $\{v_1, \ldots, v_k\} \subseteq V$, equivalently satisfying $|\{v_1, \ldots, v_k\} \cap V_i| = 1$ for all $i = 1, \ldots, k$.

Definition 6.11.3 (k-uniform multipartite n-superhypergraph)**.** Let $n \geq 1$ and $k \geq 2$. Let $V_{0,1}, \ldots, V_{0,k}$ be pairwise disjoint nonempty base sets, and define the k vertex classes by

$$V_i \;:=\; \mathcal{P}^{\,n-1}(V_{0,i}) \qquad (i = 1, \ldots, k).$$

A *k-uniform multipartite n-superhypergraph* is a hypergraph $H = (V, \mathcal{E})$ such that

$$V \;=\; V_1 \,\dot{\cup}\, \cdots \,\dot{\cup}\, V_k, \qquad \mathcal{E} \subseteq V_1 \times \cdots \times V_k,$$

and each hyperedge $e = (v_1, \ldots, v_k) \in \mathcal{E}$ selects exactly one (super)vertex $v_i \in V_i$ from each class.

In particular, when $n = 1$ we have $V_i = \mathcal{P}^0(V_{0,i}) = V_{0,i}$, so the above definition reduces to the usual k-uniform multipartite hypergraph on the base vertex classes.

Example 6.11.4 (3-uniform multipartite 2-superhypergraph)**.** Let $n = 2$ and $k = 3$. Take pairwise disjoint nonempty base sets

$$V_{0,1} := \{a, b\}, \qquad V_{0,2} := \{c\}, \qquad V_{0,3} := \{d, e\}.$$

Since $n - 1 = 1$, the three vertex classes are the powersets

$$V_1 = \mathcal{P}(V_{0,1}), \qquad V_2 = \mathcal{P}(V_{0,2}), \qquad V_3 = \mathcal{P}(V_{0,3}).$$

Choose the (super)vertex set $V \subseteq V_1 \dot\cup V_2 \dot\cup V_3$ by specifying a few elements from each class:

$$\begin{aligned} V^{(1)} &:= \big\{\{a\}, \{b\}, \{a, b\}\big\} \subseteq V_1, \\ V^{(2)} &:= \big\{\{c\}\big\} \subseteq V_2, \qquad\qquad V := V^{(1)} \dot\cup V^{(2)} \dot\cup V^{(3)}. \\ V^{(3)} &:= \big\{\{d\}, \{e\}, \{d, e\}\big\} \subseteq V_3, \end{aligned}$$

Define a hyperedge family $\mathcal{E} \subseteq V_1 \times V_2 \times V_3$ by

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

where each edge selects exactly one element from each class:

$$\begin{aligned} e_1 &:= \big(\{a\}, \{c\}, \{d\}\big), \\ e_2 &:= \big(\{b\}, \{c\}, \{d, e\}\big), \\ e_3 &:= \big(\{a, b\}, \{c\}, \{e\}\big). \end{aligned}$$

Then $H := (V, \mathcal{E})$ is a 3-uniform multipartite 2-superhypergraph: the vertex set is partitioned into three classes $V^{(1)}, V^{(2)}, V^{(3)}$, and every hyperedge $e_j \in \mathcal{E}$ is a triple (v_1, v_2, v_3) with $v_i \in V_i$, i.e., it chooses exactly one (super)vertex from each class.

Definition 6.11.5 ((Recall) Nested singleton lift within each part)**.** Let W be a nonempty set and let $m \geq 0$. Define $\iota_0 : W \to W$ by $\iota_0(w) := w$, and for $m \geq 1$ define recursively

$$\iota_m(w) \;:=\; \{\iota_{m-1}(w)\} \;\in\; \mathcal{P}^m(W).$$

Thus $\iota_m : W \hookrightarrow \mathcal{P}^m(W)$ is an injective map.

Definition 6.11.6 ((Recall) 2-section of a hypergraph)**.** Let $H = (V, \mathcal{E})$ be a hypergraph. Its *2-section* (or *shadow graph*) is the graph $\partial_2(H) = (V, E_2)$ where $\{u, v\} \in E_2$ if and only if $u \neq v$ and there exists $e \in \mathcal{E}$ with $\{u, v\} \subseteq e$.

Theorem 6.11.7 (Multipartite n-SuperHyperGraphs generalize multipartite hypergraphs and multipartite graphs)**.** *Fix integers $n \geq 1$ and $k \geq 2$.*

*(i) (*Multipartite hypergraphs are the case $n = 1$.*) Let $H = (V, \mathcal{E})$ be a k-uniform multipartite hypergraph with vertex partition $V = V_{0,1} \dot\cup \cdots \dot\cup V_{0,k}$ and $\mathcal{E} \subseteq V_{0,1} \times \cdots \times V_{0,k}$. Then H is exactly a k-uniform multipartite* 1*-superhypergraph in the sense of Definition* (*k-uniform multipartite n-superhypergraph*)*.*

More generally, for any $n \geq 1$ one obtains a k-uniform multipartite n-superhypergraph $H^{(n)} = (V^{(n)}, \mathcal{E}^{(n)})$ whose incidence structure is naturally identified with that of H, by the partwise lift

$$V^{(n)} := \bigcup_{i=1}^{k} \{\iota_{n-1}(v) : v \in V_{0,i}\} \subseteq \bigcup_{i=1}^{k} \mathcal{P}^{n-1}(V_{0,i}),$$

and

$$\mathcal{E}^{(n)} := \Big\{\big(\iota_{n-1}(v_1), \ldots, \iota_{n-1}(v_k)\big) : (v_1, \ldots, v_k) \in \mathcal{E}\Big\} \subseteq \prod_{i=1}^{k} \mathcal{P}^{n-1}(V_{0,i}).$$

*(ii) (*Multipartite graphs embed via a standard k-uniform multipartite expansion.*) Let $G = (V, E)$ be a finite k-partite graph with vertex partition*

$$V \;=\; V_{0,1}\dot{\cup}\cdots\dot{\cup}V_{0,k}, \qquad E \subseteq \bigcup_{1\le i<j\le k} (V_{0,i} \times V_{0,j}).$$

Choose new "filler" vertices $f_i \notin V$ (one for each part) and set

$$V'_{0,i} := V_{0,i} \cup \{f_i\} \qquad (i = 1, \ldots, k).$$

Define a k-uniform multipartite hypergraph $H_G = (V', \mathcal{E}_G)$ by

$$V' \;:=\; V'_{0,1}\dot{\cup}\cdots\dot{\cup}V'_{0,k},$$

and for each edge $\{u, v\} \in E$ with $u \in V_{0,i}$, $v \in V_{0,j}$ $(i \neq j)$, define the k-tuple (hyperedge)

$$e_{uv} \;:=\; (x_1, \ldots, x_k) \in V'_{0,1} \times \cdots \times V'_{0,k},$$

where $x_i = u$, $x_j = v$, and $x_\ell = f_\ell$ for all $\ell \notin \{i, j\}$. Let $\mathcal{E}_G := \{e_{uv} : \{u, v\} \in E\}$. Then H_G is k-uniform multipartite, and the original graph G is recovered as the induced subgraph of the 2-section $\partial_2(H_G)$ on the original vertex set V:

$$G \;\cong\; \partial_2(H_G)[V].$$

Consequently, by applying the partwise lift in (i) to the base partition $(V'_{0,1}, \ldots, V'_{0,k})$, every k-partite graph arises canonically from a k-uniform multipartite n-superhypergraph via a shadow/2-section and restriction to non-filler vertices.

Proof. (i) If $n = 1$, then $\mathcal{P}^0(V_{0,i}) = V_{0,i}$ and the definition of a k-uniform multipartite 1-superhypergraph coincides verbatim with that of a k-uniform multipartite hypergraph: vertices are partitioned into k classes and each hyperedge selects exactly one vertex from each class. For general $n \geq 1$, the map ι_{n-1} in Definition 6.11.5 is injective on each part, so $V^{(n)}$ is naturally identified with V (part by part). Moreover, each hyperedge $(v_1, \ldots, v_k) \in \mathcal{E}$ corresponds uniquely to the lifted hyperedge $(\iota_{n-1}(v_1), \ldots, \iota_{n-1}(v_k)) \in \mathcal{E}^{(n)}$. Thus $H^{(n)}$ is a k-uniform multipartite n-superhypergraph that is incidence-isomorphic to H.

(ii) By construction, every $e_{uv} \in \mathcal{E}_G$ contains exactly one vertex from each part $V'_{0,i}$, so H_G is k-uniform multipartite. Consider the 2-section $\partial_2(H_G)$ (Definition 6.11.6) and restrict it to V. If $\{u, v\} \in E$, then u and v appear together in the hyperedge e_{uv}, so $\{u, v\} \in E\big(\partial_2(H_G)[V]\big)$. Conversely, if $\{u, v\} \subseteq V$ is an edge of $\partial_2(H_G)$, then there exists $e_{xy} \in \mathcal{E}_G$ such that $\{u, v\} \subseteq e_{xy}$. But in any e_{xy}, the only non-filler entries are precisely the two endpoints x and y of the original graph edge (all other coordinates are fillers f_ℓ). Hence $\{u, v\} = \{x, y\} \in E$. Therefore $\partial_2(H_G)[V]$ has exactly the same edge set as G, proving $G \cong \partial_2(H_G)[V]$. Applying the lift from (i) to H_G yields the claimed realization via an n-superhypergraph. □

6.12 Complete multipartite graph, complete multipartite hypergraph, and complete multipartite n-SuperHyperGraph

A complete multipartite graph partitions vertices into k independent parts and includes every possible edge between distinct parts [236, 247–250]. A complete multipartite k-uniform hypergraph partitions vertices into k parts and contains every transversal hyperedge selecting one vertex per part. A complete multipartite k-uniform n-SuperHyperGraph partitions n-supervertices into k classes and includes all transversal superedges across classes.

Definition 6.12.1 (k-partite and complete k-partite graphs)**.** Let $k \geq 2$. A (finite) simple graph is a pair $G = (V, E)$ with $E \subseteq \binom{V}{2}$. We say that G is *k-partite* if there exist pairwise disjoint (possibly empty) sets $V_1, \ldots, V_k$ such that

$$V = V_1 \dot\cup \cdots \dot\cup V_k$$

and every edge has endpoints in different parts:

$$\{u, v\} \in E \implies \exists\, i \neq j \text{ with } u \in V_i,\ v \in V_j.$$

Such a family $(V_1, \ldots, V_k)$ is called a *k-partition* of G.

Given a k-partition $(V_1, \ldots, V_k)$, we say that G is *complete k-partite* (with respect to this partition) if, additionally, every cross-part pair is an edge:

$$\forall\, i \neq j,\ \forall\, u \in V_i,\ \forall\, v \in V_j, \quad \{u, v\} \in E.$$

If $|V_i| = n_i$ for $i = 1, \ldots, k$, such a graph is denoted $K_{n_1,\ldots,n_k}$. A graph is *complete multipartite* if it is complete k-partite for some $k \geq 2$.

Remark 6.12.2. A complete k-partite graph is the complement of a disjoint union of k cliques (one clique on each part V_i).

Definition 6.12.3 (Complete k-partite k-uniform hypergraph)**.** Let $k \geq 2$ and let $V_1, \ldots, V_k$ be pairwise disjoint nonempty sets, and set $V := V_1 \dot\cup \cdots \dot\cup V_k$. Define the family of all *transversals* (one vertex from each part) by

$$\mathcal{E}_{\text{comp}} := \Big\{ \{v_1, \ldots, v_k\} \subseteq V :\ v_i \in V_i \text{ for all } i = 1, \ldots, k \Big\}.$$

The hypergraph

$$K^{(k)}_{V_1,\ldots,V_k} := (V, \mathcal{E}_{\text{comp}})$$

is called the *complete k-partite k-uniform hypergraph* on $(V_1, \ldots, V_k)$. A hypergraph is called *complete multipartite* if it is isomorphic to $K^{(k)}_{V_1,\ldots,V_k}$ for some $k \geq 2$ and some nonempty parts $V_1, \ldots, V_k$.

Definition 6.12.4 (Complete k-partite k-uniform n-SuperHyperGraph)**.** Let $n \geq 1$ and $k \geq 2$, and let $V_1, \ldots, V_k$ be pairwise disjoint nonempty sets of n-supervertices (i.e., $V_i \subseteq \mathcal{P}^n(V_0)$ for some base set V_0). Set $V := V_1 \dot\cup \cdots \dot\cup V_k$, and define

$$E_{\text{comp}} := \Big\{ \{X_1, \ldots, X_k\} \subseteq V :\ X_i \in V_i \text{ for all } i = 1, \ldots, k \Big\}.$$

Then

$$K^{(k)}_{V_1,\ldots,V_k} := (V, E_{\text{comp}})$$

is called the *complete k-partite k-uniform n-SuperHyperGraph* on $(V_1, \ldots, V_k)$. An n-SuperHyperGraph is called *complete multipartite* if it is isomorphic to $K^{(k)}_{V_1,\ldots,V_k}$ for some $k \geq 2$ and some nonempty supervertex classes $V_1, \ldots, V_k$.

Example 6.12.5 (A concrete k-uniform multipartite n-superhypergraph)**.** We give an explicit example with $k = 3$ and $n = 2$.

Base classes. Let the three pairwise disjoint nonempty base sets be

$$V_{0,1} = \{a, b\}, \qquad V_{0,2} = \{c, d\}, \qquad V_{0,3} = \{e, f\}.$$

Since $n = 2$, we have $n - 1 = 1$, hence the three vertex classes are the powersets

$$V_1 = \mathcal{P}^1(V_{0,1}) = \mathcal{P}(V_{0,1}) = \big\{\emptyset, \{a\}, \{b\}, \{a, b\}\big\},$$

$$V_2 = \mathcal{P}(V_{0,2}) = \big\{\emptyset, \{c\}, \{d\}, \{c, d\}\big\}, \qquad V_3 = \mathcal{P}(V_{0,3}) = \big\{\emptyset, \{e\}, \{f\}, \{e, f\}\big\}.$$

Let

$$V := V_1 \,\dot\cup\, V_2 \,\dot\cup\, V_3.$$

Hyperedges (3-uniform and 3-partite). Define a hyperedge set $\mathcal{E} \subseteq V_1 \times V_2 \times V_3$ by selecting triples that pick exactly one element from each class:

$$\mathcal{E} := \Big\{\big(\{a\}, \{c, d\}, \{e\}\big),\ \big(\{a, b\}, \{d\}, \{f\}\big),\ \big(\{b\}, \{c\}, \{e, f\}\big)\Big\}.$$

Then $H = (V, \mathcal{E})$ is a 3-*uniform multipartite* 2-*superhypergraph*: each hyperedge $e = (v_1, v_2, v_3) \in \mathcal{E}$ chooses exactly one vertex $v_i \in V_i$ from each part V_1, V_2, V_3.

Remark. The empty set is permitted as a vertex in each class $V_i = \mathcal{P}(V_{0,i})$; if desired, one may restrict to $\mathcal{P}(V_{0,i}) \setminus \{\emptyset\}$ without changing the multipartite k-uniform mechanism.

Chapter 7

Planarity

In this chapter, we extend the notions of planarity and planar graphs to the SuperHyperGraph setting and investigate their fundamental properties.

7.1 Planar SuperHypergraphs

A planar graph is a finite graph drawable on the plane without edge crossings, using nonintersecting straight or curved edges [251–253]. As concepts related to planar graphs, notions such as quasi-planar graphs [254–256], planar digraphs [257, 258], co-planar graphs [259, 260], fuzzy planar graphs [261, 262], biplanar graphs [263, 264], and neutrosophic planar graphs [265–268] are well known. Planar graphs are often not structurally complex, and because their edges do not cross, they offer high visual clarity for human interpretation. For this reason, the notion of planarity has been widely applied in many research papers.

A planar hypergraph is a finite hypergraph whose incidence graph admits a planar embedding without edge crossings in the plane [28, 269]. A planar SuperHyperGraph is a finite SuperHyperGraph whose incidence graph can be embedded in the plane without any edge crossings. The relevant definitions and related notions are presented below.

Definition 7.1.1 (Incidence graph of a hypergraph)**.** Let $H = (V, E)$ be a finite hypergraph with incidence map $\partial : E \to \mathcal{P}^*(V)$. The *incidence graph* of H is the bipartite graph

$$B(H) \;:=\; (V \cup E,\; F),$$

where

$$F \;:=\; \big\{\, \{v, e\} \subseteq V \cup E \mid e \in E,\; v \in \partial(e) \,\big\}.$$

Definition 7.1.2 (Planar hypergraph)**.** [28, 269] A hypergraph $H = (V, E)$ is called *planar* if its incidence graph $B(H)$ is a planar graph, i.e. it admits a drawing in the plane with no edge crossings.

Definition 7.1.3 (Incidence graph of an n-SuperHyperGraph)**.** Let $\mathsf{H}^{(n)} = (V, E, \partial)$ be an n-SuperHyperGraph, where $\partial : E \to \mathcal{P}^*(V)$ is the incidence map (as defined earlier in the text). The *incidence graph* of $\mathsf{H}^{(n)}$ is the bipartite graph

$$B\big(\mathsf{H}^{(n)}\big) \;:=\; \big(V \cup E,\; F^{(n)}\big),$$

where

$$F^{(n)} \;:=\; \big\{\, \{v, e\} \subseteq V \cup E \mid e \in E,\; v \in \partial(e) \,\big\}.$$

Definition 7.1.4 (Planar n-SuperHyperGraph)**.** An n-SuperHyperGraph $\mathsf{H}^{(n)} = (V, E, \partial)$ is called *planar* if its incidence graph $B\big(\mathsf{H}^{(n)}\big)$ is a planar graph.

Example 7.1.5 (A planar 2-SuperHyperGraph)**.** Let the base set be $V_0 = \{a, b, c\}$ and take $n = 2$. Define three 2-supervertices (each is an element of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$) by

$$X_a := \{\{a\}\}, \qquad X_b := \{\{b\}\}, \qquad X_c := \{\{c\}\}.$$

Set

$$V := \{X_a, X_b, X_c\} \subseteq \mathcal{P}^2(V_0),$$

and define two superedges by

$$\varepsilon_1 := \{X_a, X_b\}, \qquad \varepsilon_2 := \{X_b, X_c\}, \qquad E := \{\varepsilon_1, \varepsilon_2\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Then $\mathsf{H}^{(2)} := (V, E, \partial)$ (with the natural incidence map $\partial(\varepsilon) = \varepsilon$) is a finite 2-SuperHyperGraph.

Its incidence graph $B(\mathsf{H}^{(2)})$ has vertex set $V \cup E$ and edges X—ε whenever $X \in \partial(\varepsilon)$. Hence

$$X_a—\varepsilon_1—X_b—\varepsilon_2—X_c$$

is exactly $B(\mathsf{H}^{(2)})$, which is a path and therefore planar. Consequently, $\mathsf{H}^{(2)}$ is a planar 2-SuperHyperGraph.

7.2 Outerplanar SuperHypergraph

An outerplanar graph is a planar graph that admits an embedding in which every vertex lies on the boundary of the outer face [270, 271]. Outerplanar graphs generalize the class of planar graphs, and several related variants have been studied, including fuzzy outerplanar graphs [272–276], neutrosophic outerplanar graphs [267], and outerplanar directed graphs [277–279]. Outerplanar graphs form a tractable planar subclass with strong structural characterizations, enabling efficient algorithms, clear embeddings, and useful bounds for width parameters and graph drawing applications (cf. [280, 281]). The relevant definitions and related notions are presented below.

An *outerplanar hypergraph* is defined as a hypergraph whose incidence bipartite graph is outerplanar, so that all vertices can be placed on the outer face in some planar embedding [282, 283]. An *outerplanar superhypergraph* is a superhypergraph whose extended incidence graph—obtained by adding auxiliary links between hyperedges—admits an outerplanar embedding. In other words, the enriched incidence structure must remain outerplanar when drawn in the plane.

Definition 7.2.1 (Outerplanar graph)**.** [270, 271] A (finite, simple) graph $G = (V, E)$ is *outerplanar* if there exists a plane embedding of G in which every vertex of G lies on the boundary of the unbounded (outer) face.

Definition 7.2.2 (Incidence (bipartite) representation of a hypergraph)**.** Let $H = (V, \mathcal{E})$ be a (finite) hypergraph, where $\emptyset \notin \mathcal{E} \subseteq \mathcal{P}(V)$. Its *incidence graph* (or *bipartite representation*) is the bipartite graph

$$B(H) := \big(V \,\dot\cup\, \mathcal{E},\ F\big), \qquad F := \big\{\{v, e\} : v \in V,\ e \in \mathcal{E},\ v \in e\big\}.$$

Definition 7.2.3 (Outerplanar hypergraph)**.** [282, 283] A hypergraph H is *outerplanar* if its incidence graph $B(H)$ is an outerplanar graph.

Definition 7.2.4 (Shadow of a hypergraph)**.** Let $H = (V, \mathcal{E})$ be a hypergraph. Its *shadow* (or *2-section*) is the graph

$$\partial(H) := \big(V,\ \{\{u, v\} : u \neq v,\ \exists e \in \mathcal{E} \text{ with } \{u, v\} \subseteq e\}\big).$$

Definition 7.2.5 (Outerplanar 3-uniform hypergraph (Zykov-type))**.** Let $H = (V, \mathcal{E})$ be 3-uniform (i.e., $|e| = 3$ for all $e \in \mathcal{E}$). We say that H is *outerplanar* if $\partial(H)$ has an outerplanar embedding such that, for every hyperedge $e = \{a, b, c\} \in \mathcal{E}$, the vertices a, b, c bound an interior triangular face of that embedding.

Definition 7.2.6 (n-SuperHyperGraph via iterated super-links)**.** Fix an integer $n \geq 1$. An *n-superhypergraph* is a tuple

$$\mathcal{S} = \big(V, \mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n\big)$$

such that

$$\mathcal{E}_1 \subseteq \mathcal{P}^*(V) \quad \text{and} \quad \mathcal{E}_i \subseteq \binom{\mathcal{E}_{i-1}}{2} \quad \text{for every } 2 \leq i \leq n,$$

where $\mathcal{P}^*(X) := \mathcal{P}(X) \setminus \{\varnothing\}$. Elements of $\mathcal{E}_1$ are (hyper)edges, and elements of $\mathcal{E}_i$ ($i \geq 2$) are called *level-i super-links* (links between level-$(i-1)$ objects).

For $n = 1$ this is exactly a hypergraph $(V, \mathcal{E}_1)$. For $n = 2$ this is exactly a superhypergraph $(V, \mathcal{E}_1, \Lambda)$ with $\Lambda = \mathcal{E}_2 \subseteq \binom{\mathcal{E}_1}{2}$.

Definition 7.2.7 (Extended bipartite representation of an n-superhypergraph)**.** Let $\mathcal{S} = (V, \mathcal{E}_1, \ldots, \mathcal{E}_n)$ be an n-superhypergraph. Its *extended bipartite representation* is the (simple) graph

$$B^{(n)}_{\text{ext}}(\mathcal{S}) := \big(V_{\text{ext}},\ F_{\text{ext}}\big),$$

where the vertex set is the tagged (disjoint) union

$$V_{\text{ext}} := V \,\dot\cup\, \mathcal{E}_1 \,\dot\cup\, \mathcal{E}_2 \,\dot\cup\, \cdots \,\dot\cup\, \mathcal{E}_n,$$

and the edge set is the union

$$F_{\text{ext}} := F_1 \cup F_2 \cup \cdots \cup F_n,$$

with

$$F_1 := \big\{\{v, e\} \ :\ v \in V,\ e \in \mathcal{E}_1,\ v \in e\big\},$$

and for every $2 \leq i \leq n$,

$$F_i := \big\{\{x, \lambda\} \ :\ \lambda \in \mathcal{E}_i,\ x \in \mathcal{E}_{i-1},\ x \in \lambda\big\}.$$

Definition 7.2.8 (Outerplanar n-SuperHyperGraph)**.** An n-superhypergraph $\mathcal{S}$ is *outerplanar* if

$$B_{\mathrm{ext}}^{(n)}(\mathcal{S})$$

is an outerplanar graph.

Example 7.2.9 (An outerplanar 2-SuperHyperGraph)**.** We construct a concrete 2-superhypergraph $\mathcal{S} = (V, \mathcal{E}_1, \mathcal{E}_2)$ whose extended bipartite representation $B_{\mathrm{ext}}^{(2)}(\mathcal{S})$ is outerplanar.

Level 1 (a hypergraph). Let

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

Define three level-1 hyperedges by

$$e_1 = \{a, b\}, \qquad e_2 = \{b, c\}, \qquad e_3 = \{c, d\}, \qquad \mathcal{E}_1 = \{e_1, e_2, e_3\} \subseteq \mathcal{P}^*(V).$$

Level 2 super-links between hyperedges. Define two level-2 super-links (each links two level-1 edges) by

$$\lambda_1 = \{e_1, e_2\}, \qquad \lambda_2 = \{e_2, e_3\}, \qquad \mathcal{E}_2 = \{\lambda_1, \lambda_2\} \subseteq \binom{\mathcal{E}_1}{2}.$$

Thus $\mathcal{S} = (V, \mathcal{E}_1, \mathcal{E}_2)$ is a 2-superhypergraph in the sense of Definition (n-SuperHyperGraph via iterated super-links).

Outerplanarity check via the extended representation. The extended bipartite representation $B_{\mathrm{ext}}^{(2)}(\mathcal{S})$ has vertex set

$$V_{\mathrm{ext}} \;=\; V \,\dot{\cup}\, \mathcal{E}_1 \,\dot{\cup}\, \mathcal{E}_2 = \{a, b, c, d\} \,\dot{\cup}\, \{e_1, e_2, e_3\} \,\dot{\cup}\, \{\lambda_1, \lambda_2\},$$

and edges of two types:

$$F_1 = \{\{v, e\} : v \in V,\ e \in \mathcal{E}_1,\ v \in e\}, \qquad F_2 = \{\{e, \lambda\} : e \in \mathcal{E}_1,\ \lambda \in \mathcal{E}_2,\ e \in \lambda\}.$$

Hence $B_{\mathrm{ext}}^{(2)}(\mathcal{S})$ is the path

$$a \text{---} e_1 \text{---} b \text{---} e_2 \text{---} c \text{---} e_3 \text{---} d$$

together with the two additional attachments

$$e_1 \text{---} \lambda_1 \text{---} e_2, \qquad e_2 \text{---} \lambda_2 \text{---} e_3.$$

This graph contains no K_4 or $K_{2,3}$ minor and admits an embedding with all vertices on the outer face; for instance, it is a subgraph of a cycle with chords and thus outerplanar. Therefore $B_{\mathrm{ext}}^{(2)}(\mathcal{S})$ is outerplanar, and $\mathcal{S}$ is an outerplanar 2-SuperHyperGraph.

