

Article

Not peer-reviewed version

Graph in Tree, Cycle in Cycle, Spiral Graph, and More

[Takaaki Fujita](#)*

Posted Date: 29 December 2025

doi: 10.20944/preprints202512.2458.v1

Keywords: MetaGraph; iterated MetaGraph; graph in tree; cycle in cycle; spiral graph



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Graph in Tree, Cycle in Cycle, Spiral Graph, and More

Takaaki Fujita

Independent Researcher, Tokyo, Japan; takaaki.fujita060@gmail.com

Abstract

A *MetaGraph* is a higher-level graph whose vertices are themselves graphs, with edges representing specified relations among these graphs. An *Iterated MetaGraph* extends this notion recursively: its vertices are *MetaGraphs*, thereby forming a hierarchical structure of graphs-of-graphs across multiple levels. In this paper, we introduce and study specific classes of *MetaGraphs*, including *Graph-in-Tree* and *Cycle-in-Cycle* structures. In addition, we define a new class of graphs, called *Spiral Graphs*, and investigate their fundamental properties.

Keywords: MetaGraph; iterated MetaGraph; graph in tree; cycle in cycle; spiral graph

1. Preliminaries

This section collects the basic terminology and definitions used throughout the paper. Unless stated otherwise, every graph appearing here is assumed to be *finite*.

1.1. MetaGraph (Graph of Graphs)

Classical graph theory studies combinatorial structures made of vertices and edges and uses them to represent relations and connectivity [1,2]. In this work, a *MetaGraph* is a graph whose vertices are themselves graphs; edges encode prescribed relations between these vertex-graphs (cf. [3–5]). Recent years have also seen the introduction of broader extensions of *MetaGraphs*, such as *Metahypergraphs* and *Metasuperhypergraphs* (cf. [6–8]).

Definition 1 (Metagraph (graph of graphs)). (cf. [6,9]) Fix a nonempty universe $\mathfrak{G}$ of finite graphs (undirected and loopless by default) and a nonempty family of binary relations

$$\mathcal{R} \subseteq \mathcal{P}(\mathfrak{G} \times \mathfrak{G}).$$

A metagraph over $(\mathfrak{G}, \mathcal{R})$ is a directed, labelled multigraph

$$M = (V, E, s, t, \lambda)$$

such that

$$V \subseteq \mathfrak{G}, \quad s, t : E \rightarrow V, \quad \lambda : E \rightarrow \mathcal{R},$$

and the following incidence condition holds:

$$\forall e \in E : (s(e), t(e)) \in \lambda(e).$$

The elements of V are called **meta-vertices** (each meta-vertex is a graph $G \in \mathfrak{G}$). For an edge $e \in E$ with $\lambda(e) = R$, we write $s(e) \xrightarrow{R} t(e)$ and refer to e as a **meta-edge**. If $\mathcal{R} = \{R\}$ is a singleton, we may omit edge labels. When every relation in $\mathcal{R}$ is symmetric, M may be regarded as an undirected labelled multigraph.

Example 1 (A concrete metagraph (graph of graphs)). Let $\mathfrak{G}$ be the class of all finite simple graphs, and let $\mathcal{R} = \{R_{\text{sub}}\}$ where

$$(G, H) \in R_{\text{sub}} \iff G \text{ is isomorphic to a (not necessarily induced) subgraph of } H.$$

Define three graphs

$$G_1 := P_2 \text{ (a single edge), } \quad G_2 := P_3 \text{ (a path on 3 vertices), } \quad G_3 := C_3 \text{ (a triangle).}$$

Then G_1 is a subgraph of G_2 , and G_2 is a subgraph of G_3 (by deleting one edge of the triangle), hence

$$(G_1, G_2) \in R_{\text{sub}}, \quad (G_2, G_3) \in R_{\text{sub}}.$$

Consider the directed metagraph

$$M = (V, E, s, t, \lambda)$$

with

$$V = \{G_1, G_2, G_3\}, \quad E = \{e_{12}, e_{23}\},$$

and structure maps

$$s(e_{12}) = G_1, \quad t(e_{12}) = G_2, \quad \lambda(e_{12}) = R_{\text{sub}}, \quad s(e_{23}) = G_2, \quad t(e_{23}) = G_3, \quad \lambda(e_{23}) = R_{\text{sub}}.$$

Then M is a metagraph over $(\mathfrak{G}, \mathcal{R})$ since each meta-edge satisfies the incidence condition:

$$(s(e_{12}), t(e_{12})) = (G_1, G_2) \in R_{\text{sub}}, \quad (s(e_{23}), t(e_{23})) = (G_2, G_3) \in R_{\text{sub}}.$$

1.2. Iterated MetaGraph (Graph of Graphs of $\dots$ of Graphs)

An *Iterated MetaGraph* is obtained by repeating the MetaGraph construction: its vertices are metagraphs, and this recursion produces a hierarchy of graph-of-graphs structures across multiple levels [6].