A concrete outerplanar drawing. Figure 7.1 depicts an outerplanar embedding of $B_{\mathrm{ext}}^{(2)}(\mathcal{S})$, with all vertices placed on a circle (the boundary of the outer face).

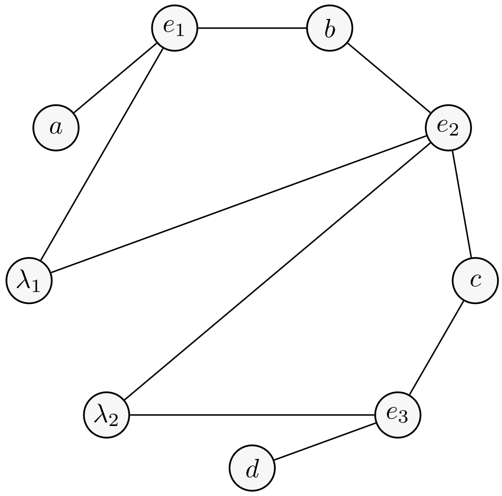


Figure 7.1.: An outerplanar embedding of the extended bipartite representation $B^{(2)}_{\mathrm{ext}}(\mathcal{S})$ for Example 7.2.9. All vertices lie on the boundary of the outer face.

7.3 Bipartite planar SuperHyperGraph

A bipartite planar graph is a bipartite graph that can be drawn in the plane without edge crossings, preserving adjacency relationships [284–287].

Definition 7.3.1 (Bipartite planar graph)**.** A (finite simple) graph is a pair $G = (V, E)$ where $V \neq \emptyset$ is a finite set and $E \subseteq \binom{V}{2}$. We call G a *bipartite planar graph* if there exists a partition

$$V = V_1 \dot{\cup} V_2$$

such that every edge has endpoints in different parts,

$$\forall \{u, v\} \in E, \quad \big(u \in V_1,\ v \in V_2\big) \text{ or } \big(u \in V_2,\ v \in V_1\big),$$

and G is *planar*, i.e., G admits a drawing in the plane with no edge crossings.

Definition 7.3.2 (Incidence graph of a hypergraph)**.** Let $H = (V, \mathcal{E})$ be a finite hypergraph, where $V \neq \emptyset$ and $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. The *incidence graph* (Levi graph) of H is the bipartite graph

$$B(H) := \big(V \dot{\cup} \mathcal{E},\ F\big), \qquad F := \big\{\{v, e\} :\ v \in V,\ e \in \mathcal{E},\ v \in e\big\}.$$

Definition 7.3.3 (Bipartite planar hypergraph)**.** Let $H = (V, \mathcal{E})$ be a finite hypergraph.

We say that H is *bipartite* if there exists a partition

$$V = V_1 \dot{\cup} V_2$$

such that every hyperedge meets both parts:

$$\forall e \in \mathcal{E}, \qquad e \cap V_1 \neq \emptyset \text{ and } e \cap V_2 \neq \emptyset.$$

We say that H is *planar* if its incidence graph $B(H)$ (Definition 7.3.2) is a planar graph.

We call H a *bipartite planar hypergraph* if it is both bipartite and planar.

Definition 7.3.4 (Incidence graph of an n-SuperHyperGraph)**.** Let V_0 be a finite nonempty base set and let $n \geq 0$. An *n-SuperHyperGraph* is a pair $\mathrm{SHG}^{(n)} = (V, E)$ such that

$$V \subseteq \mathcal{P}^n(V_0), \qquad E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Its *incidence graph* is the bipartite graph

$$B\big(\mathrm{SHG}^{(n)}\big) := \big(V \,\dot{\cup}\, E,\, F^{(n)}\big), \qquad F^{(n)} := \big\{\{X, \varepsilon\} : \; X \in V,\; \varepsilon \in E,\; X \in \varepsilon\big\}.$$

Definition 7.3.5 (Bipartite planar n-SuperHyperGraph)**.** Let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph.

We say that $\mathrm{SHG}^{(n)}$ is *bipartite* if there exists a partition

$$V = V_1 \,\dot{\cup}\, V_2$$

such that every superedge meets both parts:

$$\forall \varepsilon \in E, \qquad \varepsilon \cap V_1 \neq \emptyset \;\text{ and }\; \varepsilon \cap V_2 \neq \emptyset.$$

We say that $\mathrm{SHG}^{(n)}$ is *planar* if its incidence graph $B(\mathrm{SHG}^{(n)})$ (Definition 7.3.4) is planar.

We call $\mathrm{SHG}^{(n)}$ a *bipartite planar n-SuperHyperGraph* if it is both bipartite and planar.

Remark 7.3.6. The incidence graphs $B(H)$ and $B(\mathrm{SHG}^{(n)})$ are bipartite by construction (vertex-nodes vs. edge-nodes). In Definitions 7.3.3 and 7.3.5, the term "bipartite" instead refers to a 2-coloring/2-partition of the *vertex side* (respectively, the *supervertex side*) such that every (super)edge intersects both color classes.

Example 7.3.7 (A bipartite planar 2-SuperHyperGraph)**.** Let $V_0 = \{a, b, c\}$ and take $n = 2$. Define four 2-supervertices (elements of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$) by

$$X_1 = \{\{a\}\}, \qquad X_2 = \{\{b\}\}, \qquad X_3 = \{\{c\}\}, \qquad X_4 = \{\{a, b\}\}.$$

Set $V = \{X_1, X_2, X_3, X_4\} \subseteq \mathcal{P}^2(V_0)$ and define

$$\varepsilon_1 = \{X_1, X_3\}, \qquad \varepsilon_2 = \{X_2, X_4\}, \qquad \varepsilon_3 = \{X_1, X_4\}, \qquad E = \{\varepsilon_1, \varepsilon_2, \varepsilon_3\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Then $\mathrm{SHG}^{(2)} = (V, E)$ is a finite 2-SuperHyperGraph.

Bipartiteness. Let

$$V_1 := \{X_1, X_2\}, \qquad V_2 := \{X_3, X_4\}, \qquad \text{so } V = V_1 \dot{\cup} V_2.$$

Each superedge meets both parts:

$$\varepsilon_1 \cap V_1 = \{X_1\},\; \varepsilon_1 \cap V_2 = \{X_3\}; \qquad \varepsilon_2 \cap V_1 = \{X_2\},\; \varepsilon_2 \cap V_2 = \{X_4\}; \qquad \varepsilon_3 \cap V_1 = \{X_1\},\; \varepsilon_3 \cap V_2 = \{X_4\}.$$

Hence $\mathrm{SHG}^{(2)}$ is bipartite.

Planarity. The incidence graph $B(\mathrm{SHG}^{(2)})$ has vertex set $V \cup E$ and edges X—ε whenever $X \in \varepsilon$. Thus $B(\mathrm{SHG}^{(2)})$ has edge set

$$\{X_1\varepsilon_1,\; X_3\varepsilon_1,\; X_2\varepsilon_2,\; X_4\varepsilon_2,\; X_1\varepsilon_3,\; X_4\varepsilon_3\},$$

so it is a simple graph on seven vertices with no crossings needed; in particular it is planar. Therefore $\mathrm{SHG}^{(2)}$ is a bipartite planar 2-SuperHyperGraph in the sense of Definition 7.3.5.

7.4 Bipartite outerplanar SuperHypergraph

A bipartite outerplanar graph is an outerplanar graph whose vertex set can be partitioned into two independent sets, so every edge runs between the two parts and all vertices lie on the outer face [288–290]. A bipartite outerplanar superhypergraph is a bipartite superhypergraph (every active superedge meets both parts) whose incidence graph admits an outerplanar embedding, meaning all incidence vertices lie on the boundary of the outer face.

Definition 7.4.1 (Bipartite outerplanar graph)**.** A finite simple graph is a pair $G = (V, E)$ with $E \subseteq \binom{V}{2}$. We call G *bipartite outerplanar* if:

(i) (*Bipartite*) There exists a partition $V = V_1 \dot\cup V_2$ such that every edge has endpoints in different parts:
$$\forall \{u, v\} \in E,\ \exists\, i \neq j \text{ with } u \in V_i,\ v \in V_j.$$

(ii) (*Outerplanar*) G admits a planar embedding in which every vertex lies on the boundary of the outer face.

Equivalently, G is bipartite and outerplanar (as a crisp graph).

Definition 7.4.2 (Bipartite outerplanar hypergraph)**.** Let $H = (V, \mathcal{E})$ be a finite hypergraph with $\emptyset \notin \mathcal{E}$. We call H *bipartite outerplanar* if:

(i) (*Hypergraph bipartite / Property B*) There exists a partition $V = V_1 \dot\cup V_2$ such that every hyperedge meets both parts:
$$\forall\, e \in \mathcal{E}, \qquad e \cap V_1 \neq \emptyset \ \text{ and } \ e \cap V_2 \neq \emptyset.$$

(ii) (*Outerplanar incidence*) The incidence graph $B(H)$ (Definition **??**) is an outerplanar graph.

Definition 7.4.3 (Bipartite outerplanar n-SuperHyperGraph)**.** Let $\mathrm{SHG}^{(n)} = (V, E)$ be a finite n-SuperHyperGraph. We call $\mathrm{SHG}^{(n)}$ *bipartite outerplanar* if:

(i) (*Supervertex bipartite*) There exists a partition $V = V_1 \dot\cup V_2$ such that every superedge meets both parts:
$$\forall\, \varepsilon \in E, \qquad \varepsilon \cap V_1 \neq \emptyset \ \text{ and } \ \varepsilon \cap V_2 \neq \emptyset.$$

(ii) (*Outerplanar incidence*) The incidence graph $B(\mathrm{SHG}^{(n)})$ (Definition **??**) is an outerplanar graph.

Remark 7.4.4. The incidence graph of any hypergraph/n-SuperHyperGraph is *always bipartite* as a graph. In Definitions 7.4.2 and 7.4.3, "bipartite" additionally refers to a *2-colorability constraint on the (super)vertex side*: each (super)edge must intersect both color classes.

Example 7.4.5 (A bipartite outerplanar 2-SuperHyperGraph)**.** Let $V_0 = \{a, b, c, d\}$ and take $n = 2$. Define four 2-supervertices (elements of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$) by

$$X_1 := \{\{a\}\}, \qquad X_2 := \{\{b\}\}, \qquad X_3 := \{\{c\}\}, \qquad X_4 := \{\{d\}\},$$

and set

$$V := \{X_1, X_2, X_3, X_4\} \subseteq \mathcal{P}^2(V_0).$$

Define three superedges by

$$\varepsilon_1 := \{X_1, X_3\}, \qquad \varepsilon_2 := \{X_2, X_3\}, \qquad \varepsilon_3 := \{X_2, X_4\},$$

$$E := \{\varepsilon_1, \varepsilon_2, \varepsilon_3\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Then $\mathrm{SHG}^{(2)} = (V, E)$ is a finite 2-SuperHyperGraph.

(i) Supervertex bipartite. Let

$$V_1 := \{X_1, X_2\}, \qquad V_2 := \{X_3, X_4\}, \qquad \text{so that } V = V_1 \dot\cup V_2.$$

Each superedge meets both parts:

$$\varepsilon_1 \cap V_1 = \{X_1\},\ \varepsilon_1 \cap V_2 = \{X_3\}; \qquad \varepsilon_2 \cap V_1 = \{X_2\},\ \varepsilon_2 \cap V_2 = \{X_3\}; \qquad \varepsilon_3 \cap V_1 = \{X_2\},\ \varepsilon_3 \cap V_2 = \{X_4\}.$$

Hence $\mathrm{SHG}^{(2)}$ is bipartite in the sense of Definition 7.4.3(i).

(ii) Outerplanar incidence graph. The incidence graph $B(\mathrm{SHG}^{(2)})$ has vertex set $V \cup E$ and edges X—ε whenever $X \in \varepsilon$. Thus the adjacencies are

$$X_1 — \varepsilon_1 — X_3, \qquad X_2 — \varepsilon_2 — X_3, \qquad X_2 — \varepsilon_3 — X_4.$$

So $B(\mathrm{SHG}^{(2)})$ is a tree (indeed, a "Y"-shaped tree) with 7 vertices. Every tree is outerplanar, hence $B(\mathrm{SHG}^{(2)})$ is outerplanar. Therefore $\mathrm{SHG}^{(2)}$ is a bipartite outerplanar 2-SuperHyperGraph in the sense of Definition 7.4.3.

7.5 Apex SuperHyperGraph

An apex graph becomes planar after deleting a single vertex; equivalently, it has a vertex whose removal eliminates all crossings [291–293]. An apex superhypergraph becomes planar (in the incidence-graph sense) after deleting one supervertex; that vertex acts as an apex.

Definition 7.5.1 (Apex graph)**.** A finite simple graph G is called an *apex graph* if there exists a vertex $w \in V(G)$ such that the vertex-deleted graph $G - w$ is planar. Any such vertex w is called an *apex* of G.

Definition 7.5.2 (Planar hypergraph)**.** A hypergraph H is called *planar* if its incidence graph $B(H)$ is planar.

Definition 7.5.3 (Vertex deletion in a hypergraph)**.** Let $H = (V, \mathcal{E})$ be a hypergraph and let $v \in V$. Define the vertex-deleted hypergraph

$$H \ominus v \;:=\; \big(V \setminus \{v\},\ \mathcal{E} \ominus v\big), \qquad \mathcal{E} \ominus v \;:=\; \big\{\, e \setminus \{v\} :\ e \in \mathcal{E},\ e \setminus \{v\} \neq \emptyset \,\big\}.$$

Definition 7.5.4 (Apex hypergraph)**.** A finite hypergraph $H = (V, \mathcal{E})$ is called an *apex hypergraph* (with respect to Definition 7.5.2) if there exists a vertex $v \in V$ such that the hypergraph $H \ominus v$ is planar. Any such v is called an *apex vertex* of H.

Definition 7.5.5 (Planar n-SuperHyperGraph)**.** An n-SuperHyperGraph $\mathrm{SHG}^{(n)}$ is called *planar* if its incidence graph $B\big(\mathrm{SHG}^{(n)}\big)$ is planar.

Definition 7.5.6 (Supervertex deletion in an n-SuperHyperGraph)**.** Let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph and let $X \in V$. Define

$$\mathrm{SHG}^{(n)} \ominus X \;:=\; \big(V \setminus \{X\},\ E \ominus X\big), \qquad E \ominus X \;:=\; \big\{\, \varepsilon \setminus \{X\} :\ \varepsilon \in E,\ \varepsilon \setminus \{X\} \neq \emptyset \,\big\}.$$

Definition 7.5.7 (Apex n-SuperHyperGraph)**.** An n-SuperHyperGraph $\mathrm{SHG}^{(n)} = (V, E)$ is called an *apex n-SuperHyperGraph* (with respect to Definition 7.5.5) if there exists an n-supervertex $X \in V$ such that $\mathrm{SHG}^{(n)} \ominus X$ is planar. Any such X is called an *apex supervertex*.

Example 7.5.8 (An apex 2-SuperHyperGraph)**.** We construct a 2-SuperHyperGraph $\mathrm{SHG}^{(2)} = (V, E)$ that is not planar, but becomes planar after deleting one designated supervertex (an apex supervertex).

Step 1: A planar core. Let $V_0 = \{a, b, c\}$ and take $n = 2$. Define three 2-supervertices (elements of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$) by

$$X_1 = \{\{a\}\}, \qquad X_2 = \{\{b\}\}, \qquad X_3 = \{\{c\}\}.$$

Set

$$V_{\mathrm{core}} := \{X_1, X_2, X_3\} \subseteq \mathcal{P}^2(V_0), \qquad E_{\mathrm{core}} := \{\varepsilon_1, \varepsilon_2\},$$

where

$$\varepsilon_1 := \{X_1, X_2\}, \qquad \varepsilon_2 := \{X_2, X_3\}.$$

Then $\mathrm{SHG}^{(2)}_{\mathrm{core}} = (V_{\mathrm{core}}, E_{\mathrm{core}})$ is planar, because its incidence graph is the path

$$X_1 — \varepsilon_1 — X_2 — \varepsilon_2 — X_3.$$

Step 2: Add one supervertex that forces nonplanarity. Introduce an additional 2-supervertex

$$X_* := \{\{a, b, c\}\} \in \mathcal{P}^2(V_0), \qquad V := V_{\mathrm{core}} \cup \{X_*\}.$$

Now add three superedges

$$\varepsilon_3 := \{X_*, X_1\}, \qquad \varepsilon_4 := \{X_*, X_2\}, \qquad \varepsilon_5 := \{X_*, X_3\},$$

and set

$$E := E_{\text{core}} \cup \{\varepsilon_3, \varepsilon_4, \varepsilon_5\}.$$

Then $\mathrm{SHG}^{(2)} = (V, E)$ is a 2-SuperHyperGraph.

Nonplanarity. Consider the incidence graph $B(\mathrm{SHG}^{(2)})$ with bipartition (V, E). The subgraph induced by the vertex set

$$\{X_*, X_1, X_2, X_3\} \cup \{\varepsilon_3, \varepsilon_4, \varepsilon_5\}$$

is isomorphic to $K_{1,3}$ on the incidence side, and together with the two core superedges $\varepsilon_1, \varepsilon_2$ it creates a "fan" over the path $X_1 - X_2 - X_3$. In general, this configuration can still be planar; to force a genuine obstruction, we add one more superedge

$$\varepsilon_6 := \{X_1, X_2, X_3\}.$$

Now $B(\mathrm{SHG}^{(2)})$ contains the subdivision of $K_{3,3}$ on the bipartition

$$\{X_1, X_2, X_3\} \quad \text{and} \quad \{\varepsilon_3, \varepsilon_4, \varepsilon_5\},$$

and hence $B(\mathrm{SHG}^{(2)})$ is nonplanar. Therefore $\mathrm{SHG}^{(2)}$ is not planar.

Apex property. Delete the supervertex X_*. Then all superedges incident to X_*, namely $\varepsilon_3, \varepsilon_4, \varepsilon_5$, disappear from $\mathrm{SHG}^{(2)} \ominus X_*$. The remaining supergraph has supervertices $\{X_1, X_2, X_3\}$ and superedges $\{\varepsilon_1, \varepsilon_2, \varepsilon_6\}$, whose incidence graph is a subdivision of a triangle and is planar. Hence $\mathrm{SHG}^{(2)} \ominus X_*$ is planar, so $\mathrm{SHG}^{(2)}$ is an apex 2-SuperHyperGraph with apex supervertex X_* in the sense of Definition 7.5.7.

7.6 k-Planar Graphs, Hypergraphs, and SuperHyperGraphs

A **k-planar graph** admits a simple planar drawing in which each edge is crossed at most k times [294–296]. A **k-planar hypergraph** is one whose incidence (Levi) graph is k-planar, so each incidence edge has at most k crossings. A **k-planar n-SuperHyperGraph** is one whose supervertex–superedge incidence graph is k-planar, bounding crossings per incidence edge.

Definition 7.6.1 (k-planar drawing of a graph)**.** Let $G = (V, E)$ be a finite simple graph and let $k \in \mathbb{N}_0$. A *(topological) drawing* of G in the plane represents each vertex by a distinct point and each edge $uv \in E$ by a Jordan arc connecting u to v, such that the arc does not pass through any other vertex.

The drawing is called *simple* if:

(i) no edge crosses itself;

(ii) any two edges intersect at most once (either at a common endpoint or at a proper crossing);

(iii) no three edges cross at a common interior point; and

(iv) no two adjacent edges cross.

For an edge $e \in E$, let $\mathrm{cr}(e)$ be the number of proper crossings on the drawing of e. A simple drawing is *k-planar* if

$$\mathrm{cr}(e) \leq k \qquad (\forall e \in E).$$

Definition 7.6.2 (k-planar graph)**.** A graph G is *k-planar* if it admits a simple k-planar drawing in the sense of Definition 7.6.1. In particular, 0-planar graphs are exactly planar graphs.

Definition 7.6.3 (Incidence graph of a hypergraph)**.** Let $H = (V, \mathcal{E})$ be a finite hypergraph with $\emptyset \notin \mathcal{E} \subseteq \mathcal{P}(V)$. Its *incidence graph* (Levi graph) is the bipartite graph

$$B(H) := \big(V \,\dot\cup\, \mathcal{E},\, F\big), \qquad F := \big\{\{v, e\} : \; v \in V, \; e \in \mathcal{E}, \; v \in e\big\}.$$

Definition 7.6.4 (k-planar hypergraph)**.** Let $H = (V, \mathcal{E})$ be a finite hypergraph and let $k \in \mathbb{N}_0$. We call H *k-planar* if its incidence graph $B(H)$ (Definition 7.6.3) is a k-planar graph (Definition 7.6.2). Equivalently, H is k-planar if $B(H)$ admits a simple drawing in which every incidence edge is crossed at most k times.

Definition 7.6.5 (Incidence graph of an n-SuperHyperGraph)**.** Let V_0 be a finite nonempty base set, let $n \in \mathbb{N}_0$, and let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0, i.e.,

$$V \subseteq \mathcal{P}^n(V_0), \qquad E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

The *incidence graph* of $\mathrm{SHG}^{(n)}$ is the bipartite graph

$$B\big(\mathrm{SHG}^{(n)}\big) := \big(V \,\dot\cup\, E,\, F^{(n)}\big), \qquad F^{(n)} := \big\{\{X, \varepsilon\} : \; X \in V, \; \varepsilon \in E, \; X \in \varepsilon\big\}.$$

Definition 7.6.6 (k-planar n-SuperHyperGraph)**.** Let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph and let $k \in \mathbb{N}_0$. We call $\mathrm{SHG}^{(n)}$ *k-planar* if its incidence graph $B\big(\mathrm{SHG}^{(n)}\big)$ (Definition 7.6.5) is a k-planar graph (Definition 7.6.2). In particular, $k = 0$ yields the usual (incidence-graph) notion of planarity for superhypergraphs.

Example 7.6.7 (A 1-planar 2-SuperHyperGraph)**.** We construct a 2-SuperHyperGraph $\mathrm{SHG}^{(2)} = (V, E)$ whose incidence graph admits a drawing in which every edge is crossed at most once; hence $\mathrm{SHG}^{(2)}$ is 1-planar.

Supervertices. Let $V_0 = \{a, b, c, d\}$ and take $n = 2$. Define four 2-supervertices (elements of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$) by

$$X_1 := \{\{a\}\}, \qquad X_2 := \{\{b\}\}, \qquad X_3 := \{\{c\}\}, \qquad X_4 := \{\{d\}\},$$

and set

$$V := \{X_1, X_2, X_3, X_4\} \subseteq \mathcal{P}^2(V_0).$$

Superedges. Define four superedges by

$$\varepsilon_{13} := \{X_1, X_3\}, \qquad \varepsilon_{24} := \{X_2, X_4\}, \qquad \varepsilon_{12} := \{X_1, X_2\}, \qquad \varepsilon_{34} := \{X_3, X_4\},$$

and set

$$E := \{\varepsilon_{13}, \varepsilon_{24}, \varepsilon_{12}, \varepsilon_{34}\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Then $\mathrm{SHG}^{(2)} = (V, E)$ is a finite 2-SuperHyperGraph.

1-planarity of the incidence graph. The incidence graph $B(\mathrm{SHG}^{(2)})$ has vertex set $V \cup E$ and edges X—ε whenever $X \in \varepsilon$. Thus it consists of four length-2 paths realizing the incidences of each ε.

Consider the following plane drawing of $B(\mathrm{SHG}^{(2)})$: place the four supervertices at the corners of a square in the order X_1, X_2, X_3, X_4, and place the four superedge-vertices near the midpoints of the corresponding pairs. Draw the incidence edges X_i—ε_{ij} as straight segments. In this drawing, the only crossings occur between the pair of segments realizing ε_{13} and the pair realizing ε_{24}, and each incidence edge is crossed at most once. All other incidences ε_{12} and ε_{34} can be drawn without introducing additional crossings.

Hence $B(\mathrm{SHG}^{(2)})$ has a drawing in which every edge is crossed at most once, so it is a 1-planar graph. Therefore $\mathrm{SHG}^{(2)}$ is a 1-planar 2-SuperHyperGraph in the sense of Definition 7.6.6.

7.7 Thickness in SuperHyperGraph

A graph's thickness is the minimum number of planar spanning subgraphs whose edge-disjoint union equals the graph [297–299]. A SuperHyperGraph's thickness is the minimum number of planar superhypergraph layers whose superedge-disjoint union reconstructs the SuperHyperGraph.

Definition 7.7.1 (Thickness of a graph)**.** Let $G = (V, E)$ be a finite simple undirected graph. The *thickness* of G, denoted $\mathrm{th}(G)$, is the minimum integer $t \geq 1$ for which there exist pairwise edge-disjoint sets $E_1, \dots, E_t \subseteq E$ such that

$$E \;=\; E_1 \,\dot\cup\, \cdots \,\dot\cup\, E_t \quad \text{and} \quad G_i := (V, E_i) \text{ is planar for each } i = 1, \dots, t.$$

Equivalently, $\mathrm{th}(G)$ is the minimum number of planar spanning subgraphs whose union is G.

Definition 7.7.2 (Thickness of a hypergraph)**.** Let $H = (V, \mathcal{E})$ be a finite hypergraph. The *thickness* of H, denoted $\mathrm{th}(H)$, is the minimum integer $t \geq 1$ for which there exist pairwise disjoint subfamilies $\mathcal{E}_1, \dots, \mathcal{E}_t \subseteq \mathcal{E}$ such that

$$\mathcal{E} \;=\; \mathcal{E}_1 \,\dot\cup\, \cdots \,\dot\cup\, \mathcal{E}_t \quad \text{and} \quad H_i := (V, \mathcal{E}_i) \text{ is planar for each } i = 1, \dots, t,$$

where planarity is understood in the incidence-graph sense (Definition **??**).