Definition 2 (Unit metagraph embedding). [6] For $X \in \mathfrak{G}$ define the unit metagraph

$$U(X) := (\{X\}, \emptyset, _ _ _).$$

This yields an injective map

$$U : \mathfrak{G} \hookrightarrow \text{Obj}(\text{Meta}(\mathfrak{G}, \mathcal{R})).$$

Definition 3 (Relation lifting). Let $\mathcal{R}$ be a family of relations on $\mathfrak{G}$. Its lift $\mathcal{R}^\uparrow$ to finite metagraphs over $(\mathfrak{G}, \mathcal{R})$ is defined by

$$\forall R \in \mathcal{R}, \quad (M_1, M_2) \in \mathcal{R}^\uparrow \iff \exists x \in V(M_1), y \in V(M_2) \text{ such that } (x, y) \in R.$$

Set $\mathcal{R}^\uparrow := \{R^\uparrow : R \in \mathcal{R}\}$.

Definition 4 (Iterated object and relation universes). Define recursively for $t \in \mathbb{N}_0$:

$$\begin{aligned} \mathfrak{G}^{(0)} &:= \mathfrak{G}, & \mathcal{R}^{(0)} &:= \mathcal{R}, \\ \mathfrak{G}^{(t+1)} &:= \left\{ \text{finite metagraphs over } (\mathfrak{G}^{(t)}, \mathcal{R}^{(t)}) \right\}, & \mathcal{R}^{(t+1)} &:= (\mathcal{R}^{(t)})^\uparrow. \end{aligned}$$

Definition 5 (Iterated MetaGraph of depth t). For $t \in \mathbb{N}_0$, an iterated metagraph of depth t is a metagraph

$$M^{(t)} = (V^{(t)}, E^{(t)}, s^{(t)}, t^{(t)}, \lambda^{(t)})$$

over $(\mathfrak{G}^{(t)}, \mathcal{R}^{(t)})$; that is, $V^{(t)} \subseteq \mathfrak{G}^{(t)}$, $\lambda^{(t)} : E^{(t)} \rightarrow \mathcal{R}^{(t)}$, and

$$\forall e \in E^{(t)} : (s^{(t)}(e), t^{(t)}(e)) \in \lambda^{(t)}(e).$$

Example 2 (A concrete iterated metagraph of depth 2). Let $\mathfrak{G}$ be the class of all finite simple graphs and let $\mathcal{R} = \{R\}$ where

$$(G, H) \in R \iff |V(G)| \leq |V(H)|.$$

Thus a meta-edge $G \xrightarrow{R} H$ is permitted whenever the source graph has at most as many vertices as the target graph.

Step 0 (base graphs). Let

$$A := P_2, \quad B := P_3,$$

so $|V(A)| = 2$ and $|V(B)| = 3$.

Step 1 (metagraphs over $(\mathfrak{G}, \mathcal{R})$). Define two metagraphs $M_1, M_2 \in \mathfrak{G}^{(1)}$ as follows:

$$M_1 := (V_1, E_1, s_1, t_1, \lambda_1), \quad V_1 = \{A, B\}, \quad E_1 = \{e\},$$

with

$$s_1(e) = A, \quad t_1(e) = B, \quad \lambda_1(e) = R.$$

Also let

$$M_2 := (V_2, E_2, s_2, t_2, \lambda_2), \quad V_2 = \{B\}, \quad E_2 = \emptyset.$$

Hence M_1 has a single meta-edge $A \xrightarrow{R} B$, while M_2 consists of one meta-vertex and no meta-edges.

Step 2 (an iterated metagraph over $(\mathfrak{G}^{(1)}, \mathcal{R}^{(1)})$). Recall that $\mathcal{R}^{(1)} = \{R^\uparrow\}$ is the lifted relation on depth-1 metagraphs, where

$$(M, N) \in R^\uparrow \iff \exists x \in V(M), \exists y \in V(N) : (x, y) \in R.$$

We claim $(M_1, M_2) \in R^\uparrow$: indeed, taking $x = B \in V_1$ and $y = B \in V_2$ gives $(B, B) \in R$ since $|V(B)| \leq |V(B)|$.

Now define the depth-2 iterated metagraph

$$M^{(2)} = (V^{(2)}, E^{(2)}, s^{(2)}, t^{(2)}, \lambda^{(2)})$$

by

$$V^{(2)} = \{M_1, M_2\}, \quad E^{(2)} = \{\varepsilon\},$$

and

$$s^{(2)}(\varepsilon) = M_1, \quad t^{(2)}(\varepsilon) = M_2, \quad \lambda^{(2)}(\varepsilon) = R^\uparrow.$$

Then $M^{(2)}$ is an iterated metagraph of depth 2, because it is a metagraph over $(\mathfrak{G}^{(2)}, \mathcal{R}^{(2)})$ with vertices in $\mathfrak{G}^{(2)} = \mathfrak{G}^{(1)}$ and its unique meta-edge satisfies the incidence constraint

$$(s^{(2)}(\varepsilon), t^{(2)}(\varepsilon)) = (M_1, M_2) \in R^\uparrow = \lambda^{(2)}(\varepsilon).$$

An overview of Graphs, MetaGraphs, and Iterated MetaGraphs is presented in Table 1.

Table 1. Summary of Graph, MetaGraph, and Iterated MetaGraph

Object	Vertices are	Edges encode
Graph	atomic vertices (points)	adjacency /relations between vertices
MetaGraph (graph of graphs)	graphs $G \in \mathfrak{G}$ (meta-vertices)	specified relations between graphs (via labels $R \in \mathcal{R}$)
Iterated MetaGraph (depth t)	metagraphs of depth $t - 1$ (objects in $\mathfrak{G}^{(t)}$)	lifted relations between lower-level objects (labels in $\mathcal{R}^{(t)}$)

2. Main Results

This section presents the main results of this paper. For reference, a summary of Graph-in-Tree/Tree-in-Tree and Graph-in-Cycle/Cycle-in-Cycle is provided in Table 2. In this section, we briefly examine and discuss the fundamental properties of these concepts.

Table 2. Summary of Graph-in-Tree/Tree-in-Tree and Graph-in-Cycle/Cycle-in-Cycle

Class	Meta-level skeleton	What each meta-vertex contains
Graph in Tree	$\text{Skel}(\mathsf{T})$ is a finite tree	an arbitrary finite graph $G \in \mathfrak{G}$
Tree in Tree	$\text{Skel}(\mathsf{T})$ is a finite tree	a finite tree $T \in \mathfrak{T} \subseteq \mathfrak{G}$
Graph in Cycle	$\text{Skel}(\mathsf{C})$ is a finite simple cycle C_n ($n \geq 3$)	an arbitrary finite graph $G \in \mathfrak{G}$
Cycle in Cycle	$\text{Skel}(\mathsf{C})$ is a finite simple cycle C_n ($n \geq 3$)	a finite cycle graph $C_m \in \mathfrak{C} \subseteq \mathfrak{G}$ ($m \geq 3$)

2.1. Graph in Tree

Intuitively, a *Graph in Tree* is a tree-shaped metagraph: the meta-level incidence pattern is a tree, while each meta-vertex carries an entire (base-level) graph.

Definition 6 (Underlying (unlabelled) skeleton). *Let $M = (V, E, s, t, \lambda)$ be a metagraph over $(\mathfrak{G}, \mathcal{R})$. Its (undirected) skeleton is the simple undirected graph*

$$\text{Skel}(M) := (V, \{\{s(e), t(e)\} : e \in E\}),$$

obtained by forgetting orientations, labels, and multiplicities.

Definition 7 (Graph in Tree). *Fix a nonempty universe $\mathfrak{G}$ of finite graphs and a nonempty family of binary relations $\mathcal{R} \subseteq \mathcal{P}(\mathfrak{G} \times \mathfrak{G})$. A Graph in Tree over $(\mathfrak{G}, \mathcal{R})$ is a metagraph*

$$\mathsf{T} = (V, E, s, t, \lambda) \quad \text{over } (\mathfrak{G}, \mathcal{R})$$

such that its skeleton $\text{Skel}(\mathsf{T})$ is a (finite) tree.

Equivalently, it is the following explicit data:

- a finite (undirected) tree $T = (N, F)$,
- an assignment (node-labelling) $\iota : N \rightarrow \mathfrak{G}$,
- an assignment of relation labels $\Lambda : \vec{F} \rightarrow \mathcal{R}$ on the set

$$\vec{F} := \{(u, v) \in N \times N : \{u, v\} \in F\}$$

of oriented edges,
satisfying the incidence constraint

$$\forall (u, v) \in \vec{F} : \quad (\iota(u), \iota(v)) \in \Lambda(u, v).$$

In this viewpoint, the associated metagraph has meta-vertex set $V = \iota(N)$ and meta-edge set $E = \vec{F}$ with $s(u, v) = \iota(u)$, $t(u, v) = \iota(v)$, and $\lambda(u, v) = \Lambda(u, v)$.