Definition 7.7.3 (Thickness of an n-SuperHyperGraph)**.** Let $\mathrm{SHG}^{(n)} = (V, E)$ be a finite n-SuperHyperGraph. The *thickness* of $\mathrm{SHG}^{(n)}$, denoted $\mathrm{th}(\mathrm{SHG}^{(n)})$, is the minimum integer $t \geq 1$ for which there exist pairwise disjoint families $E_1, \ldots, E_t \subseteq E$ such that

$$E \;=\; E_1 \,\dot\cup\, \cdots \,\dot\cup\, E_t \quad \text{and} \quad \mathrm{SHG}_i^{(n)} := (V, E_i) \text{ is planar for each } i = 1, \ldots, t,$$

where planarity is defined via the incidence graph (Definition **??**).

Remark 7.7.4. In all three settings, thickness = 1 is equivalent to planarity (under the chosen planarity notion), and thickness = 2 corresponds to a decomposition into two planar layers (often called *biplanar* in the graph case).

Example 7.7.5 (Thickness of a 2-SuperHyperGraph)**.** We give a concrete 2-SuperHyperGraph whose thickness is 2.

Step 1: A nonplanar incidence graph (a $K_{3,3}$ Levi graph). Let $V_0 = \{a, b, c\}$ and take $n = 2$. Define three 2-supervertices (elements of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$) by

$$X_1 = \{\{a\}\}, \qquad X_2 = \{\{b\}\}, \qquad X_3 = \{\{c\}\},$$

and set $V = \{X_1, X_2, X_3\}$.

Define three superedges by

$$\varepsilon_1 = \{X_1, X_2, X_3\}, \qquad \varepsilon_2 = \{X_1, X_2, X_3\}, \qquad \varepsilon_3 = \{X_1, X_2, X_3\},$$

and let $E = \{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$. To avoid duplicate hyperedges, regard the three superedges as *distinct labels* even though they have the same underlying incidence set (equivalently, treat E as a multiset of superedges). Then the incidence graph $B(\mathrm{SHG}^{(2)})$ has bipartition (V, E) and every X_i is incident with every ε_j. Hence

$$B(\mathrm{SHG}^{(2)}) \;\cong\; K_{3,3},$$

which is nonplanar. Therefore $\mathrm{SHG}^{(2)} = (V, E)$ is not planar.

Step 2: Decomposition into two planar layers. Partition E into two disjoint families

$$E_1 := \{\varepsilon_1, \varepsilon_2\}, \qquad E_2 := \{\varepsilon_3\}, \qquad \text{so that } E = E_1 \dot\cup E_2.$$

Let $\mathrm{SHG}_1^{(2)} := (V, E_1)$ and $\mathrm{SHG}_2^{(2)} := (V, E_2)$. Their incidence graphs are

$$B(\mathrm{SHG}_1^{(2)}) \;\cong\; K_{3,2} \quad \text{and} \quad B(\mathrm{SHG}_2^{(2)}) \;\cong\; K_{3,1},$$

both of which are planar. Hence $\mathrm{th}(\mathrm{SHG}^{(2)}) \leq 2$.

Step 3: Minimality. Since $B(\mathrm{SHG}^{(2)}) \cong K_{3,3}$ is nonplanar, $\mathrm{SHG}^{(2)}$ cannot have thickness 1. Therefore $\mathrm{th}(\mathrm{SHG}^{(2)}) = 2$.

7.8 Fuzzy Planar Graph

A fuzzy planar graph is a fuzzy graph whose drawing is assigned a planarity value that penalizes edge crossings via normalized edge strengths [261,262,300]. Fuzzy planar graphs and their related notions have also been actively investigated in recent years from a wide range of perspectives [301–303]. A fuzzy planar hypergraph is a fuzzy hypergraph whose planarity value aggregates the intersection strengths of drawn hyperedge traces, and equals one when no such intersections occur. A fuzzy planar superhypergraph is a fuzzy n-superhypergraph whose planarity value measures crossings among drawn superedge traces, and equals one when they are absent.

Definition 7.8.1 (Fuzzy planar graph and fuzzy planarity value)**.** [261,262,300] A *fuzzy graph* is a triple $G = (V, \sigma, \mu)$, where $V \neq \emptyset$ is a finite vertex set, $\sigma : V \to [0, 1]$ is the vertex–membership map, and $\mu : V \times V \to [0, 1]$ is a symmetric edge–membership map with $\mu(v, v) = 0$ and

$$\mu(u, v) \ \leq \ \min\{\sigma(u), \sigma(v)\} \qquad (\forall\, u, v \in V).$$

Fix a geometric drawing of G in the plane, and let $P_1, \ldots, P_z$ be the (finite) set of intersection points between interiors of drawn edges (assume the drawing is in general position, so no three edges cross at one point). For an edge $e = uv$ with $\mu(u, v) > 0$, define its *normalized edge strength*

$$S(e) \ := \ \frac{\mu(u, v)}{\min\{\sigma(u), \sigma(v)\}} \ \in (0, 1].$$

If two edges e_1, e_2 intersect at a point P, define the *intersection strength* at P by

$$I_P \ := \ \frac{S(e_1) + S(e_2)}{2} \ \in (0, 1].$$

The *fuzzy planarity value* of the drawing is the scalar

$$f(G) \ := \ \frac{1}{1 + \sum_{i=1}^{z} I_{P_i}} \ \in (0, 1].$$

We call G a *fuzzy planar graph* (with respect to the chosen drawing) and refer to $f(G)$ as its *fuzzy planarity value.* In particular, $f(G) = 1$ if and only if there are no edge intersections in the drawing.

Definition 7.8.2 (Fuzzy planar hypergraph and fuzzy planarity value)**.** A *fuzzy hypergraph* is a quadruple

$$\mathcal{H} = (V, E, \sigma, \mu),$$

where $V \neq \emptyset$ is a finite set of vertices, E is a finite family of nonempty subsets of V (the *hyperedges*), $\sigma : V \to [0, 1]$ is the vertex–membership map, and $\mu : E \to [0, 1]$ is the hyperedge–membership map satisfying the *appurtenance constraint*

$$\mu(e) \ \leq \ \min_{v \in e} \sigma(v) \qquad (\forall\, e \in E).$$

Fix a geometric drawing of $\mathcal{H}$ in the plane as follows: each vertex $v \in V$ is drawn as a point, and each hyperedge $e \in E$ is drawn as a connected 1-dimensional trace γ_e (e.g., an embedded tree) whose endpoints (leaves) are exactly the vertices of e. Intersections are only counted when they occur between *interiors* of traces γ_e (so meeting at a common vertex is not an intersection).

Assume the drawing is in general position so that any intersection point involves exactly two traces. Let $P_1, \ldots, P_z$ be the (finite) set of such intersection points.

For a hyperedge $e \in E$ with $\mu(e) > 0$, define its *normalized hyperedge strength*

$$S(e) \;:=\; \frac{\mu(e)}{\min_{v\in e} \sigma(v)} \;\in (0,1].$$

If two distinct hyperedges e_1, e_2 have traces intersecting at a point P, define the *intersection strength* at P by

$$I_P \;:=\; \frac{S(e_1) + S(e_2)}{2} \;\in (0,1].$$

The *fuzzy planarity value* of the chosen drawing of $\mathcal{H}$ is

$$f(\mathcal{H}) \;:=\; \frac{1}{1 + \sum_{i=1}^{z} I_{P_i}} \;\in (0,1].$$

We call $\mathcal{H}$ a *fuzzy planar hypergraph* (with respect to the chosen drawing) and refer to $f(\mathcal{H})$ as its *fuzzy planarity value*. In particular, $f(\mathcal{H}) = 1$ if and only if the drawing has no intersections between interiors of hyperedge traces.

Definition 7.8.3 (Fuzzy planar n-SuperHyperGraph and fuzzy planarity value)**.** Fix an integer $n \geq 0$ and a finite base set V_0. For each $k \geq 0$, set

$$\mathcal{P}^0(V_0) = V_0, \qquad \mathcal{P}^{k+1}(V_0) = \mathcal{P}\big(\mathcal{P}^k(V_0)\big).$$

An *n-SuperHyperGraph* is a pair $\mathrm{SHG}^{(n)} = (V, E)$ with

$$V \subseteq \mathcal{P}^n(V_0), \qquad E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Elements of V are called *n-supervertices*, and elements of E are called *superedges*.

A *fuzzy n-SuperHyperGraph* is a quadruple

$$\mathcal{S} = (V, E, \sigma, \mu),$$

where $\mathrm{SHG}^{(n)} = (V, E)$ is an n-SuperHyperGraph, $\sigma : V \to [0,1]$ assigns a membership degree to each n-supervertex, and $\mu : E \to [0,1]$ assigns a membership degree to each superedge, satisfying

$$\mu(e) \;\leq\; \min_{x\in e} \sigma(x) \qquad (\forall\, e \in E).$$

Fix a geometric drawing of $\mathcal{S}$ in the plane in which each n-supervertex $x \in V$ is drawn as a point, and each superedge $e \in E$ is drawn as a connected 1-dimensional trace γ_e (e.g., an embedded tree) whose endpoints are exactly the n-supervertices in e. As before, intersections are counted only between interiors of traces of distinct superedges, and we assume a general-position drawing so that each intersection point involves exactly two superedges. Let $P_1, \ldots, P_z$ be the set of such intersection points.

For a superedge $e \in E$ with $\mu(e) > 0$, define its *normalized superedge strength*

$$S(e) \;:=\; \frac{\mu(e)}{\min_{x\in e} \sigma(x)} \;\in (0,1].$$

If two superedges e_1, e_2 intersect at a point P, define the *intersection strength* at P by

$$I_P := \frac{S(e_1) + S(e_2)}{2} \in (0, 1].$$

The *fuzzy planarity value* of the chosen drawing of $\mathcal{S}$ is

$$f(\mathcal{S}) := \frac{1}{1 + \sum_{i=1}^{z} I_{P_i}} \in (0, 1].$$

We call $\mathcal{S}$ a *fuzzy planar n-SuperHyperGraph* (with respect to the chosen drawing) and refer to $f(\mathcal{S})$ as its *fuzzy planarity value.* In particular, $f(\mathcal{S}) = 1$ if and only if there are no intersections between interiors of drawn superedges.

Example 7.8.4 (A fuzzy planar 2-SuperHyperGraph and its fuzzy planarity value)**.** Let $V_0 = \{a, b, c, d\}$ and take $n = 2$. Define four 2-supervertices (elements of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$) by

$$X_1 := \{\{a\}\}, \qquad X_2 := \{\{b\}\}, \qquad X_3 := \{\{c\}\}, \qquad X_4 := \{\{d\}\},$$

and set

$$V := \{X_1, X_2, X_3, X_4\} \subseteq \mathcal{P}^2(V_0).$$

Define two superedges by

$$e_1 := \{X_1, X_2, X_3\}, \qquad e_2 := \{X_2, X_3, X_4\}, \qquad E := \{e_1, e_2\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Thus $\mathrm{SHG}^{(2)} = (V, E)$ is a 2-SuperHyperGraph.

Fuzzy memberships and admissibility. Assign supervertex memberships $\sigma : V \to [0, 1]$ by

$$\sigma(X_1) = 0.9, \qquad \sigma(X_2) = 0.8, \qquad \sigma(X_3) = 0.7, \qquad \sigma(X_4) = 0.6,$$

and assign superedge memberships $\mu : E \to [0, 1]$ by

$$\mu(e_1) = 0.7, \qquad \mu(e_2) = 0.6.$$

Then the admissibility constraints of Definition 7.8.3 hold:

$$\mu(e_1) = 0.7 \le \min\{0.9, 0.8, 0.7\} = 0.7, \qquad \mu(e_2) = 0.6 \le \min\{0.8, 0.7, 0.6\} = 0.6.$$

A drawing with one interior intersection. Fix a plane drawing in which the supervertices X_1, X_2, X_3, X_4 are placed at the corners of a convex quadrilateral, and each superedge e_i is drawn as a connected trace γ_{e_i} that joins exactly its endpoints. Choose γ_{e_1} as a "Y-shaped" tree connecting X_1, X_2, X_3, and γ_{e_2} as a "Y-shaped" tree connecting X_2, X_3, X_4, arranged so that the interiors of γ_{e_1} and γ_{e_2} intersect exactly once at a point P_1 (and there are no other intersections). Thus $z = 1$.

Normalized strengths and intersection strength. Compute the normalized superedge strengths:

$$S(e_1) = \frac{\mu(e_1)}{\min_{x \in e_1} \sigma(x)} = \frac{0.7}{\min\{0.9, 0.8, 0.7\}} = \frac{0.7}{0.7} = 1,$$

$$S(e_2) = \frac{\mu(e_2)}{\min_{x \in e_2} \sigma(x)} = \frac{0.6}{\min\{0.8, 0.7, 0.6\}} = \frac{0.6}{0.6} = 1.$$

Hence the intersection strength at P_1 is

$$I_{P_1} = \frac{S(e_1) + S(e_2)}{2} = \frac{1+1}{2} = 1.$$

Fuzzy planarity value. Since there is exactly one intersection point,

$$f(\mathcal{S}) = \frac{1}{1 + \sum_{i=1}^{1} I_{P_i}} = \frac{1}{1 + I_{P_1}} = \frac{1}{1+1} = \frac{1}{2}.$$

Thus $f(\mathcal{S}) = 1/2 \in (0, 1)$, reflecting that the chosen drawing has one crossing between distinct superedges. If we instead select a crossing-free drawing ($z = 0$), then Definition 7.8.3 yields $f(\mathcal{S}) = 1$.

7.9 Intuitionistic Fuzzy Planar graph

An intuitionistic fuzzy planar graph assigns membership and nonmembership to edges; planarity holds when the active-edge support embeds crossing-free [267,304]. An intuitionistic fuzzy planar hypergraph assigns membership and nonmembership to hyperedges; planarity holds when the induced co-occurrence support graph is planar. An intuitionistic fuzzy planar superhypergraph assigns membership and nonmembership to superedges; planarity holds when the supervertex support graph admits a planar embedding.

Definition 7.9.1 (Intuitionistic fuzzy planar graph and planarity value)**.** [267,304] An *intuitionistic fuzzy multigraph* consists of a finite multigraph $G^* = (V, E^*)$ together with an intuitionistic fuzzy vertex assignment $A = (\mu_A, \nu_A)$ on V and an intuitionistic fuzzy multiedge assignment $B = (\mu_B, \nu_B)$ on E^*, such that

$$0 \le \mu_A(v) + \nu_A(v) \le 1 \quad (\forall v \in V), \qquad 0 \le \mu_B(e) + \nu_B(e) \le 1 \quad (\forall e \in E^*),$$

and (for each multiedge $e = uv$) the compatibility constraints

$$\mu_B(e) \;\le\; \min\{\mu_A(u), \mu_A(v)\}, \qquad \nu_B(e) \;\le\; \max\{\nu_A(u), \nu_A(v)\}.$$

For a multiedge $e = uv$, define its *intuitionistic edge strength* as the ordered pair

$$I_e \;:=\; (M_e, N_e) \;:=\; \left(\frac{\mu_B(e)}{\min\{\mu_A(u), \mu_A(v)\}}, \frac{\nu_B(e)}{\max\{\nu_A(u), \nu_A(v)\}} \right),$$

whenever $\mu_B(e) > 0$ and $\max\{\nu_A(u), \nu_A(v)\} > 0$; otherwise one may set $I_e = (0, 0)$ by convention.

Fix a geometric drawing of G^* in the plane, and let $P_1, \dots, P_z$ be the (finite) set of intersection points between interiors of drawn edges (assume a general-position drawing so no three edges cross at one point). If two edges e_1, e_2 intersect at a point P, define the *intersecting value* at P by

$$I_P \;:=\; (M_P, N_P) \;:=\; \left(\frac{M_{e_1} + M_{e_2}}{2}, \; \frac{N_{e_1} + N_{e_2}}{2} \right).$$

The *intuitionistic fuzzy planarity value* of the drawing is the pair

$$f(G) \;:=\; (f_M, f_N) \;:=\; \left(\frac{1}{1 + \sum_{i=1}^{z} M_{P_i}}, \frac{1}{1 + \sum_{i=1}^{z} N_{P_i}} \right) \;\in (0, 1]^2.$$

We call G an *intuitionistic fuzzy planar graph* (with respect to the chosen drawing), and $f(G)$ its *intuitionistic fuzzy planarity value.* If $z = 0$, then $f(G) = (1, 1)$.

Definition 7.9.2 (Intuitionistic fuzzy planar hypergraph and planarity value)**.** An *intuitionistic fuzzy hypergraph* is a quadruple

$$\mathcal{H} = (V, E, A, B),$$

where $V \neq \emptyset$ is a finite vertex set, E is a finite family of nonempty subsets of V (the *hyperedges*), and

$$A = (\mu_A, \nu_A) : V \to [0,1] \times [0,1], \qquad B = (\mu_B, \nu_B) : E \to [0,1] \times [0,1]$$

are intuitionistic fuzzy memberships satisfying, for all $v \in V$ and $e \in E$,

$$0 \le \mu_A(v) + \nu_A(v) \le 1, \qquad 0 \le \mu_B(e) + \nu_B(e) \le 1,$$

together with the hyperedge–vertex compatibility constraints

$$\mu_B(e) \ \le\ \min_{v \in e} \mu_A(v), \qquad \nu_B(e) \ \le\ \max_{v \in e} \nu_A(v).$$

Fix a geometric drawing of $\mathcal{H}$ in the plane in which each vertex is drawn as a point and each hyperedge $e \in E$ is drawn as a connected 1-dimensional trace γ_e (e.g., an embedded tree) whose endpoints (leaves) are exactly the vertices of e. Intersections are counted only between interiors of traces of *distinct* hyperedges, and we assume a general-position drawing so that each intersection point involves exactly two traces. Let $P_1, \dots, P_z$ be the set of such intersection points.

For a hyperedge $e \in E$, define its *intuitionistic hyperedge strength* as

$$I_e \ :=\ (M_e, N_e) \ :=\ \left(\frac{\mu_B(e)}{\min_{v \in e} \mu_A(v)}, \frac{\nu_B(e)}{\max_{v \in e} \nu_A(v)} \right),$$

whenever $\mu_B(e) > 0$ and $\max_{v \in e} \nu_A(v) > 0$; otherwise set $I_e = (0,0)$ by convention.

If two hyperedges e_1, e_2 have traces intersecting at a point P, define the *intersecting value* at P by

$$I_P \ :=\ (M_P, N_P) \ :=\ \left(\frac{M_{e_1} + M_{e_2}}{2}, \ \frac{N_{e_1} + N_{e_2}}{2} \right).$$

The *intuitionistic fuzzy planarity value* of the drawing of $\mathcal{H}$ is the pair

$$f(\mathcal{H}) \ :=\ (f_M, f_N) \ :=\ \left(\frac{1}{1 + \sum_{i=1}^{z} M_{P_i}}, \frac{1}{1 + \sum_{i=1}^{z} N_{P_i}} \right) \ \in (0,1]^2.$$

We call $\mathcal{H}$ an *intuitionistic fuzzy planar hypergraph* (with respect to the chosen drawing), and $f(\mathcal{H})$ its *intuitionistic fuzzy planarity value*. If $z = 0$, then $f(\mathcal{H}) = (1,1)$.

Definition 7.9.3 (Intuitionistic fuzzy planar n-SuperHyperGraph and planarity value)**.** Fix an integer $n \ge 0$ and a finite base set V_0. For each $k \ge 0$, define iterated powersets by

$$\mathcal{P}^0(V_0) = V_0, \qquad \mathcal{P}^{k+1}(V_0) = \mathcal{P}\big(\mathcal{P}^k(V_0)\big).$$

An *n-SuperHyperGraph* is a pair $\mathrm{SHG}^{(n)} = (V, E)$ with

$$V \subseteq \mathcal{P}^n(V_0), \qquad E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Elements of V are called *n-supervertices*, and elements of E are called *superedges*.

An *intuitionistic fuzzy n-SuperHyperGraph* is a quadruple

$$\mathcal{S} = (V, E, A, B),$$

where (V, E) is an n-SuperHyperGraph,

$$A = (\mu_A, \nu_A) : V \to [0,1] \times [0,1], \qquad B = (\mu_B, \nu_B) : E \to [0,1] \times [0,1],$$

satisfy

$$0 \le \mu_A(x) + \nu_A(x) \le 1 \quad (\forall x \in V), \qquad 0 \le \mu_B(e) + \nu_B(e) \le 1 \quad (\forall e \in E),$$

and the superedge–supervertex compatibility constraints

$$\mu_B(e) \ \le \ \min_{x \in e} \mu_A(x), \qquad \nu_B(e) \ \le \ \max_{x \in e} \nu_A(x) \qquad (\forall e \in E).$$

Fix a geometric drawing of $\mathcal{S}$ in the plane in which each n-supervertex is drawn as a point and each superedge $e \in E$ is drawn as a connected 1-dimensional trace γ_e whose endpoints are exactly the n-supervertices in e. Intersections are counted only between interiors of traces of distinct superedges, and we assume a general-position drawing so that each intersection point involves exactly two traces. Let $P_1, \dots, P_z$ be the set of such intersection points.

For a superedge $e \in E$, define its *intuitionistic superedge strength* as

$$I_e \ := \ (M_e, N_e) \ := \ \left(\frac{\mu_B(e)}{\min_{x \in e} \mu_A(x)}, \frac{\nu_B(e)}{\max_{x \in e} \nu_A(x)} \right),$$

whenever $\mu_B(e) > 0$ and $\max_{x \in e} \nu_A(x) > 0$; otherwise set $I_e = (0, 0)$ by convention.

If two superedges e_1, e_2 intersect at a point P, define the *intersecting value* at P by

$$I_P \ := \ (M_P, N_P) \ := \left(\frac{M_{e_1} + M_{e_2}}{2}, \ \frac{N_{e_1} + N_{e_2}}{2} \right).$$

The *intuitionistic fuzzy planarity value* of the drawing of $\mathcal{S}$ is the pair

$$f(\mathcal{S}) \ := \ (f_M, f_N) \ := \ \left(\frac{1}{1 + \sum_{i=1}^{z} M_{P_i}}, \frac{1}{1 + \sum_{i=1}^{z} N_{P_i}} \right) \ \in (0,1]^2.$$

We call $\mathcal{S}$ an *intuitionistic fuzzy planar n-SuperHyperGraph* (with respect to the chosen drawing), and $f(\mathcal{S})$ its *intuitionistic fuzzy planarity value.* If $z = 0$, then $f(\mathcal{S}) = (1, 1)$.

Example 7.9.4 (An intuitionistic fuzzy planar 2-SuperHyperGraph with one crossing)**.** Let $V_0 = \{a, b, c, d\}$ and take $n = 2$. Define four 2-supervertices (elements of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$) by

$$X_1 := \{\{a\}\}, \qquad X_2 := \{\{b\}\}, \qquad X_3 := \{\{c\}\}, \qquad X_4 := \{\{d\}\},$$

and set $V := \{X_1, X_2, X_3, X_4\} \subseteq \mathcal{P}^2(V_0)$. Define two superedges

$$e_1 := \{X_1, X_3\}, \qquad e_2 := \{X_2, X_4\}, \qquad E := \{e_1, e_2\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Thus $\mathrm{SHG}^{(2)} = (V, E)$ is a 2-SuperHyperGraph.

Intuitionistic memberships. Assign intuitionistic fuzzy memberships $A = (\mu_A, \nu_A) : V \to [0,1]^2$ by

$$(\mu_A,\nu_A)(X_1) = \Big(\frac{9}{10}, \frac{1}{20}\Big),$$
$$(\mu_A,\nu_A)(X_2) = \Big(\frac{4}{5}, \frac{1}{10}\Big),$$
$$(\mu_A,\nu_A)(X_3) = \Big(\frac{7}{10}, \frac{1}{5}\Big),$$
$$(\mu_A,\nu_A)(X_4) = \Big(\frac{17}{20}, \frac{2}{25}\Big),$$

so $0 \le \mu_A(X_i) + \nu_A(X_i) \le 1$ for all i. Assign intuitionistic memberships $B = (\mu_B, \nu_B) : E \to [0,1]^2$ by

$$(\mu_B,\nu_B)(e_1) = \Big(\frac{63}{100}, \frac{9}{50}\Big), \qquad (\mu_B,\nu_B)(e_2) = \Big(\frac{16}{25}, \frac{9}{100}\Big).$$

Then the compatibility constraints in Definition 7.9.3 hold:

$$\mu_B(e_1) = \frac{63}{100} \le \min\Big\{\frac{9}{10}, \frac{7}{10}\Big\} = \frac{7}{10}, \qquad \nu_B(e_1) = \frac{9}{50} \le \max\Big\{\frac{1}{20}, \frac{1}{5}\Big\} = \frac{1}{5},$$
$$\mu_B(e_2) = \frac{16}{25} \le \min\Big\{\frac{4}{5}, \frac{17}{20}\Big\} = \frac{4}{5}, \qquad \nu_B(e_2) = \frac{9}{100} \le \max\Big\{\frac{1}{10}, \frac{2}{25}\Big\} = \frac{1}{10}.$$

A drawing with one intersection. Place the four supervertices at the corners of a square and draw e_1 and e_2 as the two diagonals, so their interiors intersect once at the center point P_1. Figure 7.2 illustrates this drawing, where $z = 1$.

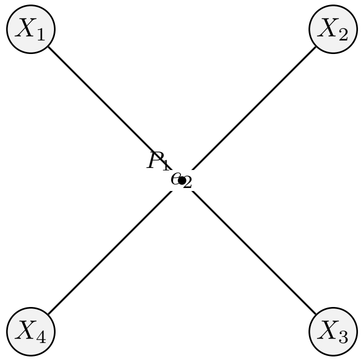


Figure 7.2.: A drawing of $\mathcal{S} = (V, E, A, B)$ with a single crossing point P_1 between e_1 and e_2.

Planarity value. Compute the intuitionistic superedge strengths $I_e = (M_e, N_e)$ (Definition 7.9.3):

$$I_{e_1} = \left(\frac{\mu_B(e_1)}{\min_{x\in e_1} \mu_A(x)}, \frac{\nu_B(e_1)}{\max_{x\in e_1} \nu_A(x)}\right) = \left(\frac{\frac{63}{100}}{\frac{7}{10}}, \frac{\frac{9}{50}}{\frac{1}{5}}\right) = \left(\frac{9}{10}, \frac{9}{10}\right),$$

$$I_{e_2} = \left(\frac{\frac{16}{25}}{\frac{4}{5}}, \frac{\frac{9}{100}}{\frac{1}{10}}\right) = \left(\frac{4}{5}, \frac{9}{10}\right).$$

Since the two traces intersect once at P_1, the intersecting value at P_1 is

$$I_{P_1} = (M_{P_1}, N_{P_1}) = \left(\frac{M_{e_1} + M_{e_2}}{2}, \frac{N_{e_1} + N_{e_2}}{2}\right) = \left(\frac{\frac{9}{10} + \frac{4}{5}}{2}, \frac{\frac{9}{10} + \frac{9}{10}}{2}\right) = \left(\frac{17}{20}, \frac{9}{10}\right).$$

Therefore the intuitionistic fuzzy planarity value of this drawing equals

$$f(\mathcal{S}) = (f_M, f_N) = \left(\frac{1}{1 + M_{P_1}}, \frac{1}{1 + N_{P_1}} \right) = \left(\frac{1}{1 + \frac{17}{20}}, \frac{1}{1 + \frac{9}{10}} \right) = \left(\frac{20}{37}, \frac{10}{19} \right).$$

In particular, since there is one crossing, $f(\mathcal{S}) \in (0,1)^2$; if we choose a crossing-free drawing ($z = 0$), then $f(\mathcal{S}) = (1,1)$ by Definition 7.9.3.

7.10 Neutrosophic Planar graph

A neutrosophic planar graph assigns T, I, F degrees to edges; planarity holds when the active-edge support embeds without crossings [265–268]. A neutrosophic planar hypergraph assigns T, I, F degrees to hyperedges; planarity holds when the induced co-occurrence support graph is planar. A neutrosophic planar superhypergraph assigns T, I, F degrees to superedges; planarity holds when the supervertex support graph admits a planar embedding.

Definition 7.10.1 (Neutrosophic planar graph and planarity value)**.** A *neutrosophic multigraph* is a finite multigraph $G^* = (V, E^*)$ together with three maps

$$T,\ I,\ F:\ E^* \longrightarrow [0,1],$$

assigning to each multiedge $e \in E^*$ its truth-, indeterminacy-, and falsity-degrees, respectively. (Equivalently, each e carries the triple $(T(e), I(e), F(e)) \in [0,1]^3$.)

Fix a geometric drawing of G^* in the plane, and let $P_1, \ldots, P_z$ be the (finite) set of intersection points between interiors of drawn edges (assume a general-position drawing so no three edges cross at one point). If two edges e_1, e_2 intersect at a point P, define the *intersecting neutrosophic value* at P by

$$\mathbf{I}_P \ := \ (T_P, I_P, F_P) \ := \ \left(\frac{T(e_1) + T(e_2)}{2}, \ \frac{I(e_1) + I(e_2)}{2}, \ \frac{F(e_1) + F(e_2)}{2} \right).$$

The *neutrosophic planarity value* of the chosen drawing is the triple

$$f(G) \ := \ (f_T, f_I, f_F) \ := \ \left(\frac{1}{1 + \sum_{i=1}^{z} T_{P_i}}, \frac{1}{1 + \sum_{i=1}^{z} I_{P_i}}, \frac{1}{1 + \sum_{i=1}^{z} F_{P_i}} \right) \ \in (0,1]^3.$$

We call G^* *neutrosophic planar* if it admits a drawing with $f(G) = (1,1,1)$, equivalently, a drawing with no edge intersections (so all three sums are 0).

Definition 7.10.2 (Neutrosophic planar hypergraph and planarity value)**.** A *neutrosophic hypergraph* is a quadruple

$$\mathcal{H} = (V, E, T, I, F),$$

where $V \neq \emptyset$ is a finite vertex set, E is a finite family of nonempty subsets of V (the *hyperedges*), and

$$T,\ I,\ F:\ E \longrightarrow [0,1]$$

assign to each hyperedge $e \in E$ its truth-, indeterminacy-, and falsity-degrees.

Fix a geometric drawing of $\mathcal{H}$ in the plane in which each vertex is drawn as a point and each hyperedge $e \in E$ is drawn as a connected 1-dimensional trace γ_e (e.g., an embedded tree) whose endpoints (leaves) are exactly the vertices of e. Intersections are counted only between interiors of traces of *distinct* hyperedges. Assume a general-position drawing so that each intersection point involves exactly two traces. Let $P_1, \dots, P_z$ be the set of such intersection points.

If two hyperedges e_1, e_2 have traces intersecting at a point P, define the *intersecting neutrosophic value* at P by

$$\mathbf{I}_P \;:=\; (T_P, I_P, F_P) \;:=\; \left(\frac{T(e_1)+T(e_2)}{2},\; \frac{I(e_1)+I(e_2)}{2},\; \frac{F(e_1)+F(e_2)}{2}\right).$$

The *neutrosophic planarity value* of the chosen drawing is

$$f(\mathcal{H}) \;:=\; (f_T, f_I, f_F) \;:=\; \left(\frac{1}{1+\sum_{i=1}^{z} T_{P_i}}, \frac{1}{1+\sum_{i=1}^{z} I_{P_i}}, \frac{1}{1+\sum_{i=1}^{z} F_{P_i}}\right) \in (0,1]^3.$$

We call $\mathcal{H}$ *neutrosophic planar* if it admits a drawing with $f(\mathcal{H}) = (1,1,1)$, i.e., a drawing with no intersections between interiors of distinct hyperedge traces.

Definition 7.10.3 (Neutrosophic planar n-SuperHyperGraph and planarity value)**.** Fix an integer $n \geq 0$ and a finite base set V_0. For each $k \geq 0$, define iterated powersets by

$$\mathcal{P}^0(V_0) = V_0, \qquad \mathcal{P}^{k+1}(V_0) = \mathcal{P}\big(\mathcal{P}^k(V_0)\big).$$

An *n-SuperHyperGraph* is a pair $\mathrm{SHG}^{(n)} = (V, E)$ with

$$V \subseteq \mathcal{P}^n(V_0), \qquad E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Elements of V are called *n-supervertices*, and elements of E are called *superedges*.

A *neutrosophic n-SuperHyperGraph* is a quintuple

$$\mathcal{S} = (V, E, T, I, F),$$

where (V, E) is an n-SuperHyperGraph and

$$T,\; I,\; F:\; E \longrightarrow [0,1]$$

assign to each superedge its truth-, indeterminacy-, and falsity-degrees.

Fix a geometric drawing of $\mathcal{S}$ in the plane in which each n-supervertex is drawn as a point and each superedge $e \in E$ is drawn as a connected 1-dimensional trace γ_e whose endpoints are exactly the n-supervertices in e. Intersections are counted only between interiors of traces of distinct superedges, and we assume a general-position drawing so that each intersection point involves exactly two traces. Let $P_1, \dots, P_z$ be the set of such intersection points.

If two superedges e_1, e_2 intersect at a point P, define the *intersecting neutrosophic value* at P by

$$\mathbf{I}_P \;:=\; (T_P, I_P, F_P) \;:=\; \left(\frac{T(e_1)+T(e_2)}{2},\; \frac{I(e_1)+I(e_2)}{2},\; \frac{F(e_1)+F(e_2)}{2}\right).$$

The *neutrosophic planarity value* of the chosen drawing is

$$f(\mathcal{S}) \;:=\; (f_T, f_I, f_F) \;:=\; \left(\frac{1}{1+\sum_{i=1}^{z} T_{P_i}}, \frac{1}{1+\sum_{i=1}^{z} I_{P_i}}, \frac{1}{1+\sum_{i=1}^{z} F_{P_i}}\right) \in (0,1]^3.$$

We call $\mathcal{S}$ *neutrosophic planar* if it admits a drawing with $f(\mathcal{S}) = (1,1,1)$, i.e., a drawing with no intersections between interiors of distinct drawn superedges.

Example 7.10.4 (A neutrosophic planar 2-SuperHyperGraph and its neutrosophic planarity value)**.** Let $V_0 = \{a, b, c, d\}$ and take $n = 2$. Define four 2-supervertices (elements of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$) by

$$X_1 := \{\{a\}\}, \qquad X_2 := \{\{b\}\}, \qquad X_3 := \{\{c\}\}, \qquad X_4 := \{\{d\}\},$$

and set

$$V := \{X_1, X_2, X_3, X_4\} \subseteq \mathcal{P}^2(V_0).$$

Define two superedges by

$$e_1 := \{X_1, X_2, X_3\}, \qquad e_2 := \{X_2, X_3, X_4\}, \qquad E := \{e_1, e_2\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Thus $\mathrm{SHG}^{(2)} = (V, E)$ is a finite 2-SuperHyperGraph.

Neutrosophic memberships on superedges. Define $T, I, F : E \to [0, 1]$ by

$$(T(e_1), I(e_1), F(e_1)) = (0.80, 0.10, 0.15), \qquad (T(e_2), I(e_2), F(e_2)) = (0.60, 0.25, 0.20).$$

(If your convention requires $T + I + F \leq 1$ or $T + I + F = 1$, you may rescale; the present example uses the most common independent-component convention.)

A drawing with one interior intersection. Fix a plane drawing in which X_1, X_2, X_3, X_4 are placed in convex position and e_1, e_2 are drawn as connected traces $\gamma_{e_1}, \gamma_{e_2}$ that connect exactly their endpoints, arranged so that the interiors intersect exactly once at a point P_1 and there are no other intersections. Hence $z = 1$.

Intersecting neutrosophic value. At the unique intersection point P_1,

$$\begin{aligned}\mathbf{I}_{P_1} &= (T_{P_1}, I_{P_1}, F_{P_1}) \\ &= \left(\frac{T(e_1) + T(e_2)}{2}, \frac{I(e_1) + I(e_2)}{2}, \frac{F(e_1) + F(e_2)}{2}\right) \\ &= \left(\frac{0.80 + 0.60}{2}, \frac{0.10 + 0.25}{2}, \frac{0.15 + 0.20}{2}\right) = (0.70, 0.175, 0.175).\end{aligned}$$

Neutrosophic planarity value. Since there is exactly one intersection point,

$$\begin{aligned}f(\mathcal{S}) = (f_T, f_I, f_F) &= \left(\frac{1}{1 + \sum_{i=1}^{1} T_{P_i}}, \frac{1}{1 + \sum_{i=1}^{1} I_{P_i}}, \frac{1}{1 + \sum_{i=1}^{1} F_{P_i}}\right) \\ &= \left(\frac{1}{1 + 0.70}, \frac{1}{1 + 0.175}, \frac{1}{1 + 0.175}\right) \\ &= \left(\frac{10}{17}, \frac{40}{47}, \frac{40}{47}\right).\end{aligned}$$

Thus $f(\mathcal{S}) \in (0, 1)^3$, reflecting that the chosen drawing has one crossing. If we instead select a crossing-free drawing ($z = 0$), then Definition 7.10.3 gives $f(\mathcal{S}) = (1, 1, 1)$, and $\mathcal{S}$ is neutrosophic planar.

7.11 Plithogenic Planar Graph

A plithogenic planar graph is a DAF/DCF-annotated graph whose active-edge support admits a crossing-free planar embedding [267]. A plithogenic planar hypergraph is a DAF/DCF-annotated hypergraph whose active hyperedges induce a planar vertex co-occurrence support graph. A plithogenic planar superhypergraph is a DAF/DCF-annotated n-superhypergraph whose active superedges induce a planar support graph on supervertices.

Definition 7.11.1 (Plithogenic planar graph and plithogenic planarity value)**.** Let $G^* = (V, E^*)$ be a finite (multi)graph.[1] A *plithogenic graph* (edge-based version) is a triple

$$\mathcal{G} = (G^*, \mathrm{DAF}, \mathrm{DCF}),$$

where

$$\mathrm{DAF} : E^* \to [0,1]$$

assigns to each edge its *degree of appurtenance* (DAF), and

$$\mathrm{DCF} : E^* \times E^* \to [0,1]$$

assigns to each ordered pair (e_1, e_2) its *degree of contradiction* (DCF).[2]

Fix a geometric drawing of G^* in the plane and let $P_1, \dots, P_z$ be the (finite) set of intersection points between interiors of drawn edges (assume a general-position drawing so no three edges cross at one point). If two edges e_1, e_2 intersect at a point P, define the *degree of intersection* at P by the ordered pair

$$\mathbf{I}_P \;:=\; \big(\mathrm{DAF}_P, \mathrm{DCF}_P\big) \;:=\; \left(\frac{\mathrm{DAF}(e_1) + \mathrm{DAF}(e_2)}{2}, \; \frac{\mathrm{DCF}(e_1, e_2) + \mathrm{DCF}(e_2, e_1)}{2}\right) \;\in [0,1]^2.$$

The *plithogenic planarity value* of the drawing is the pair

$$f(\mathcal{G}) \;:=\; (f_A, f_C) \;:=\; \left(\frac{1}{1 + \sum_{i=1}^{z} \mathrm{DAF}_{P_i}}, \frac{1}{1 + \sum_{i=1}^{z} \mathrm{DCF}_{P_i}}\right) \;\in (0,1]^2.$$

We call $\mathcal{G}$ a *plithogenic planar graph* (with respect to the chosen drawing) if $f(\mathcal{G}) = (1, 1)$, equivalently, if the drawing has no edge intersections (so $z = 0$, hence both sums vanish). More generally, smaller values of $f(\mathcal{G})$ indicate larger cumulative intersection effects in the drawing.

Definition 7.11.2 (Plithogenic planar hypergraph and plithogenic planarity value)**.** Let $H = (V, E)$ be a finite hypergraph, where $V \neq \emptyset$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. A *plithogenic hypergraph* (hyperedge-based version) is a triple

$$\mathcal{H} = (H, \mathrm{DAF}, \mathrm{DCF}),$$

where

$$\mathrm{DAF} : E \to [0,1], \qquad \mathrm{DCF} : E \times E \to [0,1]$$

assign degrees of appurtenance and contradiction to hyperedges (and ordered pairs of hyperedges), respectively.

[1] If one prefers simple graphs, take $E^* \subseteq \binom{V}{2}$.

[2] In many applications, $\mathrm{DCF}(e, e) = 0$ and one may assume symmetry $\mathrm{DCF}(e_1, e_2) = \mathrm{DCF}(e_2, e_1)$, but symmetry is not required here.

Fix a geometric drawing of H in the plane in which each vertex is drawn as a point and each hyperedge $e \in E$ is drawn as a connected 1-dimensional trace γ_e (e.g., an embedded tree) whose endpoints (leaves) are exactly the vertices in e. Intersections are counted only between interiors of traces of *distinct* hyperedges, and we assume a general-position drawing so that each intersection point involves exactly two traces. Let $P_1, \ldots, P_z$ be the set of such intersection points.

If hyperedges e_1, e_2 intersect at a point P, define the *degree of intersection* at P by

$$\mathbf{I}_P := \big(\mathrm{DAF}_P, \mathrm{DCF}_P\big) := \left(\frac{\mathrm{DAF}(e_1) + \mathrm{DAF}(e_2)}{2}, \frac{\mathrm{DCF}(e_1, e_2) + \mathrm{DCF}(e_2, e_1)}{2}\right) \in [0,1]^2.$$

The *plithogenic planarity value* of the drawing is

$$f(\mathcal{H}) := (f_A, f_C) := \left(\frac{1}{1 + \sum_{i=1}^{z} \mathrm{DAF}_{P_i}}, \frac{1}{1 + \sum_{i=1}^{z} \mathrm{DCF}_{P_i}}\right) \in (0,1]^2.$$

We call $\mathcal{H}$ a *plithogenic planar hypergraph* (with respect to the chosen drawing) if $f(\mathcal{H}) = (1,1)$, i.e., if the drawing has no intersections between interiors of distinct hyperedge traces.

Definition 7.11.3 (Plithogenic planar n-SuperHyperGraph and plithogenic planarity value)**.** Fix an integer $n \geq 0$ and a finite base set V_0. Define iterated powersets by

$$\mathcal{P}^0(V_0) = V_0, \qquad \mathcal{P}^{k+1}(V_0) = \mathcal{P}\big(\mathcal{P}^k(V_0)\big) \quad (k \geq 0).$$

An *n-SuperHyperGraph* is a pair $\mathrm{SHG}^{(n)} = (V, E)$ with

$$V \subseteq \mathcal{P}^n(V_0), \qquad E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Elements of V are *n-supervertices* and elements of E are *superedges*.

A *plithogenic n-SuperHyperGraph* (superedge-based version) is a triple

$$\mathcal{S} = \big(\mathrm{SHG}^{(n)}, \mathrm{DAF}, \mathrm{DCF}\big) = \big((V, E), \mathrm{DAF}, \mathrm{DCF}\big),$$

where

$$\mathrm{DAF} : E \to [0,1], \qquad \mathrm{DCF} : E \times E \to [0,1]$$

assign degrees of appurtenance and contradiction to superedges (and ordered pairs of superedges).

Fix a geometric drawing of (V, E) in the plane in which each n-supervertex is drawn as a point and each superedge $e \in E$ is drawn as a connected 1-dimensional trace γ_e whose endpoints are exactly the n-supervertices in e. Intersections are counted only between interiors of traces of distinct superedges, and we assume a general-position drawing so that each intersection point involves exactly two traces. Let $P_1, \ldots, P_z$ be the set of such intersection points.

If two superedges e_1, e_2 intersect at a point P, define the *degree of intersection* at P by

$$\mathbf{I}_P := \big(\mathrm{DAF}_P, \mathrm{DCF}_P\big) := \left(\frac{\mathrm{DAF}(e_1) + \mathrm{DAF}(e_2)}{2}, \frac{\mathrm{DCF}(e_1, e_2) + \mathrm{DCF}(e_2, e_1)}{2}\right) \in [0,1]^2.$$

The *plithogenic planarity value* of the drawing is

$$f(\mathcal{S}) := (f_A, f_C) := \left(\frac{1}{1 + \sum_{i=1}^{z} \mathrm{DAF}_{P_i}}, \frac{1}{1 + \sum_{i=1}^{z} \mathrm{DCF}_{P_i}}\right) \in (0,1]^2.$$

We call $\mathcal{S}$ a *plithogenic planar n-SuperHyperGraph* (with respect to the chosen drawing) if $f(\mathcal{S}) = (1,1)$, i.e., if the drawing has no intersections between interiors of distinct drawn superedges.

Example 7.11.4 (A plithogenic planar 2-SuperHyperGraph and its plithogenic planarity value)**.** Let $V_0 = \{a, b, c, d\}$ and take $n = 2$. Define four 2-supervertices (elements of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$) by

$$X_1 := \{\{a\}\}, \qquad X_2 := \{\{b\}\}, \qquad X_3 := \{\{c\}\}, \qquad X_4 := \{\{d\}\},$$

and set

$$V := \{X_1, X_2, X_3, X_4\} \subseteq \mathcal{P}^2(V_0).$$

Define two superedges by

$$e_1 := \{X_1, X_2, X_3\}, \qquad e_2 := \{X_2, X_3, X_4\}, \qquad E := \{e_1, e_2\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Thus $\mathrm{SHG}^{(2)} = (V, E)$ is a finite 2-SuperHyperGraph.

Plithogenic parameters. Define the degree of appurtenance $\mathrm{DAF} : E \to [0, 1]$ by

$$\mathrm{DAF}(e_1) = 0.8, \qquad \mathrm{DAF}(e_2) = 0.6,$$

and define the degree of contradiction $\mathrm{DCF} : E \times E \to [0, 1]$ (ordered pairs) by

$$\mathrm{DCF}(e_1, e_1) = 0, \quad \mathrm{DCF}(e_2, e_2) = 0, \quad \mathrm{DCF}(e_1, e_2) = 0.2, \quad \mathrm{DCF}(e_2, e_1) = 0.3.$$

Then $\mathcal{S} = \big((V, E), \mathrm{DAF}, \mathrm{DCF}\big)$ is a plithogenic 2-SuperHyperGraph in the sense of Definition 7.11.3.

A planar drawing (no interior intersections). Choose a plane drawing in which the supervertices are placed on a line in the order X_1, X_2, X_3, X_4, and draw each superedge trace γ_{e_1} and γ_{e_2} as a simple arc above the line that connects exactly the supervertices in that edge. Because the two edges share the endpoints X_2 and X_3 and can be drawn as nested arcs, their interiors can be arranged to be disjoint. Hence the set of interior intersection points is empty, so $z = 0$.

Plithogenic planarity value. Since $z = 0$, the sums over intersection points vanish, and therefore the plithogenic planarity value is

$$f(\mathcal{S}) = (f_A, f_C) = \left(\frac{1}{1 + \sum_{i=1}^{0} \mathrm{DAF}_{P_i}}, \frac{1}{1 + \sum_{i=1}^{0} \mathrm{DCF}_{P_i}} \right) = (1, 1).$$

Consequently, $\mathcal{S}$ is a *plithogenic planar* 2-SuperHyperGraph (with respect to the chosen drawing), as required by Definition 7.11.3.

7.12 Uncertain Planar graph

Uncertain planar graphs assign uncertainty degrees to vertices and edges; keep edges with nonzero degree; require resulting support graph to be planar simple. Uncertain planar hypergraphs assign degrees to vertices and hyperedges; activate nonzero hyperedges; build cooccurrence support graph; require this support graph planar in plane. Uncertain planar superhypergraphs assign degrees on n supervertices and superedges; keep nonzero superedges; form support graph on supervertices; demand planar embedding without crossings.

Definition 7.12.1 (Uncertain planar graph)**.** Let M be an uncertain model with degree domain $\mathrm{Dom}(M) \subseteq [0,1]^k$. Let $G^* = (V, E)$ be a finite simple undirected graph. An *uncertain graph of type* M is a triple $\mathcal{G}_M = (V, E, \mu_M)$ with $\mu_M : V \cup E \to \mathrm{Dom}(M)$.

Fix a distinguished zero tuple $\mathbf{0} = (0, \dots, 0) \in [0,1]^k$, and define the *active (support) edge set* by

$$E^+ \;:=\; \{\, e \in E : \; \mu_M(e) \neq \mathbf{0} \,\}.$$

Let $G^+ := (V, E^+)$ be the support graph.

We call $\mathcal{G}_M$ an *uncertain planar graph* if the support graph G^+ is planar, i.e., G^+ admits a planar embedding with no edge crossings.

Definition 7.12.2 (Uncertain planar hypergraph)**.** Let M be an uncertain model with degree domain $\mathrm{Dom}(M) \subseteq [0,1]^k$. Let $H^* = (V, \mathcal{E})$ be a finite hypergraph with $\emptyset \notin \mathcal{E} \subseteq \mathcal{P}(V)$. An *uncertain hypergraph of type* M is a triple $\mathcal{H}_M = (V, \mathcal{E}, \mu_M)$ with $\mu_M : V \cup \mathcal{E} \to \mathrm{Dom}(M)$.

Define the *active (support) hyperedge set* by

$$\mathcal{E}^+ \;:=\; \{\, e \in \mathcal{E} : \; \mu_M(e) \neq \mathbf{0} \,\}.$$

Define the *support (2-section) graph* of $\mathcal{H}_M$ by

$$\mathrm{supp}(\mathcal{H}_M) \;:=\; (V, E_{\mathrm{supp}}), \qquad E_{\mathrm{supp}} := \Big\{ \{u, v\} \in \tbinom{V}{2} : \; \exists\, e \in \mathcal{E}^+ \text{ with } \{u, v\} \subseteq e \Big\}.$$

We call $\mathcal{H}_M$ an *uncertain planar hypergraph* if $\mathrm{supp}(\mathcal{H}_M)$ is a planar graph.

Remark 7.12.3. The planarity notion in Definition 7.12.2 is imposed on the vertex-level 2-section (co-occurrence) graph. An alternative, also common in hypergraph theory, is to require the incidence graph $B(H^*)$ to be planar; the two definitions are not equivalent in general.

Definition 7.12.4 (Uncertain planar n-SuperHyperGraph)**.** Let M be an uncertain model with degree domain $\mathrm{Dom}(M) \subseteq [0,1]^k$. Let V_0 be a finite base set and $n \in \mathbb{N}_0$. Let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0 with

$$\emptyset \neq V \subseteq \mathcal{P}^n(V_0), \qquad \emptyset \neq E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

An *uncertain n-SuperHyperGraph of type* M is a triple

$$\mathcal{S}_M^{(n)} = (V, E, \mu_M), \qquad \mu_M : V \cup E \to \mathrm{Dom}(M).$$

Define the *active (support) superedge set* by

$$E^+ \;:=\; \{\, \varepsilon \in E : \; \mu_M(\varepsilon) \neq \mathbf{0} \,\}.$$

Define the *support graph* of $\mathcal{S}_M^{(n)}$ on the supervertex set V by

$$\mathrm{supp}(\mathcal{S}_M^{(n)}) \;:=\; (V, E_{\mathrm{supp}}), \qquad E_{\mathrm{supp}} := \Big\{ \{X, Y\} \in \tbinom{V}{2} : \; \exists\, \varepsilon \in E^+ \text{ with } \{X, Y\} \subseteq \varepsilon \Big\}.$$

We call $\mathcal{S}_M^{(n)}$ an *uncertain planar n-SuperHyperGraph* if $\mathrm{supp}(\mathcal{S}_M^{(n)})$ is a planar graph.

Remark 7.12.5. Definition 7.12.4 is intrinsic at the supervertex/superedge level. If desired, one may relate this to base-level structures by applying a flattening map (e.g., Flat_n) to supervertices, but flattening is not required for the planarity definition.

7.13 Fuzzy OuterPlanar Graph

A *fuzzy outerplanar graph* is a fuzzy graph whose positive-membership edges induce an outerplanar support graph, so all vertices can be placed on the outer face [273–276].

Definition 7.13.1 (Fuzzy outerplanar graph)**.** [273,274] A *fuzzy graph* is a triple $G = (V, \sigma, \mu)$, where $V \neq \emptyset$ is finite, $\sigma : V \to [0, 1]$ is the vertex–membership map, and $\mu : V \times V \to [0, 1]$ is a symmetric edge–membership map with $\mu(v, v) = 0$ and

$$\mu(u, v) \leq \min\{\sigma(u), \sigma(v)\} \qquad (\forall u, v \in V).$$

Let

$$E^+ \;:=\; \big\{\{u, v\} \subseteq V : \; u \neq v, \; \mu(u, v) > 0\big\}$$

be the set of *active* (crisp) edges, and let $G^+ = (V, E^+)$ be the *support graph.*

We say that G is a *fuzzy outerplanar graph* if the support graph G^+ admits a planar embedding in which

(i) no two edges cross except possibly at common endpoints, and

(ii) every vertex of V lies on the boundary of the outer (unbounded) face.

Equivalently, G is fuzzy outerplanar if and only if G^+ is outerplanar.

Definition 7.13.2 (Fuzzy outerplanar hypergraph)**.** A *fuzzy hypergraph* is a quadruple

$$\mathcal{H} = (V, E, \sigma, \mu),$$

where $V \neq \emptyset$ is finite, $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$ is a finite family of hyperedges, $\sigma : V \to [0, 1]$ is the vertex–membership map, and $\mu : E \to [0, 1]$ is the hyperedge–membership map satisfying the appurtenance constraint

$$\mu(e) \;\leq\; \min_{v \in e} \sigma(v) \qquad (\forall e \in E).$$

Let

$$E^+ \;:=\; \{\, e \in E : \mu(e) > 0 \,\}$$

be the set of *active* hyperedges.

Define the *support graph* of $\mathcal{H}$ as the (crisp) graph

$$\mathrm{supp}(\mathcal{H}) \;:=\; (V, E_{\mathrm{supp}}), \qquad E_{\mathrm{supp}} \;:=\; \Big\{\{u, v\} \subseteq V : \; u \neq v, \; \exists\, e \in E^+ \text{ with } u, v \in e\Big\},$$

i.e., $\{u, v\}$ is an edge in $\mathrm{supp}(\mathcal{H})$ whenever u and v co-occur in some active hyperedge.

We say that $\mathcal{H}$ is a *fuzzy outerplanar hypergraph* if its support graph $\mathrm{supp}(\mathcal{H})$ is outerplanar; equivalently, if $\mathrm{supp}(\mathcal{H})$ admits a planar embedding in which every vertex lies on the boundary of the outer face.

Definition 7.13.3 (Fuzzy outerplanar n-SuperHyperGraph)**.** Fix an integer $n \geq 0$ and a finite base set V_0. For each $k \geq 0$, define iterated powersets by

$$\mathcal{P}^0(V_0) = V_0, \qquad \mathcal{P}^{k+1}(V_0) = \mathcal{P}\big(\mathcal{P}^k(V_0)\big).$$

An *n-SuperHyperGraph* is a pair $\mathrm{SHG}^{(n)} = (V, E)$ such that

$$V \subseteq \mathcal{P}^n(V_0), \qquad E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

A *fuzzy n-SuperHyperGraph* is a quadruple

$$\mathcal{S} = (V, E, \sigma, \mu),$$

where (V, E) is an n-SuperHyperGraph, $\sigma : V \to [0, 1]$ assigns a membership degree to each n-supervertex, and $\mu : E \to [0, 1]$ assigns a membership degree to each superedge, satisfying

$$\mu(e) \;\leq\; \min_{x \in e} \sigma(x) \qquad (\forall e \in E).$$

Let $E^+ := \{\, e \in E : \mu(e) > 0 \,\}$ be the set of active superedges.

Define the *support graph* of $\mathcal{S}$ as the crisp graph

$$\mathrm{supp}(\mathcal{S}) \;:=\; (V, E_{\mathrm{supp}}), \qquad E_{\mathrm{supp}} \;:=\; \Big\{\{x, y\} \subseteq V :\; x \neq y,\; \exists\, e \in E^+ \text{ with } x, y \in e\Big\}.$$

Thus, two n-supervertices are adjacent in $\mathrm{supp}(\mathcal{S})$ precisely when they co-occur in some active superedge.

We say that $\mathcal{S}$ is a *fuzzy outerplanar n-SuperHyperGraph* if $\mathrm{supp}(\mathcal{S})$ is outerplanar; equivalently, if $\mathrm{supp}(\mathcal{S})$ admits a planar embedding in which every n-supervertex lies on the boundary of the outer face.

Example 7.13.4 (A fuzzy outerplanar 2-SuperHyperGraph)**.** Let $V_0 = \{a, b, c, d\}$ and take $n = 2$. Define four 2-supervertices (elements of $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$) by

$$X_1 := \{\{a\}\}, \qquad X_2 := \{\{b\}\}, \qquad X_3 := \{\{c\}\}, \qquad X_4 := \{\{d\}\},$$

and set

$$V := \{X_1, X_2, X_3, X_4\} \subseteq \mathcal{P}^2(V_0).$$

Define three superedges by

$$e_{12} := \{X_1, X_2\}, \qquad e_{23} := \{X_2, X_3\}, \qquad e_{34} := \{X_3, X_4\},$$

$$E := \{e_{12}, e_{23}, e_{34}\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Thus (V, E) is a 2-SuperHyperGraph.

Fuzzy memberships and admissibility. Assign supervertex memberships $\sigma : V \to [0, 1]$ by

$$\sigma(X_1) = 0.9, \qquad \sigma(X_2) = 0.8, \qquad \sigma(X_3) = 0.7, \qquad \sigma(X_4) = 0.6,$$

and superedge memberships $\mu : E \to [0, 1]$ by

$$\mu(e_{12}) = 0.8, \qquad \mu(e_{23}) = 0.7, \qquad \mu(e_{34}) = 0.5.$$

Then the admissibility constraints of Definition 7.13.3 hold:

$$\mu(e_{12}) = 0.8 \le \min\{0.9, 0.8\} = 0.8,$$
$$\mu(e_{23}) = 0.7 \le \min\{0.8, 0.7\} = 0.7,$$
$$\mu(e_{34}) = 0.5 \le \min\{0.7, 0.6\} = 0.6.$$

Hence $\mathcal{S} = (V, E, \sigma, \mu)$ is a fuzzy 2-SuperHyperGraph, and all three edges are active:

$$E^+ = \{e_{12}, e_{23}, e_{34}\}.$$

Outerplanarity of the support graph. The support graph $\mathrm{supp}(\mathcal{S}) = (V, E_{\mathrm{supp}})$ has an edge $\{X_i, X_j\}$ whenever X_i and X_j co-occur in some active superedge. Since the active superedges are exactly $\{X_1, X_2\}$, $\{X_2, X_3\}$, and $\{X_3, X_4\}$, we obtain

$$E_{\mathrm{supp}} = \big\{\{X_1, X_2\}, \{X_2, X_3\}, \{X_3, X_4\}\big\}.$$

Thus $\mathrm{supp}(\mathcal{S})$ is the path

$$X_1 \text{—} X_2 \text{—} X_3 \text{—} X_4,$$

which is outerplanar (indeed, every tree is outerplanar). Therefore $\mathcal{S}$ is a fuzzy outerplanar 2-SuperHyperGraph in the sense of Definition 7.13.3.

7.14 Intuitionistic fuzzy outerplanar graph

An intuitionistic fuzzy outerplanar graph is an intuitionistic fuzzy graph whose positive-membership edges form an outerplanar support graph, with all vertices on the outer face.

Definition 7.14.1 (Intuitionistic fuzzy outerplanar graph)**.** An *intuitionistic fuzzy graph* is a tuple

$$G = (V, E, \mu_V, \nu_V, \mu_E, \nu_E),$$

where $V \neq \emptyset$ is finite, $E \subseteq \binom{V}{2}$ is a (crisp) edge set, and

$$\mu_V, \nu_V : V \to [0, 1], \qquad \mu_E, \nu_E : E \to [0, 1]$$

satisfy, for all $v \in V$ and $uv \in E$,

$$0 \le \mu_V(v) + \nu_V(v) \le 1, \qquad 0 \le \mu_E(uv) + \nu_E(uv) \le 1,$$

together with the standard compatibility constraint

$$\mu_E(uv) \ \le \ \min\{\mu_V(u), \mu_V(v)\}.$$

(Optionally, one may also require $\nu_E(uv) \ge \max\{\nu_V(u), \nu_V(v)\}$; it is not needed for outerplanarity.)

Define the set of *active edges* by

$$E^+ \ := \ \{\, uv \in E : \mu_E(uv) > 0 \,\},$$

and let $G^+ = (V, E^+)$ be the *support graph*.

We say that G is *intuitionistic fuzzy outerplanar* if the support graph G^+ admits a planar embedding in which

(i) no two edges cross except possibly at common endpoints, and

(ii) every vertex lies on the boundary of the outer (unbounded) face.

Equivalently, G is intuitionistic fuzzy outerplanar if and only if G^+ is an outerplanar (crisp) graph.

Definition 7.14.2 (Intuitionistic fuzzy outerplanar hypergraph)**.** An *intuitionistic fuzzy hypergraph* is a tuple

$$\mathcal{H} = (V, E, \mu_V, \nu_V, \mu_E, \nu_E),$$

where $V \neq \emptyset$ is finite, $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$ is a finite family of hyperedges, and

$$\mu_V, \nu_V : V \to [0,1], \qquad \mu_E, \nu_E : E \to [0,1]$$

satisfy, for all $v \in V$ and $e \in E$,

$$0 \leq \mu_V(v) + \nu_V(v) \leq 1, \qquad 0 \leq \mu_E(e) + \nu_E(e) \leq 1,$$

together with the appurtenance constraint

$$\mu_E(e) \;\leq\; \min_{v \in e} \mu_V(v) \qquad (\forall e \in E).$$

Let $E^+ := \{\, e \in E : \mu_E(e) > 0 \,\}$ be the set of *active hyperedges*.

Define the *support graph* of $\mathcal{H}$ as the crisp graph

$$\operatorname{supp}(\mathcal{H}) \;:=\; (V, E_{\text{supp}}), \qquad E_{\text{supp}} \;:=\; \Big\{ \{u,v\} \subseteq V :\; u \neq v,\; \exists\, e \in E^+ \text{ with } u, v \in e \Big\}.$$

Thus $\{u, v\} \in E_{\text{supp}}$ if and only if u and v co-occur in some active hyperedge.

We say that $\mathcal{H}$ is *intuitionistic fuzzy outerplanar* if $\operatorname{supp}(\mathcal{H})$ is an outerplanar graph; equivalently, if $\operatorname{supp}(\mathcal{H})$ admits a planar embedding in which every vertex lies on the boundary of the outer face.

Definition 7.14.3 (Intuitionistic fuzzy outerplanar n-SuperHyperGraph)**.** Fix an integer $n \geq 0$ and a finite base set V_0. For each $k \geq 0$, define iterated powersets by

$$\mathcal{P}^0(V_0) = V_0, \qquad \mathcal{P}^{k+1}(V_0) = \mathcal{P}\big(\mathcal{P}^k(V_0)\big).$$

An *n-SuperHyperGraph* is a pair $\mathrm{SHG}^{(n)} = (V, E)$ with

$$V \subseteq \mathcal{P}^n(V_0), \qquad E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

A *intuitionistic fuzzy n-SuperHyperGraph* is a tuple

$$\mathcal{S} = (V, E, \mu_V, \nu_V, \mu_E, \nu_E),$$

where (V, E) is an n-SuperHyperGraph,

$$\mu_V, \nu_V : V \to [0,1], \qquad \mu_E, \nu_E : E \to [0,1],$$

satisfy, for all $x \in V$ and $e \in E$,

$$0 \leq \mu_V(x) + \nu_V(x) \leq 1, \qquad 0 \leq \mu_E(e) + \nu_E(e) \leq 1,$$

and the appurtenance constraint

$$\mu_E(e) \ \leq \ \min_{x \in e} \mu_V(x) \qquad (\forall e \in E).$$

Let $E^+ := \{ e \in E : \mu_E(e) > 0 \}$ be the set of *active superedges.*

Define the *support graph* of $\mathcal{S}$ as the crisp graph

$$\mathrm{supp}(\mathcal{S}) \ := \ (V, E_{\mathrm{supp}}), \qquad E_{\mathrm{supp}} \ := \ \Big\{ \{x, y\} \subseteq V : \ x \neq y, \ \exists\, e \in E^+ \text{ with } x, y \in e \Big\}.$$

Thus two n-supervertices are adjacent in $\mathrm{supp}(\mathcal{S})$ precisely when they co-occur in some active superedge.

We say that $\mathcal{S}$ is *intuitionistic fuzzy outerplanar* if $\mathrm{supp}(\mathcal{S})$ is an outerplanar graph; equivalently, if $\mathrm{supp}(\mathcal{S})$ admits a planar embedding in which every n-supervertex lies on the boundary of the outer face.

7.15 Neutrosophic OuterPlanar graph

Neutrosophic OuterPlanar graph is a neutrosophic graph whose active edges, determined by positive truth-membership, form an outerplanar support graph embedding [267].

Definition 7.15.1 (Neutrosophic outerplanar graph and outerplanarity value)**.** A *(single-valued) neutrosophic graph* may be modeled by a crisp graph $G^* = (V, E)$ together with vertex and edge degree triples

$$(T_V, I_V, F_V) : V \to [0,1]^3, \qquad (T_E, I_E, F_E) : E \to [0,1]^3,$$

subject to the standard SVNS consistency constraints (componentwise bounds).

We say that G is *neutrosophic outerplanar* if it can be embedded in the plane so that

(i) no two edges intersect except at their endpoints, and

(ii) all vertices lie on the boundary of the exterior (outer) region.

More generally, for a fixed drawing with intersection points $P_1, \ldots, P_z$ between interiors of edges, one may define the *intersecting value* at each P_i as a triple (typically an average of the incident edge triples), and then define the *neutrosophic outerplanarity value*

$$f(G) = (f_T, f_I, f_F) := \left(\frac{1}{1 + \sum_{i=1}^{z} T_{P_i}}, \ \frac{1}{1 + \sum_{i=1}^{z} I_{P_i}}, \ \frac{1}{1 + \sum_{i=1}^{z} F_{P_i}} \right),$$

so that $f(G) = (1, 1, 1)$ when the drawing has no intersections.

Definition 7.15.2 (Neutrosophic outerplanar hypergraph)**.** Let $H^* = (V, \mathcal{E})$ be a finite hypergraph with $\emptyset \notin \mathcal{E} \subseteq \mathcal{P}(V)$. A *(single-valued) neutrosophic hypergraph* on H^* is a tuple

$$\mathcal{H} = \big(V, \mathcal{E},\ T_V, I_V, F_V,\ T_{\mathcal{E}}, I_{\mathcal{E}}, F_{\mathcal{E}}\big),$$

where

$$T_V, I_V, F_V : V \to [0,1], \qquad T_{\mathcal{E}}, I_{\mathcal{E}}, F_{\mathcal{E}} : \mathcal{E} \to [0,1],$$

satisfy $0 \le T_V(v) + I_V(v) + F_V(v) \le 3$ for all $v \in V$ and $0 \le T_{\mathcal{E}}(e) + I_{\mathcal{E}}(e) + F_{\mathcal{E}}(e) \le 3$ for all $e \in \mathcal{E}$. (Optionally, one may impose appurtenance constraints such as $T_{\mathcal{E}}(e) \le \min_{v \in e} T_V(v)$, and similarly for $I_{\mathcal{E}}, F_{\mathcal{E}}$.)

Define the *active hyperedge set* by

$$\mathcal{E}^+ \ :=\ \{\, e \in \mathcal{E} :\ T_{\mathcal{E}}(e) + I_{\mathcal{E}}(e) + F_{\mathcal{E}}(e) > 0 \,\}.$$

Define the *support (2-section) graph* of $\mathcal{H}$ as

$$\operatorname{supp}(\mathcal{H}) \ :=\ (V, E_{\text{supp}}), \qquad E_{\text{supp}} := \Big\{ \{u,v\} \in \tbinom{V}{2} :\ \exists\, e \in \mathcal{E}^+ \text{ with } \{u,v\} \subseteq e \Big\}.$$

We call $\mathcal{H}$ a *neutrosophic outerplanar hypergraph* if $\operatorname{supp}(\mathcal{H})$ is an outerplanar graph (equivalently, it has an embedding with all vertices on the outer face).

Definition 7.15.3 (Neutrosophic outerplanar n-SuperHyperGraph)**.** Fix $n \ge 0$ and a finite base set V_0. Let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph with

$$V \subseteq \mathcal{P}^n(V_0), \qquad E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

A *(single-valued) neutrosophic n-SuperHyperGraph* is a tuple

$$\mathcal{S} = \big(V, E,\ T_V, I_V, F_V,\ T_E, I_E, F_E\big),$$

where

$$T_V, I_V, F_V : V \to [0,1], \qquad T_E, I_E, F_E : E \to [0,1],$$

satisfy $0 \le T_V(X) + I_V(X) + F_V(X) \le 3$ for all $X \in V$ and $0 \le T_E(\varepsilon) + I_E(\varepsilon) + F_E(\varepsilon) \le 3$ for all $\varepsilon \in E$, together with the (natural) appurtenance constraints

$$T_E(\varepsilon) \le \min_{X \in \varepsilon} T_V(X), \qquad I_E(\varepsilon) \le \min_{X \in \varepsilon} I_V(X), \qquad F_E(\varepsilon) \le \min_{X \in \varepsilon} F_V(X) \qquad (\forall \varepsilon \in E).$$

Define the *active superedge set* by

$$E^+ \ :=\ \{\, \varepsilon \in E :\ T_E(\varepsilon) + I_E(\varepsilon) + F_E(\varepsilon) > 0 \,\}.$$

Define the *support graph* of $\mathcal{S}$ on the supervertex set V by

$$\operatorname{supp}(\mathcal{S}) \ :=\ (V, E_{\text{supp}}), \qquad E_{\text{supp}} := \Big\{ \{X,Y\} \in \tbinom{V}{2} :\ \exists\, \varepsilon \in E^+ \text{ with } \{X,Y\} \subseteq \varepsilon \Big\}.$$

We call $\mathcal{S}$ a *neutrosophic outerplanar n-SuperHyperGraph* if $\operatorname{supp}(\mathcal{S})$ is an outerplanar graph.

7.16 Plithogenic OuterPlanar graph

Plithogenic OuterPlanar graph is a plithogenic graph whose active edges (nonzero appurtenance under DAF/DCF) admit an outerplanar embedding with all vertices on boundary outer.

Definition 7.16.1 (Plithogenic outerplanar graph)**.** Let $G^* = (V, E)$ be a finite simple undirected graph. A *plithogenic graph* (edge-based version) on G^* is a triple

$$\mathcal{G} = (G^*, \mathrm{DAF}, \mathrm{DCF}),$$

where

$$\mathrm{DAF} : E \to [0,1] \quad \text{and} \quad \mathrm{DCF} : E \times E \to [0,1]$$

assign, respectively, a *degree of appurtenance* (DAF) to each edge and a *degree of contradiction* (DCF) to each ordered pair of edges.

Define the *active edge set* (support) by

$$E^+ \ := \ \{\, e \in E : \ \mathrm{DAF}(e) > 0 \,\},$$

and let $G^+ := (V, E^+)$ be the support graph. We call $\mathcal{G}$ a *plithogenic outerplanar graph* if G^+ is outerplanar, i.e., if G^+ admits a planar embedding in which every vertex lies on the boundary of the outer face.

Remark 7.16.2. Definition 7.16.1 separates the *geometric* property (outerplanarity of the support) from the *plithogenic* annotation $(\mathrm{DAF}, \mathrm{DCF})$. If one prefers an intersection-sensitive notion, one may additionally evaluate a fixed drawing by attaching to each edge-crossing point P a pair $(\mathrm{DAF}_P, \mathrm{DCF}_P)$ computed from the incident edges; outerplanarity then corresponds to the absence of crossings (hence vacuously $(\mathrm{DAF}_P, \mathrm{DCF}_P) = (0,0)$ for all P).

Definition 7.16.3 (Plithogenic outerplanar hypergraph)**.** Let $H^* = (V, \mathcal{E})$ be a finite hypergraph with $\emptyset \notin \mathcal{E} \subseteq \mathcal{P}(V)$. A *plithogenic hypergraph* (hyperedge-based version) on H^* is a triple

$$\mathcal{H} = (H^*, \mathrm{DAF}, \mathrm{DCF}),$$

where

$$\mathrm{DAF} : \mathcal{E} \to [0,1], \qquad \mathrm{DCF} : \mathcal{E} \times \mathcal{E} \to [0,1]$$

assign degrees of appurtenance and contradiction to hyperedges (and ordered pairs of hyperedges).

Define the *active hyperedge set* by

$$\mathcal{E}^+ \ := \ \{\, e \in \mathcal{E} : \ \mathrm{DAF}(e) > 0 \,\}.$$

Define the *support (2-section) graph* of $\mathcal{H}$ as

$$\mathrm{supp}(\mathcal{H}) \ := \ (V, E_{\mathrm{supp}}), \qquad E_{\mathrm{supp}} := \Big\{ \{u, v\} \in \tbinom{V}{2} : \ \exists\, e \in \mathcal{E}^+ \text{ with } \{u, v\} \subseteq e \Big\}.$$

We call $\mathcal{H}$ a *plithogenic outerplanar hypergraph* if $\mathrm{supp}(\mathcal{H})$ is outerplanar.

Remark 7.16.4. The support graph $\mathrm{supp}(\mathcal{H})$ records pairwise co-occurrence in active hyperedges; thus outerplanarity is imposed at the vertex level. This choice yields a clean extension consistent with the outerplanar-graph case.

7.17 Uncertain OuterPlanar graph

An Uncertain OuterPlanar graph is an uncertain graph of type M whose active edges, having nonzero degree, form an outerplanar support graph embedding in plane. An Uncertain Outer-Planar n-superhypergraph is an uncertain n-superhypergraph of type M whose active superedges induce an outerplanar support graph; planar embedding exists on supervertices only.

Definition 7.17.1 (Uncertain outerplanar graph)**.** Let M be an uncertain model with degree domain $\mathrm{Dom}(M) \subseteq [0,1]^k$. Let $G^* = (V, E)$ be a finite simple undirected graph. An *uncertain graph of type M* is a triple $\mathcal{G}_M = (V, E, \mu_M)$, with $\mu_M : V \cup E \to \mathrm{Dom}(M)$.

Fix a distinguished zero tuple $\mathbf{0} = (0, \dots, 0) \in [0,1]^k$, and define the *active (support) edge set* by

$$E^+ := \{ e \in E : \mu_M(e) \neq \mathbf{0} \}.$$

Let $G^+ := (V, E^+)$ be the support graph.

We call $\mathcal{G}_M$ an *uncertain outerplanar graph* if the support graph G^+ is outerplanar, i.e., G^+ admits a planar embedding in which every vertex lies on the boundary of the outer face.

Definition 7.17.2 (Uncertain outerplanar hypergraph)**.** Let M be an uncertain model with degree domain $\mathrm{Dom}(M) \subseteq [0,1]^k$. Let $H^* = (V, \mathcal{E})$ be a finite hypergraph with $\emptyset \notin \mathcal{E} \subseteq \mathcal{P}(V)$. An *uncertain hypergraph of type M* is a triple $\mathcal{H}_M = (V, \mathcal{E}, \mu_M)$ with $\mu_M : V \cup \mathcal{E} \to \mathrm{Dom}(M)$.

Define the *active (support) hyperedge set* by

$$\mathcal{E}^+ := \{ e \in \mathcal{E} : \mu_M(e) \neq \mathbf{0} \}.$$

Define the *support (2-section) graph* of $\mathcal{H}_M$ by

$$\mathrm{supp}(\mathcal{H}_M) := (V, E_{\mathrm{supp}}), \qquad E_{\mathrm{supp}} := \Big\{ \{u,v\} \in \tbinom{V}{2} : \exists\, e \in \mathcal{E}^+ \text{ with } \{u,v\} \subseteq e \Big\}.$$

We call $\mathcal{H}_M$ an *uncertain outerplanar hypergraph* if $\mathrm{supp}(\mathcal{H}_M)$ is outerplanar.

Remark 7.17.3. The use of the 2-section support graph yields a clean outerplanarity notion at the vertex level. If one prefers an incidence-based definition, one may alternatively require the incidence graph $B(H^*)$ to be outerplanar; both are reasonable but they are not equivalent in general.

Definition 7.17.4 (Uncertain outerplanar n-SuperHyperGraph)**.** Let M be an uncertain model with degree domain $\mathrm{Dom}(M) \subseteq [0,1]^k$. Let V_0 be a finite base set and $n \in \mathbb{N}_0$. Let $\mathrm{SHG}^{(n)} = (V, E)$ be an n-SuperHyperGraph on V_0 with

$$\emptyset \neq V \subseteq \mathcal{P}^n(V_0), \qquad \emptyset \neq E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

An *uncertain n-SuperHyperGraph of type M* is a triple

$$\mathcal{S}_M^{(n)} = (V, E, \mu_M), \qquad \mu_M : V \cup E \to \mathrm{Dom}(M).$$

Define the *active (support) superedge set* by

$$E^+ \;:=\; \{\,\varepsilon \in E :\; \mu_M(\varepsilon) \neq \mathbf{0}\,\}.$$

Define the *support graph* of $\mathcal{S}_M^{(n)}$ on the supervertex set V by

$$\operatorname{supp}(\mathcal{S}_M^{(n)}) \;:=\; (V, E_{\mathrm{supp}}), \qquad E_{\mathrm{supp}} := \Big\{\{X,Y\} \in \tbinom{V}{2} :\; \exists\,\varepsilon \in E^+ \text{ with } \{X,Y\} \subseteq \varepsilon\Big\}.$$

We call $\mathcal{S}_M^{(n)}$ an *uncertain outerplanar n-SuperHyperGraph* if $\operatorname{supp}(\mathcal{S}_M^{(n)})$ is outerplanar.

Remark 7.17.5. Definition 7.17.4 is intrinsic at the supervertex/superedge level. If desired, one may connect this to a base-level viewpoint by applying a flattening map (e.g., Flat_n) to supervertices, but flattening is not required for the outerplanarity definition.

Chapter 8

Hierarchical Graph Structure

In this chapter, we investigate a Hierarchical Graph Structure in order to model connections between different levels.

8.1 Hierarchical SuperHyperGraphs

A hierarchical superhypergraph is a superhypergraph whose vertices live across multiple powerset levels, with edges allowed to join mixed-level supervertices, while maintaining downward-closure coherence. In this book, we primarily work with n-SuperHyperGraphs.

Definition 8.1.1 (Hierarchical SuperHyperGraph of height r)**.** Let V_0 be a finite, nonempty base set. For $k \geq 0$ define iterated powersets

$$\mathcal{P}^0(V_0) := V_0, \qquad \mathcal{P}^{k+1}(V_0) := \mathcal{P}\big(\mathcal{P}^k(V_0)\big),$$

and fix an integer $r \geq 0$. Set the *hierarchical universe*

$$\mathcal{U}_r(V_0) \;:=\; \bigcup_{k=0}^{r} \Big(\mathcal{P}^k(V_0) \setminus \{\emptyset\}\Big).$$

For $x \in \mathcal{U}_r(V_0)$, define its *level*

$$\ell(x) \;:=\; \min\{\, k \in \{0, 1, \ldots, r\} : x \in \mathcal{P}^k(V_0)\,\}.$$

A *hierarchical superhypergraph of height* r on V_0 is a pair

$$\mathbb{H}^{\langle r \rangle} = (V, E)$$

such that

(H1) (*Hierarchical vertex set*) V is a finite nonempty set with

$$V \subseteq \mathcal{U}_r(V_0).$$

Elements of V are called *hierarchical supervertices.*

(H2) (*Cross-level edges*) E is a finite family of nonempty subsets of V:

$$E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Elements of E are called *hierarchical superhyperedges*. In particular, a superhyperedge may contain supervertices of *different* levels.

(H3) (*Coherence / downward closure*) If $X \in V$ and $\ell(X) \geq 1$, then

$$X \subseteq V.$$

Equivalently, whenever a higher-level supervertex is present, all its immediate constituents are also present as supervertices.

For each $k \in \{0, \ldots, r\}$ we define the *k-th layer* by

$$V_k := \{ x \in V : \ell(x) = k \}, \qquad \text{so that} \qquad V = \dot{\bigcup}_{k=0}^{r} V_k.$$

Remark 8.1.2 (Relation to ordinary n-SuperHyperGraphs)**.** If, for some $n \leq r$, the hierarchical vertex set satisfies $V \subseteq \mathcal{P}^n(V_0)$, then every supervertex has level $\leq n$ and (V, E) is an n-SuperHyperGraph in the sense of Definition 2.1.6. Conversely, any n-SuperHyperGraph $\mathrm{SHG}^{(n)} = (V, E)$ is a hierarchical superhypergraph of height $r = n$ (by taking the same V and E); the novelty of Definition 8.1.1 is that it also permits *mixed-level* supervertices inside a single superhyperedge.

Example 8.1.3 (A hierarchical SuperHyperGraph of height $r = 2$)**.** Let the base set be

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

For $r = 2$, the hierarchical universe is $\mathcal{U}_2(V_0) = \big(\mathrm{PS}^0(V_0) \cup \mathrm{PS}^1(V_0) \cup \mathrm{PS}^2(V_0)\big) \setminus \{\emptyset\}$.

Hierarchical supervertices. We choose supervertices from three levels:

$$V_0^* := \{a, b, c\} \subseteq \mathrm{PS}^0(V_0), \qquad V_1^* := \{\{a, b\}, \{b, c\}\} \subseteq \mathrm{PS}^1(V_0),$$

and a single level-2 supervertex

$$X := \big\{\{a, b\}, \{b, c\}\big\} \in \mathrm{PS}^2(V_0) = \mathrm{PS}(\mathrm{PS}(V_0)).$$

Define the vertex set by

$$V := V_0^* \,\dot{\cup}\, V_1^* \,\dot{\cup}\, \{X\} \subseteq \mathcal{U}_2(V_0).$$

Then the level map satisfies

$$\ell(a) = \ell(b) = \ell(c) = 0, \quad \ell(\{a, b\}) = \ell(\{b, c\}) = 1, \quad \ell(X) = 2.$$

Coherence (downward closure). Since $\ell(X) = 2 \geq 1$, condition (H3) requires $X \subseteq V$, i.e., every element of X must already be a supervertex. Indeed, $X = \{\{a, b\}, \{b, c\}\}$ and both $\{a, b\}, \{b, c\} \in V_1^* \subseteq V$. Moreover, because $\ell(\{a, b\}) = \ell(\{b, c\}) = 1$, coherence also requires $\{a, b\} \subseteq V$ and $\{b, c\} \subseteq V$, which holds because $a, b, c \in V_0^* \subseteq V$.

Cross-level superhyperedges. Define a family of hierarchical superhyperedges $E \subseteq \mathrm{PS}(V) \setminus \{\emptyset\}$ by

$$e_1 := \{a, \{a, b\}\}, \qquad e_2 := \{\{a, b\}, X\}, \qquad e_3 := \{b, \{b, c\}, X\}, \qquad E := \{e_1, e_2, e_3\}.$$

Each edge mixes levels (for instance, e_3 contains level 0, level 1, and level 2 vertices), which is allowed by (H2). Therefore

$$\mathbb{H}^{\langle 2\rangle} := (V, E)$$

is a hierarchical superhypergraph of height 2 on V_0 in the sense of Definition 8.1.1.

Layer decomposition. The layers are

$$V_0 = \{a, b, c\}, \qquad V_1 = \{\{a, b\}, \{b, c\}\}, \qquad V_2 = \{X\}, \qquad V = \dot{\bigcup}_{k=0}^{2} V_k.$$

8.2 Hierarchical Path

A hierarchical path is a simple vertex sequence where consecutive vertices are adjacent either by sharing a hyperedge or by immediate constituent relations across consecutive levels.

Definition 8.2.1 (Hierarchical adjacency and hierarchical path)**.** Let $\mathbb{H}^{\langle r\rangle} = (V, E)$ be a hierarchical superhypergraph of height r on V_0 (Definition 8.1.1), with level map $\ell : V \to \{0, 1, \dots, r\}$.

(1) Immediate constituents. For $X, Y \in V$, we say that Y is an *immediate constituent* of X, and write $Y \prec X$, if

$$Y \in X \qquad \text{and} \qquad \ell(X) = \ell(Y) + 1.$$

(Condition (H3) guarantees $Y \in V$ whenever $X \in V$ and $\ell(X) \geq 1$.)

(2) Hierarchical adjacency. Define a symmetric adjacency relation $\sim_{\mathbb{H}}$ on V by declaring that, for distinct $x, y \in V$,

$$x \sim_{\mathbb{H}} y \quad \Longleftrightarrow \quad \Big(\exists\, \varepsilon \in E : \ \{x, y\} \subseteq \varepsilon\Big) \ \text{ or } \ (x \prec y) \ \text{ or } \ (y \prec x).$$

Thus two hierarchical supervertices are adjacent either *horizontally* (they co-occur in some superhyperedge), or *vertically* (one is an immediate constituent of the other).

(3) Hierarchical path. A *hierarchical path* in $\mathbb{H}^{\langle r\rangle}$ from $s \in V$ to $t \in V$ is a vertex sequence

$$P = (v_0, v_1, \dots, v_k) \qquad (k \geq 0),$$

such that:

(i) $v_0 = s$ and $v_k = t$;

(ii) $v_{i-1} \sim_{\mathbb{H}} v_i$ for every $i = 1, \dots, k$;

(iii) $v_0, \ldots, v_k$ are pairwise distinct (so P is *simple*).

The integer k is called the *length* of P.

Remark 8.2.2. If one deletes the vertical adjacency $\prec$ from Definition 8.2.1 and keeps only co-membership in superhyperedges, then a hierarchical path reduces to a (Berge-type) path in the underlying hypergraph (V, E). The present definition explicitly exploits the hierarchical constituent structure.

Example 8.2.3 (A hierarchical path with vertical and horizontal steps)**.** Let $V_0 := \{a, b, c\}$ and $r := 1$. Consider the hierarchical universe $\mathcal{U}_1(V_0) = V_0 \cup (\mathcal{P}(V_0) \setminus \{\emptyset\})$. Let

$$V := \{a, b, c,\ A,\ B\}, \qquad A := \{a, b\}, \quad B := \{b, c\},$$

so $\ell(a) = \ell(b) = \ell(c) = 0$ and $\ell(A) = \ell(B) = 1$. Note that (H3) holds since $A, B \in V$ implies their constituents a, b, c are also in V. Define the edge family

$$E := \{\varepsilon\}, \qquad \varepsilon := \{A, B\},$$

which is a mixed-level superhyperedge (here both endpoints happen to be level 1).

Define immediate constituents by membership: $a \prec A$, $b \prec A$, $b \prec B$, $c \prec B$. Then the sequence

$$P = (a,\ A,\ B,\ c)$$

is a hierarchical path from a to c:

$$a \sim_{\mathbb{H}} A \ \ (\text{since } a \prec A), \qquad A \sim_{\mathbb{H}} B \ \ (\text{since } \{A, B\} \subseteq \varepsilon \in E), \qquad B \sim_{\mathbb{H}} c \ \ (\text{since } c \prec B).$$

Hence P connects a base-level vertex a to a base-level vertex c by moving upward to A, horizontally across the superhyperedge ε, and downward from B to c.

8.3 Hierarchical Tree

A hierarchical tree is a hierarchical superhypergraph whose hierarchical adjacency graph is connected and acyclic, yielding a unique hierarchical path between any two vertices.

Definition 8.3.1 (Hierarchical Tree)**.** Let $\mathbb{H}^{\langle r \rangle} = (V, E)$ be a hierarchical superhypergraph of height r. We call $\mathbb{H}^{\langle r \rangle}$ a *hierarchical tree* if its hierarchical adjacency graph $G_{\text{hier}}(\mathbb{H}^{\langle r \rangle})$ (Definition **??**) is a tree; that is,

$$G_{\text{hier}}(\mathbb{H}^{\langle r \rangle}) \text{ is connected and contains no (graph) cycle.}$$

Equivalently, $\mathbb{H}^{\langle r \rangle}$ is a hierarchical tree if and only if for every pair of distinct $u, v \in V$ there exists a *unique* hierarchical path from u to v (in the sense of Definition 8.2.1).

Example 8.3.2 (A hierarchical tree of height 2)**.** Let $V_0 := \{a, b, c, d\}$ and $r := 2$. Define

$$A := \{a, b\} \in \mathcal{P}^1(V_0), \qquad B := \{c, d\} \in \mathcal{P}^1(V_0), \qquad R := \{A, B\} \in \mathcal{P}^2(V_0).$$

Let

$$V := \{a, b, c, d, A, B, R\} \subseteq \mathcal{U}_2(V_0), \qquad E := \emptyset.$$

Then (H3) holds: since $R \in V$ we have $A, B \in V$, and since $A, B \in V$ we have $a, b, c, d \in V$. The immediate constituent relations are

$$a \prec A, \quad b \prec A, \quad c \prec B, \quad d \prec B, \quad A \prec R, \quad B \prec R.$$

Hence the vertical edges of $G_{\text{hier}}(\mathbb{H}^{\langle 2\rangle})$ are

$$\{a, A\}, \{b, A\}, \{c, B\}, \{d, B\}, \{A, R\}, \{B, R\},$$

and there are no horizontal edges because $E = \emptyset$. Thus $G_{\text{hier}}(\mathbb{H}^{\langle 2\rangle})$ is a tree on 7 vertices with 6 edges, so $\mathbb{H}^{\langle 2\rangle}$ is a hierarchical tree.

For instance, the unique hierarchical path from a to d is

$$(a,\ A,\ R,\ B,\ d),$$

which moves upward through constituents and then downward to the target.

8.4 Hierarchical Cycle

A hierarchical cycle is a simple cycle in the hierarchical adjacency graph, combining vertical constituent links and horizontal co-membership edges.

Definition 8.4.1 (Hierarchical cycle)**.** Let $\mathbb{H}^{\langle r\rangle} = (V, E)$ be a hierarchical superhypergraph, and let $G_{\text{hier}}(\mathbb{H}^{\langle r\rangle}) = (V, E_{\text{hier}})$ be its hierarchical adjacency graph (Definition **??**).

A *hierarchical cycle* in $\mathbb{H}^{\langle r\rangle}$ is a (simple) cycle in the graph $G_{\text{hier}}(\mathbb{H}^{\langle r\rangle})$; that is, a vertex sequence

$$C = (v_0, v_1, \ldots, v_{k-1}, v_k) \qquad (k \geq 3),$$

such that:

(i) $v_0 = v_k$;

(ii) $v_0, v_1, \ldots, v_{k-1}$ are pairwise distinct;

(iii) $\{v_{i-1}, v_i\} \in E_{\text{hier}}$ for each $i = 1, \ldots, k$.

The integer k is the *length* of the hierarchical cycle.

Remark 8.4.2 (Interpretation)**.** A hierarchical cycle may mix *vertical* steps (immediate constituent links) and *horizontal* steps (co-membership in a hierarchical superhyperedge). In particular, hierarchical cycles capture feedback patterns that are invisible if one only looks at hyperedge co-membership (Berge cycles) or only at the inclusion hierarchy.

Example 8.4.3 (A minimal hierarchical cycle)**.** Let $V_0 := \{a, b, c\}$ and $r := 1$. Define the level-1 supervertices

$$A := \{a, b\}, \qquad B := \{b, c\}.$$

Let

$$V := \{a, b, c, A, B\} \subseteq \mathcal{U}_1(V_0), \qquad E := \{\varepsilon\}, \qquad \varepsilon := \{A, B\}.$$

Then (H3) holds because $A, B \in V$ implies $a, b, c \in V$. The immediate constituent relations are

$$b \prec A, \qquad b \prec B$$

(and also $a \prec A$, $c \prec B$, though they are not needed below). Moreover, since $\{A, B\} \subseteq \varepsilon \in E$, we have a horizontal adjacency $\{A, B\} \in E_{\text{horiz}}$.

Hence the hierarchical adjacency graph $G_{\text{hier}}(\mathbb{H}^{\langle 1 \rangle})$ contains the edges

$$\{A, b\}, \quad \{b, B\}, \quad \{A, B\}.$$

Therefore,

$$C = (A,\ b,\ B,\ A)$$

is a hierarchical cycle of length 3:

$$\{A, b\} \in E_{\text{vert}}, \qquad \{b, B\} \in E_{\text{vert}}, \qquad \{B, A\} \in E_{\text{horiz}}.$$

This cycle uses two vertical steps (constituent relations) and one horizontal step (a superhyperedge connection).

Chapter 9

Conclusion

In this book, we extended fundamental graph structures—including paths, trees, cycles, planarity, bipartiteness, and related notions—to the SuperHyperGraph setting, and we examined their key properties. We hope that future work will further explore these mathematical structures and advance algorithm design as well as quantitative analyses using computational experiments.

Disclaimer

Funding

This study did not receive any financial or external support from organizations or individuals.

Acknowledgments

We extend our sincere gratitude to everyone who provided insights, inspiration, and assistance throughout this research. We particularly thank our readers for their interest and acknowledge the authors of the cited works for laying the foundation that made our study possible. We also appreciate the support from individuals and institutions that provided the resources and infrastructure needed to produce and share this book. Finally, we are grateful to all those who supported us in various ways during this project.

Data Availability

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

Ethical Approval

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

Use of Generative AI and AI-Assisted Tools

I use generative AI and AI-assisted tools for tasks such as English grammar checking, and I do not employ them in any way that violates ethical standards.

Conflicts of Interest

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

Disclaimer

This work presents theoretical concepts that have not yet undergone practical testing or validation. Future researchers are encouraged to apply and assess these ideas in empirical contexts. While every effort has been made to ensure accuracy and appropriate referencing, unintentional errors or omissions may still exist. Readers are advised to verify referenced materials on their own. The views and conclusions expressed here are the authors' own and do not necessarily reflect those of their affiliated organizations.

Appendix A

Appendix: Iterated Multipartite graph

An iterated multipartite graph is a multipartite graph whose vertices are multipartite graphs, recursively layered over a finite base set.

Definition A.0.1 (Multipartite graph)**.** A *finite simple graph* is a pair $G = (V, E)$ where V is a finite nonempty set and $E \subseteq \binom{V}{2}$. Fix an integer $k \geq 2$. The graph G is called *k-partite* if there exist pairwise disjoint sets $V_1, \ldots, V_k$ such that

$$V \;=\; V_1 \,\dot{\cup}\, \cdots \,\dot{\cup}\, V_k \quad \text{and} \quad \forall\, \{u, v\} \in E,\; \exists\, i \neq j \text{ with } u \in V_i,\; v \in V_j.$$

A graph is called *multipartite* if it is k-partite for some $k \geq 2$.

Definition A.0.2 (The multipartite-graph operator)**.** For any (not necessarily numeric) set S, define

$$\mathcal{P}(S) \;:=\; \Big\{ G = (V, E) \;\Big|\; \emptyset \neq V \subseteq S \text{ is finite},\; E \subseteq \binom{V}{2},\; G \text{ is multipartite} \Big\}.$$

Thus $\mathcal{P}(S)$ is the class of all finite multipartite graphs whose vertices are elements of S.

Definition A.0.3 (Iterated multipartite graphs)**.** Fix a finite nonempty *base set* V_0 (of "atomic" objects). Define iteratively the hierarchy

$$\mathcal{P}^0(V_0) \;:=\; V_0, \qquad \mathcal{P}^{n+1}(V_0) \;:=\; \mathcal{P}\big(\mathcal{P}^n(V_0)\big) \qquad (n \in \mathbb{N}_0).$$

For $n \geq 1$, an *n-iterated multipartite graph on* V_0 is any element

$$G^{(n)} \in \mathcal{P}^n(V_0).$$

Equivalently, $G^{(1)}$ is an ordinary multipartite graph with vertex set $V(G^{(1)}) \subseteq V_0$, and for $n \geq 2$, $G^{(n)}$ is a multipartite graph whose vertices are $(n-1)$-iterated multipartite graphs:

$$V\big(G^{(n)}\big) \subseteq \mathcal{P}^{n-1}(V_0), \qquad E\big(G^{(n)}\big) \subseteq \binom{V(G^{(n)})}{2}.$$

Example A.0.4 (A 2-iterated multipartite graph)**.** Let the base set be

$$V_0 := \{a, b, c, d, e\}.$$

Define the first multipartite graph $G_1^{(1)}$ by the bipartition

$$V(G_1^{(1)}) = \{a, b, c\} = U_1 \,\dot\cup\, W_1, \qquad U_1 := \{a\}, \;\; W_1 := \{b, c\},$$

and the complete bipartite edge set between the parts:

$$E(G_1^{(1)}) = \big\{\{a, b\}, \{a, c\}\big\}.$$

Define the second multipartite graph $G_2^{(1)}$ by the tripartition

$$V(G_2^{(1)}) = \{c, d, e\} = U_2 \,\dot\cup\, W_2 \,\dot\cup\, Z_2, \qquad U_2 := \{c\},\; W_2 := \{d\},\; Z_2 := \{e\},$$

and let it be the complete tripartite graph on these parts:

$$E(G_2^{(1)}) = \big\{\{c, d\}, \{c, e\}, \{d, e\}\big\}.$$

Then $G_1^{(1)}$ and $G_2^{(1)}$ are elements of $\mathcal{P}^1(V_0)$.

Let

$$V\big(G^{(2)}\big) := \{G_1^{(1)},\, G_2^{(1)}\} \subseteq \mathcal{P}^1(V_0), \qquad E\big(G^{(2)}\big) := \big\{\{G_1^{(1)}, G_2^{(1)}\}\big\}.$$

Declare the bipartition of $V(G^{(2)})$ to be

$$V(G^{(2)}) = \{G_1^{(1)}\} \,\dot\cup\, \{G_2^{(1)}\}.$$

Hence $G^{(2)}$ is a (bi)partite graph on the vertex set $\{G_1^{(1)}, G_2^{(1)}\}$, where each vertex is itself an ordinary multipartite graph on V_0. Therefore,

$$G^{(2)} \in \mathcal{P}\big(\mathcal{P}^1(V_0)\big) = \mathcal{P}^2(V_0),$$

so $G^{(2)}$ is a 2-iterated multipartite graph on V_0 in the sense of Definition A.0.3.

Appendix (List of Tables)

*

Appendix (List of Figures)

*

Bibliography

[1] Takaaki Fujita and Florentin Smarandache. *HyperGraph and SuperHyperGraph Theory with Applications*. Neutrosophic Science International Association (NSIA) Publishing House, 2026.

[2] Takaaki Fujita and Florentin Smarandache. *HyperGraph and SuperHyperGraph Theory with Applications (II): Graph Property and Parameter*, volume II of *HyperGraph and SuperHyperGraph Theory with Applications*. Neutrosophic Science International Association (NSIA) Publishing House, 1.0 edition, 2026.

[3] Takaaki Fujita and Florentin Smarandache. *HyperGraph and SuperHyperGraph Theory with Applications (IV): Uncertain Graph Theory*, volume IV of *HyperGraph and SuperHyperGraph Theory with Applications*. Neutrosophic Science International Association (NSIA) Publishing House, 1.0 edition, 2026.

[4] Takaaki Fujita and Florentin Smarandache. *HyperGraph and SuperHyperGraph Theory with Applications (III): Intersection Graph and Graph Labeling*, volume III of *HyperGraph and SuperHyperGraph Theory with Applications*. Neutrosophic Science International Association (NSIA) Publishing House, 2026.

[5] Reinhard Diestel. *Graph theory*. Springer (print edition); Reinhard Diestel (eBooks), 2024.

[6] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, pages 3558–3565, 2019.

[7] 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.

[8] Yue Gao, Zizhao Zhang, Haojie Lin, Xibin Zhao, Shaoyi Du, and Changqing Zou. Hypergraph learning: Methods and practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2548–2566, 2020.

[9] 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.

[10] 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.

[11] Takaaki Fujita and Florentin Smarandache. A concise study of some superhypergraph classes. *Neutrosophic Sets and Systems*, 77:548–593, 2024.

[12] Takaaki Fujita and Florentin Smarandache. Fundamental computational problems and algorithms for superhypergraphs. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*, page 240, 2024.

[13] N. B. Nalawade, M. S. Bapat, S. G. Jakkewad, G. A. Dhanorkar, and D. J. Bhosale. Structural properties of zero-divisor hypergraph and superhypergraph over $\mathbb{Z}_n$: Girth and helly property. *Panamerican Mathematical Journal*, 35(4S):485, 2025.

[14] Mohammad Hamidi, Florentin Smarandache, and Elham Davneshvar. Spectrum of superhypergraphs via flows. *Journal of Mathematics*, 2022(1):9158912, 2022.

[15] Takaaki Fujita. Note for line and total superhypergraphs: Connecting vertices, edges, edges of edges, edges of edges of edges in hierarchical systems. *Journal of Mathematical Analysis and Modeling*, 6(3):11–43, 2025.

[16] Mohammad Hamidi and Mohadeseh Taghinezhad. *Application of Superhypergraphs-Based Domination Number in Real World*. Infinite Study, 2023.

[17] Takaaki Fujita and Florentin Smarandache. Topological generalizations of graphs: Integrating hypergraph and superhypergraph perspectives. *Turkish Journal of Mathematics and Computer Science*, 17(2):322–337, 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] Anandhan Prathik, K Uma, and J Anuradha. An overview of application of graph theory. *International Journal of ChemTech Research*, 9(2):242–248, 2016.

[20] Wai-Kai Chen. *Graph theory and its engineering applications*, volume 5. World Scientific Publishing Company, 1997.

[21] Jonathan L Gross, Jay Yellen, and Mark Anderson. *Graph theory and its applications*. Chapman and Hall/CRC, 2018.

[22] Takaaki Fujita and Florentin Smarandache. Directed n-superhypergraphs incorporating bipolar fuzzy information: A multi-tier framework for modeling bipolar uncertainty in complex networks. *Neutrosophic Sets and Systems*, 88:164–183, 2025.

[23] Salomón Marcos Berrocal Villegas, Willner Montalvo Fritas, Carmen Rosa Berrocal Villegas, María Yissel Flores Fuentes Rivera, Roberto Espejo Rivera, Laura Daysi Bautista Puma, and Dante Manuel Macazana Fernández. Using plithogenic n-superhypergraphs to assess the degree of relationship between information skills and digital competencies. *Neutrosophic Sets and Systems*, 84(1):41, 2025.

[24] Takaaki Fujita. Molecular fuzzy graphs, hypergraphs, and superhypergraphs. *Journal of Intelligent Decision and Computational Modelling*, 1(3):158–171, 2025.

[25] E. J. Mogro, J. R. Molina, G. J. S. Canas, and P. H. Soria. Tree tobacco extract (*Nicotiana glauca*) as a plithogenic bioinsecticide alternative for controlling fruit fly (*Drosophila immigrans*) using *n*-superhypergraphs. *Neutrosophic Sets and Systems*, 74:57–65, 2024.

[26] Takaaki Fujita. A unified graph-based framework for modeling legal citations: From Graphs to HyperGraphs and SuperHyperGraphs. *SciNexuses*, 2:132–143, 2025.

[27] Eduardo Martín Campoverde Valencia, Jessica Paola Chuisaca Vásquez, and Francisco Ángel Becerra Lois. Multineutrosophic analysis of the relationship between survival and business growth in the manufacturing sector of azuay province, 2020–2023, using plithogenic n-superhypergraphs. *Neutrosophic Sets and Systems*, 84(1):28, 2025.

[28] Alain Bretto. Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer*, 1, 2013.

[29] Junwu Chen and Philippe Schwaller. Molecular hypergraph neural networks. *The Journal of Chemical Physics*, 160(14), 2024.

[30] Shuyi Ji, Yifan Feng, Donglin Di, Shihui Ying, and Yue Gao. Mode hypergraph neural network. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[31] Nicholas Casetti, Pragnay Nevatia, Junwu Chen, Philippe Schwaller, and Connor W Coley. Comment on "molecular hypergraph neural networks"[j. chem. phys. 160, 144307 (2024)]. *The Journal of Chemical Physics*, 161(20), 2024.

[32] Wenjie Du, Shuai Zhang, Zhaohui Cai, Xuqiang Li, Zhiyuan Liu, Junfeng Fang, Jianmin Wang, Xiang Wang, and Yang Wang. Molecular merged hypergraph neural network for explainable solvation gibbs free energy prediction. *Research*, 8:0740, 2025.

[33] Julio Cesar Méndez Bravo, Claudia Jeaneth Bolanos Piedrahita, Manuel Alberto Méndez Bravo, and Luis Manuel Pilacuan-Bonete. Integrating smed and industry 4.0 to optimize processes with plithogenic n-superhypergraphs. *Neutrosophic Sets and Systems*, 84:328–340, 2025.

[34] Takaaki Fujita. Multi-superhypergraph neural networks: A generalization of multi-hypergraph neural networks. *Neutrosophic Computing and Machine Learning*, 39:328–347, 2025.

[35] Berrocal Villegas Salomón Marcos, Montalvo Fritas Willner, Berrocal Villegas Carmen Rosa, Flores Fuentes Rivera María Yissel, Espejo Rivera Roberto, Laura Daysi Bautista Puma, and Dante Manuel Macazana Fernández. Using plithogenic n-superhypergraphs to assess the degree of relationship between information skills and digital competencies. *Neutrosophic Sets and Systems*, 84:513–524, 2025.

[36] Nelly Hodelín Amable, Elizabeth Esther Vergel De Salazar, Martha Gloria Martínez Isaac, Olivia Catalina Olavarría Sánchez, and Johanna Mariuxi Solís Palma. Representation of motivational dynamics in school environments through plithogenic n-superhypergraphs with family participation. *Neutrosophic Sets and Systems*, 92:570–583, 2025.

[37] Ehab Roshdy, Marwa Khashaba, and Mariam Emad Ahmed Ali. Neutrosophic super-hypergraph fusion for proactive cyberattack countermeasures: A soft computing framework. *Neutrosophic Sets and Systems*, 94:232–252, 2025.

[38] Takaaki Fujita and Florentin Smarandache. Soft directed n-superhypergraphs with some real-world applications. *European Journal of Pure and Applied Mathematics*, 18(4):6643–6643, 2025.

[39] Takaaki Fujita. Directed acyclic superhypergraphs (dash): A general framework for hierarchical dependency modeling. *Neutrosophic Knowledge*, 6:72–86, 2025.

[40] Takaaki Fujita. Metahypergraphs, metasuperhypergraphs, and iterated metagraphs: Modeling graphs of graphs, hypergraphs of hypergraphs, superhypergraphs of superhypergraphs, and beyond. *Current Research in Interdisciplinary Studies*, 4(5):1–23, 2025.

[41] Thomas Jech. *Set theory: The third millennium edition, revised and expanded*. Springer, 2003.

[42] Claude Berge. *Hypergraphs: combinatorics of finite sets*, volume 45. Elsevier, 1984.

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

[44] Lin-Peng Zhang, Hajo Broersma, Ervin Győri, Casey Tompkins, and Ligong Wang. Connected tur\'{a} n numbers for berge paths in hypergraphs. *arXiv preprint arXiv:2409.03323*, 2024.

[45] Zhiyang He and Michael Tait. Hypergraphs with few berge paths of fixed length between vertices. *SIAM Journal on Discrete Mathematics*, 33(3):1472–1481, 2019.

[46] Chris Cornelis, Pieter M. M. De Kesel, and Etienne E. Kerre. Shortest paths in fuzzy weighted graphs. *International Journal of Intelligent Systems*, 19, 2004.

[47] Timothy M Chan and Dimitrios Skrepetos. All-pairs shortest paths in geometric intersection graphs. In *Workshop on Algorithms and Data Structures*, pages 253–264. Springer, 2017.

[48] Haitao Wang and Jie Xue. Near-optimal algorithms for shortest paths in weighted unit-disk graphs. *Discrete & Computational Geometry*, 64(4):1141–1166, 2020.

[49] Timothy M Chan and Dimitrios Skrepetos. Approximate shortest paths and distance oracles in weighted unit-disk graphs. *Journal of Computational Geometry*, 10(2):3–20, 2019.

[50] Michalis Potamias, Francesco Bonchi, Carlos Castillo, and Aristides Gionis. Fast shortest path distance estimation in large networks. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 867–876, 2009.

[51] David Karger, Rajeev Motwani, and Gurumurthy DS Ramkumar. On approximating the longest path in a graph. *Algorithmica*, 18(1):82–98, 1997.

[52] Kyriaki Ioannidou, George B Mertzios, and Stavros D Nikolopoulos. The longest path problem has a polynomial solution on interval graphs. *Algorithmica*, 61(2):320–341, 2011.

[53] Kyriaki Ioannidou and Stavros D Nikolopoulos. The longest path problem is polynomial on cocomparability graphs. *Algorithmica*, 65(1):177–205, 2013.

[54] Yoshihiro Takahara, Sachio Teramoto, and Ryuhei Uehara. Longest path problems on ptolemaic graphs. *IEICE transactions on information and systems*, 91(2):170–177, 2008.

[55] Takaaki Fujita. Modeling complex hierarchical systems with weighted and signed superhypergraphs: Foundations and applications. *Open Journal of Discrete Applied Mathematics (ODAM)*, 8(3):20–39, 2025.

[56] Gabriel Valiente. *Algorithms on trees and graphs*, volume 112. Springer, 2002.

[57] Jean-Pierre Serre. *Trees*. Springer Science & Business Media, 2002.

[58] Takaaki Fujita. Survey of trees, forests, and paths in fuzzy and neutrosophic graphs. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*, page 477, 2025.

[59] Isolde Adler, Georg Gottlob, and Martin Grohe. Hypertree width and related hypergraph invariants. *European Journal of Combinatorics*, 28(8):2167–2181, 2007.

[60] 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.

[61] Takaaki Fujita and Talal Ali Al-Hawary. Short note of superhyperclique-width and local superhypertree-width. *Neutrosophic Sets and Systems*, 86:811–837, 2025.

[62] Takaaki Fujita. Superhyperbranch-width and superhypertree-width. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*, page 367, 2025.

[63] 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, 2024.

[64] Calvin C. Elgot, Stephen L. Bloom, and Ralph Tindell. On the algebraic structure of rooted trees. *Journal of Computer and System Sciences*, 16:236–273, 1978.

[65] Karel Culík. Extensions of rooted trees and their applications. *Discret. Math.*, 18:131–148, 1977.

[66] Sumit Chakraborty, Maumita Chakraborty, and Rajat Kumar Pal. Generation of all rooted trees up to a given height. *Innovations in Systems and Software Engineering*, 20:467 – 475, 2022.

[67] Jan von Knop, Wolfgang R. Mueller, Zeljko Jericevic, and Nenad Trinajstic. Computer enumeration and generation of trees and rooted trees. *J. Chem. Inf. Comput. Sci.*, 21:91–99, 1981.

[68] Paul Laubie. Hypertrees and embedding of the fman operad. *International Mathematics Research Notices*, 2025(15):rnaf233, 2025.

[69] R Sundara Rajan, Paul Manuel, and Indra Rajasingh. Embeddings between hypercubes and hypertrees. *Journal of Graph Algorithms and Applications*, 19(1):361–373, 2015.

[70] Rajeev Raman, Venkatesh Raman, and Srinivasa Rao Satti. Succinct indexable dictionaries with applications to encoding k-ary trees, prefix sums and multisets. *ACM Transactions on Algorithms (TALG)*, 3(4):43–es, 2007.

[71] Anthony E Trojanowski. Ranking and listing algorithms for k-ary trees. *SIAM Journal on Computing*, 7(4):492–509, 1978.

[72] MC Er. Efficient generation of k-ary trees in natural order. *The Computer Journal*, 35(3):306–308, 1992.

[73] Limin Xiang, Kazuo Ushijima, and Selim G Akl. Generating regular k-ary trees efficiently. *The Computer Journal*, 43(4):290–300, 2000.

[74] Dominique Roelants van Baronaigien. A loopless gray-code algorithm for listing k-ary trees. *Journal of Algorithms*, 35(1):100–107, 2000.

[75] Murray Sherk. Self-adjusting k-ary search trees. *Journal of Algorithms*, 19(1):25–44, 1995.

[76] Mauricio A Valle, Gonzalo A Ruz, and Rodrigo Morrás. Market basket analysis: Complementing association rules with minimum spanning trees. *Expert Systems with Applications*, 97:146–162, 2018.

[77] Louigi Addario-Berry, Nicolas Broutin, Christina Goldschmidt, and Grégory Miermont. The scaling limit of the minimum spanning tree of the complete graph. *Annals of probability: An official journal of the Institute of Mathematical Statistics*, 45(5), 2017.

[78] Ronald L. Graham and Pavol Hell. On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 7:43–57, 1985.

[79] Hamed Ahmadi and José Ramon Martí. Minimum-loss network reconfiguration: A minimum spanning tree problem. *Sustainable Energy, Grids and Networks*, 1:1–9, 2015.

[80] Kartick Mohanta, Arindam Dey, Narayan C. Debnath, and Anita Pal. An algorithmic approach for finding minimum spanning tree in a intuitionistic fuzzy graph. In *Proceedings of 32nd International Conference*, 2019.

[81] Arindam Dey, Sahanur Mondal, and Tandra Pal. Robust and minimum spanning tree in fuzzy environment. *Int. J. Comput. Sci. Math.*, 10:513–524, 2019.

[82] Arindam Dey, Le Hoang Son, Anita Pal, and Hoang Viet Long. Fuzzy minimum spanning tree with interval type 2 fuzzy arc length: formulation and a new genetic algorithm. *Soft Computing*, 24:3963 – 3974, 2019.

[83] Ágnes Vathy-Fogarassy, Balazs Feil, and János Abonyi. Minimal spanning tree based fuzzy clustering. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 1:2499–2504, 2007.

[84] Jun Ye. Single-valued neutrosophic minimum spanning tree and its clustering method. *Journal of Intelligent Systems*, 23:311 – 324, 2014.

[85] Kalyan Adhikary, Prashnatita Pal, and Jayanta Poray. The minimum spanning tree problem on networks with neutrosophic numbers. *Neutrosophic Sets and Systems*, 63, 2024.

[86] Said Broumi, Assia Bakali, Mohamed Talea, Florentin Smarandache, and Rajkumar Verma. Computing minimum spanning tree in interval valued bipolar neutrosophic environment. *International Journal of Modeling and Optimization*, 7:300–304, 2017.

[87] J. Malarvizhi. The neutrosophic minimum spanning tree problem in connected undirected weighted complete strong neutrosophic graphs. *International Journal of Applied Mathematics*, 2025.

[88] Tobias Polzin and Siavash Vahdati Daneshmand. On steiner trees and minimum spanning trees in hypergraphs. *Operations Research Letters*, 31(1):12–20, 2003.

[89] Yang Ting, Sun Yugeng, Wang Zhaoxia, Zhang Juwei, and Ding Yingqiang. Study of the minimum spanning hyper-tree routing algorithm in wireless sensor networks. In *2007 IET Conference on Wireless, Mobile and Sensor Networks (CCWMSN07)*, pages 245–248. IET, 2007.

[90] Junqi Wang, Hailong Li, Gang Qu, Kim M Cecil, Jonathan R Dillman, Nehal A Parikh, and Lili He. Dynamic weighted hypergraph convolutional network for brain functional connectome analysis. *Medical image analysis*, 87:102828, 2023.

[91] Shreya Banerjee, Arghya Mukherjee, and Prasanta K Panigrahi. Quantum blockchain using weighted hypergraph states. *Physical Review Research*, 2(1):013322, 2020.

[92] Cezary Z Janikow. Fuzzy decision trees: issues and methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(1):1–14, 1998.

[93] Yufei Yuan and Michael J Shaw. Induction of fuzzy decision trees. *Fuzzy Sets and systems*, 69(2):125–139, 1995.

[94] Necati Olgun and Ahmed Hatip. The effect of the neutrosophic logic on the decision tree. *Quadruple Neutrosophic Theory and Applications, Pons Editions Brussels, Belgium, EU*, 17:238–253, 2020.

[95] Bo Pang. Neutrosophic information gain for decision tree construction: Application to teaching performance of tennis instructors in sports universities. *Neutrosophic Sets and Systems*, 91:126–135, 2025.

[96] Ross H Johnson and Paul R Winn. Utilization of graph theory and dynamic programming as substitutes for decision trees. *Journal of the Academy of Marketing Science*, 1(2):119–127, 1973.

[97] Izabela Kutschenreiter-Praszkiewicz. Decision rule induction based on the graph theory. In *Application of Decision Science in Business and Management*. IntechOpen, 2019.

[98] Elena Drabiková and Erika Fecková Škrabul'áková. Decision trees-a powerful tool in mathematical and economic modeling. In *2017 18th International Carpathian Control Conference (ICCC)*, pages 34–39. IEEE, 2017.

[99] Mohammad Hamidi and Marzieh Rahmati. On binary decision hypertree (hyperdiagram). *AUT Journal of Mathematics and Computing*, 5(2):117–130, 2024.

[100] R Sahu and AK Sharma. Understanding the unique properties of fuzzy concept in binary trees. *Indian Journal of Science and Technology*, 16(33):2631–2636, 2023.

[101] Hongze Qiu and Haitang Zhang. Fuzzy sliq decision tree based on classification sensitivity. *International Journal of Modern Education and Computer Science*, 3(5):18, 2011.

[102] Jia-Wei Hong and Arnold L Rosenberg. Graphs that are almost binary trees. *SIAM Journal on Computing*, 11(2):227–242, 1982.

[103] David Schaller, Manuela Geiß, Marc Hellmuth, and Peter F Stadler. Best match graphs with binary trees. In *International Conference on Algorithms for Computational Biology*, pages 82–93. Springer, 2021.

[104] Auparajita Krishnaa. Some algorithms of graph theory in cryptology. *Indian Journal of Advanced Mathematics, 4 (1), 9*, 15, 2024.

[105] Hong Jia-Wei and Arnold L Rosenberg. Graphs that are almost binary trees (preliminary version). In *Proceedings of the thirteenth annual ACM symposium on Theory of computing*, pages 334–341, 1981.

[106] Mohammad Hamidi and Marzieh Rahmati. On binary decision hypertree (hyperdiagram). *AUT Journal of Mathematics and Computing*, 5(2):117–130, 2024.

[107] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[108] Si-Qing Zheng. A simple and powerful representation of binary search trees. In *Great Lakes CS Conference on New Research Results in Computer Science*, pages 192–198. Springer, 1989.

[109] Victor N Kasyanov and Vladimir A Evstigneev. *Graph theory for programmers: algorithms for processing trees*, volume 515. Springer Science & Business Media, 2000.

[110] Ramin Kazemi and Sedigheh Zamani Mehreyan. Path length of protected nodes in random binary search trees. *Journal of Discrete Mathematics and Its Applications*, 10(4):321–332, 2025.

[111] Paweł Gawrychowski and Wojciech Janczewski. Simpler adjacency labeling for planar graphs with b-trees. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 24–36. SIAM, 2022.

[112] Douglas Comer. Ubiquitous b-tree. *ACM Computing Surveys (CSUR)*, 11(2):121–137, 1979.

[113] Georgios Theodorakis, James Clarkson, and Jim Webber. An empirical evaluation of variable-length record b+ trees on a modern graph database system. In *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*, pages 343–349. IEEE, 2024.

[114] Haim Kaplan and Robert Endre Tarjan. Thin heaps, thick heaps. *ACM Transactions on Algorithms (TALG)*, 4(1):1–14, 2008.

[115] Uday P Khedker, Amitabha Sanyal, and Amey Karkare. Heap reference analysis using access graphs. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 30(1):1–es, 2007.

[116] Rakesh Ghiya and Laurie J Hendren. Is it a tree, a dag, or a cyclic graph? a shape analysis for heap-directed pointers in c. In *Proceedings of the 23rd ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 1–15, 1996.

[117] Liqiao Xia, Pai Zheng, KL Keung, Chenyu Xiao, Tao Jing, and Liang Liu. From fault tree to fault graph: Bayesian network embedding-based fault isolation for complex equipment. *Manufacturing Letters*, 35:983–990, 2023.

[118] P Camarda, F Corsi, and A Trentadue. An efficient simple algorithm for fault tree automatic synthesis from the reliability graph. *IEEE transactions on Reliability*, 27(3):215–221, 2009.

[119] Sebastian Junges, Dennis Guck, Joost-Pieter Katoen, Arend Rensink, and Mariëlle Stoelinga. Fault trees on a diet: -automated reduction by graph rewriting-. In *International Symposium on Dependable Software Engineering: Theories, Tools, and Applications*, pages 3–18. Springer, 2015.

[120] Yasser A Mahmood, Alireza Ahmadi, Ajit Kumar Verma, Ajit Srividya, and Uday Kumar. Fuzzy fault tree analysis: a review of concept and application. *International Journal of System Assurance Engineering and Management*, 4(1):19–32, 2013.

[121] Refaul Ferdous, Faisal Khan, Brian Veitch, and Paul R Amyotte. Methodology for computer aided fuzzy fault tree analysis. *Process safety and environmental protection*, 87(4):217–226, 2009.

[122] Gin-Shuh Liang and Mao-Jiun J Wang. Fuzzy fault-tree analysis using failure possibility. *Microelectronics Reliability*, 33(4):583–597, 1993.

[123] 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.

[124] Hans L Bodlaender. Treewidth: Algorithmic techniques and results. In *International Symposium on Mathematical Foundations of Computer Science*, pages 19–36. Springer, 1997.

[125] Neil Robertson and Paul D Seymour. Graph minors. vii. disjoint paths on a surface. *Journal of Combinatorial Theory, Series B*, 45(2):212–254, 1988.

[126] Édouard Bonnet. Treewidth inapproximability and tight eth lower bound. *arXiv preprint arXiv:2406.11628*, 2024.

[127] Hans L Bodlaender, Alexander Grigoriev, and Arie MCA Koster. Treewidth lower bounds with brambles. In *Algorithms–ESA 2005: 13th Annual European Symposium, Palma de Mallorca, Spain, October 3-6, 2005. Proceedings 13*, pages 391–402. Springer, 2005.

[128] Hans L Bodlaender and Arie MCA Koster. Treewidth computations i. upper bounds. *Information and Computation*, 208(3):259–275, 2010.

[129] Takaaki Fujita. Short note of supertree-width and n-superhypertree-width. *Neutrosophic Sets and Systems*, 77:54–78, 2024.

[130] Neil Robertson and Paul D. Seymour. Graph minors. ii. algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986.

[131] Daniel J Harvey and David R Wood. Parameters tied to treewidth. *Journal of Graph Theory*, 84(4):364–385, 2017.

[132] Neil Robertson and Paul D Seymour. Graph minors. iv. tree-width and well-quasi-ordering. *Journal of Combinatorial Theory, Series B*, 48(2):227–254, 1990.

[133] Georg Gottlob, Gianluigi Greco, Nicola Leone, and Francesco Scarcello. Hypertree decompositions: Questions and answers. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 57–74, 2016.

[134] John N Mordeson and Premchand S Nair. Cycles and cocycles of fuzzy graphs. *Information Sciences*, 90(1-4):39–49, 1996.

[135] Sunil Mathew and MS Sunitha. Cycle connectivity in fuzzy graphs. *Journal of Intelligent & Fuzzy Systems*, 24(3):549–554, 2013.

[136] Florentin Smarandache, WB Kandasamy, and K Ilanthenral. *Neutrosophic graphs: A new dimension to graph theory*. EuropaNova ASBL, Brussels, Belgium, 2015.

[137] Dennis Clemens, Julia Ehrenmüller, and Yury Person. A dirac-type theorem for berge cycles in random hypergraphs. *arXiv preprint arXiv:1903.09057*, 2019.

[138] Deepak Bal, Ross Berkowitz, Pat Devlin, and Mathias Schacht. Hamiltonian berge cycles in random hypergraphs. *Combinatorics, Probability and Computing*, 30(2):228–238, 2021.

[139] Paul Dorbec, Sylvain Gravier, and Gábor N Sárközy. Monochromatic hamiltonian t-tight berge-cycles in hypergraphs. *Journal of Graph Theory*, 59(1):34–44, 2008.

[140] Christopher Umans and William Lenhart. Hamiltonian cycles in solid grid graphs. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 496–505. IEEE, 1997.

[141] Trevor I. Fenner and Alan M. Frieze. On the existence of hamiltonian cycles in a class of random graphs. *Discret. Math.*, 45:301–305, 1983.

[142] Olga Bodroža-Pantić, Harris Kwong, and Milan Pantić. Some new characterizations of hamiltonian cycles in triangular grid graphs. *Discrete Applied Mathematics*, 201:1–13, 2016.

[143] Sun-Yuan Hsieh, Gen-Huey Chen, and Chin-Wen Ho. Fault-free hamiltonian cycles in faulty arrangement graphs. *IEEE Transactions on Parallel and Distributed Systems*, 10(3):223–237, 1999.

[144] Sovan Samanta and Biswajit Sarkar. Generalized fuzzy euler graphs and generalized fuzzy hamiltonian graphs. *Journal of Intelligent & Fuzzy Systems*, 35(3):3413–3419, 2018.

[145] Rao Li. Sufficient conditions for hamiltonian fuzzy graphs. *Annals of Pure and Applied Mathematics*, 2022.

[146] Esra Cakr, Ziya Ulukan, and Tankut Acarman. Shortest fuzzy hamiltonian cycle on transportation network using minimum vertex degree and time-dependent dijkstra's algorithm. *IFAC-PapersOnLine*, 2021.

[147] A. Nagoor Gani and S. R. Latha. A new algorithm to find fuzzy hamilton cycle in a fuzzy network using adjacency matrix and minimum vertex degree. *SpringerPlus*, 5, 2016.

[148] Nagarajan Deivanayagam Pillai, Lathamaheswari Malayalan, Said Broumi, Florentin Smarandache, and Kavikumar Jacob. New algorithms for hamiltonian cycle under interval neutrosophic environment. In *Neutrosophic Graph Theory and Algorithms*, 2020.

[149] Malayalan Lathamaheswari, Said Broumi, and Florentin Smarandache. New algorithms for bipolar single-valued neutrosophic hamiltonian cycle. *Neutrosophic Operational Research*, 2021.

[150] Bhargav P. Narayanan and Mathias Schacht. Sharp thresholds for nonlinear hamiltonian cycles in hypergraphs. *Random Structures & Algorithms*, 57:244 – 255, 2019.

[151] Marek Karpinski, Andrzej Rucinski, and Edyta Szymanska. Computational complexity of the hamiltonian cycle problem in dense hypergraphs. In *Latin American Symposium on Theoretical Informatics*, 2010.

[152] Daniel Kuhn and Deryk Osthus. Hamilton cycles in graphs and hypergraphs: an extremal perspective. *arXiv: Combinatorics*, 2014.

[153] Qinghua Liu, Xiaojiao Chen, and Xiaoteng Tang. Spherical fuzzy bipartite graph based qfd methodology (sfbg-qfd): Assistive products design application. *Expert Syst. Appl.*, 239:122279, 2024.

[154] Armen S Asratian, Tristan MJ Denley, and Roland Häggkvist. *Bipartite graphs and their applications*, volume 131. Cambridge university press, 1998.

[155] Georgios A Pavlopoulos, Panagiota I Kontou, Athanasia Pavlopoulou, Costas Bouyioukos, Evripides Markou, and Pantelis G Bagos. Bipartite graphs in systems biology and medicine: a survey of methods and applications. *GigaScience*, 7(4):giy014, 2018.

[156] Sivasankar Shanmugam, Thirumal Perumal Aishwarya, and Nagesh Shreya. Bridge domination in fuzzy graphs. *Journal of fuzzy extension and applications*, 4(3):148–154, 2023.

[157] 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.

[158] M Kanchana and K Kavitha. Sensitivity analysis and application of single-valued neutrosophic transportaion problem. *Journal of King Saud University-Science*, 36(11):103567, 2024.

[159] Xueting Liao, Danyang Zheng, and Xiaojun Cao. Coronavirus pandemic analysis through tripartite graph clustering in online social networks. *Big Data Mining and Analytics*, 4(4):242–251, 2021.

[160] Xiaocan Li, Kun Xie, Xin Wang, Gaogang Xie, Kenli Li, Jiannong Cao, Dafang Zhang, and Jigang Wen. Tripartite graph aided tensor completion for sparse network measurement. *IEEE Transactions on Parallel and Distributed Systems*, 34(1):48–62, 2022.

[161] Kelong Mao, Xi Xiao, Jieming Zhu, Biao Lu, Ruiming Tang, and Xiuqiang He. Item tagging for information retrieval: A tripartite graph neural network based approach. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 2327–2336, 2020.

[162] Alain Hertz. Bipartable graphs. *J. Comb. Theory B*, 45:1–12, 1988.

[163] Midori Kobayashi, Keiko Kotani, Nobuaki Mutoh, and Gisaku Nakamura. Uniform coverings of 2-paths in the complete bipartite directed graph. In *AIP Conference Proceedings*. AIP Publishing LLC, 2015.

[164] Andrey Grinblat and Viktor Lopatkin. Bipartite graphs as polynomials and polynomials as bipartite graphs. *Journal of Algebra and Its Applications*, 20(05):2150083, 2021.

[165] Arthur H. Busch and Garth Isaak. Recognizing bipartite tolerance graphs in linear time. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, 2007.

[166] Arthur H. Busch. Recognizing bipartite unbounded tolerance graphs in linear time, 2007.

[167] Y.Daniel Liang and Norbert Blum. Circular convex bipartite graphs: Maximum matching and hamiltonian circuits. *Inf. Process. Lett.*, 56:215–219, 1995.

[168] Tian Liu, Min Lu, Zhao Lu, and Ke Xu. Circular convex bipartite graphs: Feedback vertex sets. *Theor. Comput. Sci.*, 556:55–62, 2014.

[169] Jing Huang. Representation characterizations of chordal bipartite graphs. *J. Comb. Theory B*, 96:673–683, 2006.

[170] Martin Charles Golumbic and Clinton F. Goss. Perfect elimination and chordal bipartite graphs. *J. Graph Theory*, 2:155–163, 1978.

[171] Subhabrata Paul, Dinabandhu Pradhan, and Shaily Verma. Vertex-edge domination in interval and bipartite permutation graphs. *Discussiones Mathematicae Graph Theory*, 0, 2021.

[172] Jan Derbisz. A polynomial kernel for vertex deletion into bipartite permutation graphs. *ArXiv*, abs/2111.14005, 2021.

[173] Ming Li, Siwei Zhou, Yuting Chen, Changqin Huang, and Yunliang Jiang. Educross: Dual adversarial bipartite hypergraph learning for cross-modal retrieval in multimodal educational slides. *Information Fusion*, 109:102428, 2024.

[174] Chidambaram Annamalai. Finding perfect matchings in bipartite hypergraphs. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1814–1823. SIAM, 2016.

[175] Hui Chen and Alan Frieze. Coloring bipartite hypergraphs. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 345–358. Springer, 1996.

[176] Kung-Jui Pai, Shyue-Ming Tang, Jou-Ming Chang, and Jinn-Shyong Yang. Completely independent spanning trees on complete graphs, complete bipartite graphs and complete tripartite graphs. In *Advances in Intelligent Systems and Applications-Volume 1: Proceedings of the International Computer Symposium ICS 2012 Held at Hualien, Taiwan, December 12–14, 2012*, pages 107–113. Springer, 2013.

[177] Zi-Ke Zhang, Tao Zhou, and Yi-Cheng Zhang. Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs. *Physica A: Statistical Mechanics and its Applications*, 389(1):179–186, 2010.

[178] Linhong Zhu, Aram Galstyan, James Cheng, and Kristina Lerman. Tripartite graph clustering for dynamic sentiment analysis on social media. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1531–1542, 2014.

[179] Ming-Sheng Shang, Zi-Ke Zhang, Tao Zhou, and Yi-Cheng Zhang. Collaborative filtering with diffusion-based similarity on tripartite graphs. *Physica A: Statistical Mechanics and its Applications*, 389(6):1259–1264, 2010.

[180] Mike J Grannell and Martin Knor. On the number of triangular embeddings of complete graphs and complete tripartite graphs. *Journal of Graph Theory*, 69(4):370–382, 2012.

[181] Peter A Bradshaw. Triangle packing on tripartite graphs is hard. *Rose-Hulman Undergraduate Mathematics Journal*, 20(1):7, 2019.

[182] Xin Liu and Tsuyoshi Murata. Detecting communities in tripartite hypergraphs. *arXiv preprint arXiv:1011.1043*, 2010.

[183] Saptarshi Ghosh, Pushkar Kane, and Niloy Ganguly. Identifying overlapping communities in folksonomies or tripartite hypergraphs. In *Proceedings of the 20th international conference companion on World wide web*, pages 39–40, 2011.

[184] Dmitry I Ignatov. On closure operators related to maximal tricliques in tripartite hypergraphs. *Discrete Applied Mathematics*, 249:74–84, 2018.

[185] Kalaichelvan Kalaiarasi, L. Mahalakshmi, Nasreen Kausar, Sajida Kousar, and Parameshwari Kattel. Perfect fuzzy soft tripartite graphs and their complements. *Discrete Dynamics in Nature and Society*, 2022.

[186] Sonia Alouane-Ksouri, Minyar Sassi Hidri, and Kamel Barkaoui. A tripartite graph-based fuzzy co-similarities for document retrieval. *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 575–580, 2014.

[187] S. Sathish and D. Vidhya. Different view on complete tripartite fuzzy graph in shift based company workers. *Advances in Mathematics: Scientific Journal*, 9(3):1407–1413, 2020.

[188] Serafino Cicerone and Gabriele Di Stefano. On the extension of bipartite to parity graphs. *Discret. Appl. Math.*, 95:181–195, 1999.

[189] P. Devi Priya and S. Monikandan. Reconstruction of 2-connected parity graphs. *Australasian Journal of Combinatorics*, 80(2), 2021.

[190] Serafino Cicerone and Gabriele Di Stefano. Graph classes between parity and distance-hereditary graphs. *Discret. Appl. Math.*, 95:197–216, 1999.

[191] Serafino Cicerone and Gabriele Di Stefano. On the equivalence in complexity among basic problems on bipartite and parity graphs. In *International Symposium on Algorithms and Computation*, 1997.

[192] Witold Lipski and Franco P. Preparata. Efficient algorithms for finding maximum matchings in convex bipartite graphs and related problems. *Acta Informatica*, 15:329–346, 1981.

[193] Fred W. Glover. Maximum matching in a convex bipartite graph. *Naval Research Logistics Quarterly*, 14:313–316, 1967.

[194] Yu Song, Tian Liu, and Ke Xu. Independent domination on tree convex bipartite graphs. In *FAW-AAIM*, 2012.

[195] Min-Sheng Lin and Chien Min Chen. Counting independent sets in tree convex bipartite graphs. *Discret. Appl. Math.*, 218:113–122, 2017.

[196] Gerandy Brito, Ioana Dumitriu, and Kameron Decker Harris. Spectral gap in random bipartite biregular graphs and applications. *Comb. Probab. Comput.*, 31:229–267, 2018.

[197] Yizhe Zhu. On the second eigenvalue of random bipartite biregular graphs. *Journal of Theoretical Probability*, 36:1269–1303, 2020.

[198] Snjezana Majstorović, Ivan Gutman, and Antoaneta Klobucar. Tricyclic biregular graphs whose energy exceeds the number of vertices. *Mathematical Communications*, 15:213–222, 2010.

[199] Bhagyashree Y. Bam, Charusheela M. Deshpande, and Lata N. Kamble. The existence of quasi regular and bi-regular self-complementary 3-uniform hypergraphs. *Discussiones Mathematicae Graph Theory*, 36:419 – 426, 2016.

[200] Marthe Bonamy, Matthew Johnson, Ioannis Lignos, Viresh Patel, and Daniël Paulusma. Reconfiguration graphs for vertex colourings of chordal and chordal bipartite graphs. *Journal of Combinatorial Optimization*, 27(1):132–143, 2014.

[201] Jing Huang. Representation characterizations of chordal bipartite graphs. *Journal of Combinatorial Theory, Series B*, 96(5):673–683, 2006.

[202] Martin Charles Golumbic and Clinton F Goss. Perfect elimination and chordal bipartite graphs. *Journal of Graph Theory*, 2(2):155–163, 1978.

[203] Mihály Bakonyi and Aaron Bono. Several results on chordal bipartite graphs. *Czechoslovak Mathematical Journal*, 47(4):577–583, 1997.

[204] Alan J Hoffman. On the line graph of the complete bipartite graph. *The Annals of Mathematical Statistics*, 35(2):883–885, 1964.

[205] Hikoe Enomoto, Tomoki Nakamigawa, and Katsuhiro Ota. On the pagenumber of complete bipartite graphs. *journal of combinatorial theory, Series B*, 71(1):111–120, 1997.

[206] Nora Hartsfield and WF Smyth. The sum number of complete bipartite graphs. In *Graphs, Matrices, and Designs*, pages 205–212. Routledge, 2017.

[207] Lowell W Beineke and Richard K Guy. The coarseness of the complete bipartite graph. *Canadian Journal of Mathematics*, 21:1086–1096, 1969.

[208] Jianfeng Hou and Shufei Wu. On bisections of graphs without complete bipartite graphs. *Journal of Graph Theory*, 98(4):630–641, 2021.

[209] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[210] 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.

[211] Witold Pedrycz. *Fuzzy control and fuzzy systems*. Research Studies Press Ltd., 1993.

[212] Muhammad Akram, Danish Saleem, and Talal Al-Hawary. Spherical fuzzy graphs with application to decision-making. *Mathematical and Computational Applications*, 25(1):8, 2020.

[213] Azriel Rosenfeld. Fuzzy graphs. In *Fuzzy sets and their applications to cognitive and decision processes*, pages 77–95. Elsevier, 1975.

[214] Zeeshan Saleem Mufti, Ali Tabraiz, Muhammad Farhan Hanif, et al. Molecular insights into tetracene through fuzzy topological indices in chemical graph theory. *Chemical Papers*, 79(5):2937–2953, 2025.

[215] Hajime Nobuhara and K. Hirota. A solution for eigen fuzzy sets of adjoint max-min composition and its application to image analysis. *IEEE International Symposium on Intelligent Signal Processing, 2003*, pages 27–30, 2003.

[216] TM Nishad, Talal Ali Al-Hawary, and B Mohamed Harif. General fuzzy graphs. *Ratio Mathematica*, 47, 2023.

[217] Talal Al-Hawary. Complete fuzzy graphs. *International Journal of Mathematical Combinatorics*, 4:26, 2011.

[218] Muhammad Akram and Noura Omair Alshehri. Tempered interval-valued fuzzy hypergraphs. *University politehnica of bucharest scientific bulletin-series a-applied mathematics and physics*, 77(1):39–48, 2015.

[219] John N Mordeson and Premchand S Nair. *Fuzzy graphs and fuzzy hypergraphs*, volume 46. Physica, 2012.

[220] Florentin Smarandache. *A unifying field in logics: neutrosophic logic. Neutrosophy, neutrosophic set, neutrosophic probability: neutrsophic logic. Neutrosophy, neutrosophic set, neutrosophic probability*. Infinite Study, 2005.

[221] Florentin Smarandache. Neutrosophy: neutrosophic probability, set, and logic: analytic synthesis & synthetic analysis. 1998.

[222] Florentin Smarandache. *Neutrosophic Overset, Neutrosophic Underset, and Neutrosophic Offset. Similarly for Neutrosophic Over-/Under-/Off-Logic, Probability, and Statistics*. Infinite Study, 2016.

[223] Haibin Wang, Florentin Smarandache, Yanqing Zhang, and Rajshekhar Sunderraman. *Single valued neutrosophic sets*. Infinite study, 2010.

[224] Takaaki Fujita and Florentin Smarandache. *Neutrosophic soft n-super-hypergraphs with real-world applications*. Infinite Study, 2025.

[225] Juanjuan Ding, Wenhui Bai, and Chao Zhang. A new multi-attribute decision making method with single-valued neutrosophic graphs. *International Journal of Neutrosophic Science*, 2021.

[226] M Hamidi and A Borumand Saeid. Accessible single-valued neutrosophic graphs. *Journal of Applied Mathematics and Computing*, 57:121–146, 2018.

[227] S Satham Hussain, N Durga, Rahmonlou Hossein, and Ghorai Ganesh. New concepts on quadripartitioned single-valued neutrosophic graph with real-life application. *International Journal of Fuzzy Systems*, 24(3):1515–1529, 2022.

[228] Muhammad Akram and Hafiza Saba Nawaz. Implementation of single-valued neutrosophic soft hypergraphs on human nervous system. *Artificial Intelligence Review*, 56(2):1387–1425, 2023.

[229] Muhammad Akram and Hafiza Saba Nawaz. Algorithms for the computation of regular single-valued neutrosophic soft hypergraphs applied to supranational asian bodies. *Journal of Applied Mathematics and Computing*, 68(6):4479–4506, 2022.

[230] Shouxian Zhu. Neutrosophic n-superhypernetwork: A new approach for evaluating short video communication effectiveness in media convergence. *Neutrosophic Sets and Systems*, 85:1004–1017, 2025.

[231] Said Broumi, Mohamed Talea, Assia Bakali, and Florentin Smarandache. Single valued neutrosophic graphs. *Journal of New theory*, 10:86–101, 2016.

[232] Muhammad Akram, Sundas Shahzadi, and AB Saeid. Single-valued neutrosophic hypergraphs. *TWMS Journal of Applied and Engineering Mathematics*, 8(1):122–135, 2018.

[233] Muhammad Akram and Anam Luqman. Certain networks models using single-valued neutrosophic directed hypergraphs. *Journal of Intelligent & Fuzzy Systems*, 33(1):575–588, 2017.

[234] Muhammad Akram and Anam Luqman. Intuitionistic single-valued neutrosophic hypergraphs. *Opsearch*, 54:799–815, 2017.

[235] Takaaki Fujita and Florentin Smarandache. A unified framework for u-structures and functorial structure: Managing super, hyper, superhyper, tree, and forest uncertain over/under/off models. *Neutrosophic Sets and Systems*, 91:337–380, 2025.

[236] Friedrich Esser and Frank Harary. On the spectrum of a complete multipartite graph. *European Journal of Combinatorics*, 1(3):211–218, 1980.

[237] Kazuhiko Ushio, Shinsei Tazawa, and Sumiyasu Yamamoto. On claw-decomposition of a complete multipartite graph. *Hiroshima Mathematical Journal*, 8(1):207–210, 1978.

[238] Akshay Vashist, Casimir A Kulikowski, and Ilya Muchnik. Ortholog clustering on a multipartite graph. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(1):17–27, 2007.

[239] Allan Lo, Andrew Treglown, and Yi Zhao. Complete subgraphs in a multipartite graph. *Combinatorics, Probability and Computing*, 31(6):1092–1101, 2022.

[240] Fang Tian. On the number of linear multipartite hypergraphs with given size. *Graphs and Combinatorics*, 37(6):2487–2496, 2021.

[241] Artchariya Muaengwaeng. Defective colorings on complete bipartite and multipartite k-uniform hypergraphs. *hulalongkorn University Theses and Dissertations (Chula ETD)*, 2018.

[242] KK Myithili and R Keerthika. Transversal core of intuitionistic fuzzy k-partite hypergraphs. *Ratio Mathematica*, 46, 2023.

[243] KK Myithili and R Keerthika. Isomorphic properties on intuitionistic fuzzy k-partite hypergraphs. *Advances and Applications in Mathematical Sciences*, 21(7):3653–3672, 2022.

[244] Jie Han and Yi Zhao. Tur\'an number of complete multipartite graphs in multipartite graphs. *arXiv preprint arXiv:2405.16561*, 2024.

[245] Shinsei Tazawa, Kazuhiko Ushio, and Sumiyasu Yamamoto. Partite-claw-decomposition of a complete multipartite graph. *Hiroshima Mathematical Journal*, 8(1):195–206, 1978.

[246] Khee Meng Koh and BP Tan. The diameter of an orientation of a complete multipartite graph. *Discrete Mathematics*, 149(1-3):131–139, 1996.

[247] Ligong Wang and Xiaodong Liu. Integral complete multipartite graphs. *Discrete mathematics*, 308(17):3860–3870, 2008.

[248] Brian Jacobson, Andrew Niedermaier, and Victor Reiner. Critical groups for complete multipartite graphs and cartesian products of complete graphs. *Journal of Graph Theory*, 44(3):231–250, 2003.

[249] Pierre Aboulker, Guillaume Aubian, and Pierre Charbit. Heroes in oriented complete multipartite graphs. *Journal of Graph Theory*, 105(4):652–669, 2024.

[250] Ya-Lei Jin and Xiao-Dong Zhang. Complete multipartite graphs are determined by their distance spectra. *Linear Algebra and its Applications*, 448:285–291, 2014.

[251] Daniel Gonçalves, Lucas Isenmann, and Claire Pennarun. Planar graphs as l-intersection or l-contact graphs. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 172–184. SIAM, 2018.

[252] Jan Kratochvíl and Aleš Kuběna. On intersection representations of co-planar graphs. *Discrete Mathematics*, 178(1-3):251–255, 1998.

[253] Steven Chaplick and Torsten Ueckerdt. Planar graphs as vpg-graphs. In *Graph Drawing: 20th International Symposium, GD 2012, Redmond, WA, USA, September 19-21, 2012, Revised Selected Papers 20*, pages 174–186. Springer, 2013.

[254] Eyal Ackerman and Gábor Tardos. On the maximum number of edges in quasi-planar graphs. *Journal of Combinatorial Theory, Series A*, 114(3):563–571, 2007.

[255] Jacob Fox, János Pach, and Andrew Suk. Quasiplanar graphs, string graphs, and the erdős–gallai problem. *European Journal of Combinatorics*, 119:103811, 2024.

[256] Eyal Ackerman. Quasi-planar graphs. In *Beyond Planar Graphs: Communications of NII Shonan Meetings*, pages 31–45. Springer, 2020.

[257] Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *Journal of the ACM (JACM)*, 51(6):993–1024, 2004.

[258] Gregory Gutin, Ton Kloks, Chuan Min Lee, and Anders Yeo. Kernels in planar digraphs. *Journal of Computer and System Sciences*, 71(2):174–184, 2005.

[259] Jan Kratochvíl and Alev Antonín Kubna. On intersection representations of co-planar graphs. *Discret. Math.*, 178:251–255, 1998.

[260] Piotr Micek, Marcin Kozik, and Iwona Cieslik. On-line coloring of i s -free graphs and co-planar graphs. In *Proceedings of the Annual Discrete Mathematics and Theoretical Computer Science AF*, 2006.

[261] Ganesh Ghorai and Madhumangal Pal. Faces and dual of m-polar fuzzy planar graphs. *J. Intell. Fuzzy Syst.*, 31:2043–2049, 2016.

[262] Muhammad Akram, Ayesha Bashir, and Sovan Samanta. Complex pythagorean fuzzy planar graphs. *International Journal of Applied and Computational Mathematics*, 6:1–27, 2020.

[263] Lowell W Beineke. Biplanar graphs:: A survey. *Computers & Mathematics with Applications*, 34(11):1–8, 1997.

[264] Alfredo García, Ferran Hurtado, Matias Korman, Inês Matos, Maria Saumell, Rodrigo I Silveira, Javier Tejel, and Csaba D Tóth. Geometric biplane graphs i: Maximal graphs. *Graphs and Combinatorics*, 31(2):407–425, 2015.

[265] P Chellamani, D Ajay, Mohammed M Al-Shamiri, and Rashad Ismail. *Pythagorean Neutrosophic Planar Graphs with an Application in Decision-Making*. Infinite Study, 2023.

[266] Rupkumar Mahapatra, Sovan Samanta, and Madhumangal Pal. Generalized neutrosophic planar graphs and its application. *Journal of Applied Mathematics and Computing*, 65(1):693–712, 2021.

[267] Takaaki Fujita and Florentin Smarandache. *Survey of planar and outerplanar graphs in fuzzy and neutrosophic graphs*. Infinite Study, 2025.

[268] JP Thempaavai et al. Inverse neutrosophic planar graphs and its application in optimizing the expenditure in inter-cropping. *Neutrosophic Sets and Systems*, 85:588–609, 2025.

[269] Hubert De Fraysseix, Patrice Ossona de Mendez, and Pierre Rosenstiehl. Representation of planar hypergraphs by contacts of triangles. In *International Symposium on Graph Drawing*, pages 125–136. Springer, 2007.

[270] Zhi Liu, Ardashir Mohammadzadeh, Hamza Turabieh, Majdi M. Mafarja, Shahab S. Band, and Amir H. Mosavi. A new online learned interval type-3 fuzzy control system for solar energy management systems. *IEEE Access*, 9:10498–10508, 2021.

[271] Thomas Dissaux and Nicolas Nisse. Pathlength of outerplanar graphs. In *Latin American Symposium on Theoretical Informatics*, pages 172–187. Springer, 2022.

[272] Deivanai Jaisankar, Sujatha Ramalingam, Deivanayagampillai Nagarajan, and Tadesse Walelign. Fuzzy outerplanar graphs and its applications. *Int. J. Comput. Intell. Syst.*, 17:231, 2024.

[273] Deivanai Jaisankar, Sujatha Ramalingam, Nagarajan Deivanayagampillai, and Tadesse Walelign. Network design for bypass roads using interval valued fuzzy outerplanar graphs. *Scientific Reports*, 15(1):1–27, 2025.

[274] Deivanai Jaisankar and Sujatha Ramalingam. A multipolar fuzzy outerplanar graph approach to one-way road network construction. *Journal of Computational Analysis & Applications*, 34(8), 2025.

[275] Deivanai Jaisankar, Sujatha Ramalingam, and Gizachew Bayou Zegeye. A fuzzy graph theoretic approach to face shape recognition using cubic outerplanar structures. *Scientific Reports*, 2025.

[276] Deivanai Jaisankar, Sujatha Ramalingam, Nagarajan Deivanayagampillai, and Tadesse Walelign. Bipolar fuzzy outerplanar graphs approach in image shrinking. *Scientific Reports*, 15(1):24587, 2025.

[277] Douglas B West and Jennifer I Wise. Bar visibility numbers for hypercubes and outerplanar digraphs. *Graphs and Combinatorics*, 33(1):221–231, 2017.

[278] Chun-Hsiang Chan and Hsu-Chun Yen. On contact representations of directed planar graphs. In *Computing and Combinatorics: 24th International Conference, COCOON 2018, Qing Dao, China, July 2-4, 2018, Proceedings 24*, pages 218–229. Springer, 2018.

[279] Hristo N Djidjev, Grammati E Pantziou, and Christos D Zaroliagis. Computing shortest paths and distances in planar graphs. In *International Colloquium on Automata, Languages, and Programming*, pages 327–338. Springer, 1991.

[280] Muhammad Anwarul Azim, Sk Ruhul Azgor, Sadia Sharmin, and Md. Saidur Rahman. On the twin-width of outerplanar graphs. In *International Conference on Combinatorial Optimization and Applications*, 2024.

[281] Joshua Geis and Johannes Zink. From planar via outerplanar to outerpath - engineering np-hardness constructions (poster abstract). In *International Symposium Graph Drawing and Network Visualization*, 2024.

[282] Mark N Ellingham, Linyuan Lu, and Zhiyu Wang. Maximum spectral radius of outerplanar 3-uniform hypergraphs. *Journal of Graph Theory*, 100(4):671–685, 2022.

[283] René van Bevern, Iyad A Kanj, Christian Komusiewicz, Rolf Niedermeier, and Manuel Sorge. The role of twins in computing planar supports of hypergraphs. *arXiv preprint arXiv:1511.09389*, 2015.

[284] Samir Datta, Raghav Kulkarni, and Sambuddha Roy. Deterministically isolating a perfect matching in bipartite planar graphs. *Theory of Computing Systems*, 47(3):737–757, 2010.

[285] Yuanqiu Huang, Zhangdong Ouyang, and Fengming Dong. On the sizes of bipartite 1-planar graphs. *arXiv preprint arXiv:2007.13308*, 2020.

[286] Derek A Holton, Bennet Manvel, and Brendan D McKay. Hamiltonian cycles in cubic 3-connected bipartite planar graphs. *J. Comb. Theory, Ser. B*, 38(3):279–297, 1985.

[287] Gary Gordon. Workable gears, archimedian solids and planar bipartite graphs. *The American Mathematical Monthly*, 101(6):527–534, 1994.

[288] Chenglong Deng and Xuding Zhu. Truncated degree at-orientations of outerplanar graphs. *arXiv preprint arXiv:2412.20811*, 2024.

[289] Boštjan Brešar, Nicolas Gastineau, and Olivier Togni. Packing colorings of subcubic outerplanar graphs. *Aequationes mathematicae*, 94(5):945–967, 2020.

[290] Ying Wang, Yiqiao Wang, Weifan Wang, and Shuyu Cui. Strong chromatic index of outerplanar graphs. *Axioms*, 11(4):168, 2022.

[291] Bojan Mohar. Face covers and the genus problem for apex graphs. *Journal of Combinatorial Theory, Series B*, 82(1):102–117, 2001.

[292] Jagdeep Singh, Vaidy Sivaraman, and Thomas Zaslavsky. Apex graphs and cographs. *arXiv preprint arXiv:2310.02551*, 2023.

[293] Dibyayan Chakraborty and Kshitij Gajjar. Finding geometric representations of apex graphs is np-hard. In *International Conference and Workshops on Algorithms and Computation*, pages 161–174. Springer, 2022.

[294] Patrizio Angelini, Michael A Bekos, Franz J Brandenburg, Giordano Da Lozzo, Giuseppe Di Battista, Walter Didimo, Giuseppe Liotta, Fabrizio Montecchiani, and Ignaz Rutter. On the relationship between k-planar and k-quasi-planar graphs. In *Graph-Theoretic Concepts in Computer Science: 43rd International Workshop, WG 2017, Eindhoven, The Netherlands, June 21-23, 2017, Revised Selected Papers 43*, pages 59–74. Springer, 2017.

[295] Michael A Bekos, Prosenjit Bose, Aaron Büngener, Vida Dujmović, Michael Hoffmann, Michael Kaufmann, Pat Morin, Saeed Odak, and Alexandra Weinberger. On k-planar graphs without short cycles. *arXiv preprint arXiv:2408.16085*, 2024.

[296] Todor Antić. Convex-geometric k-planar graphs are convex-geometric (k+ 1)-quasiplanar. In *International Workshop on Combinatorial Algorithms*, pages 138–150. Springer, 2024.

[297] John H Halton. On the thickness of graphs of given degree. *Information Sciences*, 54(3):219–238, 1991.

[298] Lowell W Beineke and Frank Harary. The thickness of the complete graph. *Canadian Journal of Mathematics*, 17:850–859, 1965.

[299] Vida Dujmovic and David R Wood. Graph treewidth and geometric thickness parameters. *Discrete & Computational Geometry*, 37(4):641–670, 2007.

[300] Ganesh Ghorai and Madhumangal Pal. A study on m-polar fuzzy planar graphs. *Int. J. Comput. Sci. Math.*, 7:283–292, 2016.

[301] Waheed Ahmad Khan, Arsh E Mah Niaz, Trung Tuan Nguyen, Minh-Hoan Pham, Thi Minh Ngoc Tong, and Hai V. Pham. A fuzzy soft planar graph with application in image segmentation. *Scientific Reports*, 15, 2025.

[302] Hao Guan, Saira Hameed, Sadaf, Aysha Khan, and Jana Shafi. A spherical fuzzy planar graph approach to optimize wire configuration in transformers. *Frontiers in Physics*, 2025.

[303] Rahul Mondal and Ganesh Ghorai. Inverse fuzzy mixed planar graphs with application. *International Journal of Applied and Computational Mathematics*, 10, 2024.

[304] Noura Alshehri and Muhammad Akram. Intuitionistic fuzzy planar graphs. *Discrete Dynamics in Nature and Society*, 2014(1):397823, 2014.

Hypergraphs generalize ordinary graphs by allowing an edge to connect an arbitrary nonempty subset of the vertex set. By iterating the powerset construction, one obtains nested, higher-order vertex objects and arrives at finite SuperHyperGraphs, in which vertices are set-valued across multiple layers and edges encode relations among such set-valued vertices. Despite this expressive modeling capacity, the structural theory of SuperHyperGraphs—including systematic studies of their basic properties and parameters—remains comparatively underdeveloped. In this volume, we extend fundamental graph-theoretic notions to the SuperHyperGraph setting, focusing in particular on paths, trees, cycles, planarity, bipartiteness, and closely related concepts, and we investigate their key properties.

ISBN 978-1-59973-856-7

9 781599 738567 >