Remark 1 (Rooted form). *If a root $\rho \in N$ is fixed, one may orient every edge away from ρ , obtaining a rooted version in which Λ is specified only on parent-to-child arrows. This is often convenient when interpreting T as a hierarchical “tree of graphs”.*

Example 3 (A concrete Graph in Tree). *Let $\mathfrak{G}$ be the class of all finite simple graphs and let $\mathcal{R} = \{R\}$ consist of the single relation*

$$(G, H) \in R \quad :\Longleftrightarrow \quad |V(G)| \leq |V(H)|.$$

(Thus, a meta-edge $G \xrightarrow{R} H$ is allowed precisely when the source graph has at most as many vertices as the target graph.)

Define three base graphs:

$$G_1 := P_2 \text{ (a single edge)}, \quad G_2 := P_3 \text{ (a path on 3 vertices)}, \quad G_3 := C_3 \text{ (a triangle)}.$$

Consider the metagraph

$$T = (V, E, s, t, \lambda)$$

with meta-vertex set $V = \{G_1, G_2, G_3\}$ and meta-edge set $E = \{e_{12}, e_{23}\}$ specified by

$$s(e_{12}) = G_1, \quad t(e_{12}) = G_2, \quad \lambda(e_{12}) = R, \quad s(e_{23}) = G_2, \quad t(e_{23}) = G_3, \quad \lambda(e_{23}) = R.$$

Then T is a Graph in Tree: its skeleton $\text{Skel}(T)$ is the path

$$G_1 - G_2 - G_3,$$

which is a tree, and the incidence constraints hold because $|V(G_1)| = 2 \leq 3 = |V(G_2)|$ and $|V(G_2)| = 3 \leq 3 = |V(G_3)|$.

Theorem 1 (Every Graph-in-Tree is a MetaGraph). *Fix a nonempty universe $\mathfrak{G}$ of finite graphs and a nonempty family of binary relations $\mathcal{R} \subseteq \mathcal{P}(\mathfrak{G} \times \mathfrak{G})$. Let T be a Graph in Tree over $(\mathfrak{G}, \mathcal{R})$ in the sense of Definition 7. Then T is a metagraph over $(\mathfrak{G}, \mathcal{R})$ (i.e. a MetaGraph in the sense of Definition [MetaGraph (graph of graphs)]).*

Proof. By Definition 7, a Graph in Tree over $(\mathfrak{G}, \mathcal{R})$ is defined to be a metagraph

$$T = (V, E, s, t, \lambda) \quad \text{over } (\mathfrak{G}, \mathcal{R})$$

whose skeleton $\text{Skel}(T)$ is a finite tree.

In particular, the data (V, E, s, t, λ) satisfy all axioms required of a metagraph over $(\mathfrak{G}, \mathcal{R})$:

$$V \subseteq \mathfrak{G}, \quad s, t : E \rightarrow V, \quad \lambda : E \rightarrow \mathcal{R},$$

and the incidence condition holds for every $e \in E$,

$$(s(e), t(e)) \in \lambda(e).$$

These are exactly the defining conditions of a metagraph (MetaGraph) over $(\mathfrak{G}, \mathcal{R})$. The additional requirement that $\text{Skel}(T)$ is a tree is an extra structural constraint on this metagraph and does not alter metagraphhood. Hence T is a MetaGraph. $\square$

Let

$$T = (V, E, s, t, \lambda)$$

be a Graph in Tree over $(\mathfrak{G}, \mathcal{R})$ in the sense of Definition 7, and let $\text{Skel}(T)$ denote its skeleton (Definition 6).

Theorem 2 (Weak connectivity). *Every Graph in Tree T is weakly connected; equivalently, the underlying undirected adjacency graph on V induced by E is connected.*

Proof. By definition, the skeleton $\text{Skel}(T)$ is the simple undirected graph

$$\text{Skel}(T) = (V, \{\{s(e), t(e)\} : e \in E\}).$$

A Graph in Tree is defined by the condition that $\text{Skel}(T)$ is a finite tree, and every tree is connected. Hence the undirected adjacency graph induced by E is connected, which is precisely weak connectivity of T . $\square$

Proposition 1 (Acyclicity at the meta-level). *The skeleton $\text{Skel}(T)$ contains no (undirected) cycles. In particular, for any distinct $X, Y \in V$ there exists a unique simple undirected path in $\text{Skel}(T)$ connecting X and Y .*

Proof. Since $\text{Skel}(T)$ is a tree, it is by definition acyclic. A standard characterization of trees is that between any two vertices there is a unique simple path. Applying this characterization to $\text{Skel}(T)$ yields the claim. $\square$

Theorem 3 (Meta-edge lower bound and the tree case). *Let $n := |V|$. Then $n \geq 1$ and*

$$|E| \geq n - 1.$$

Moreover, if T has no multiplicities at the skeleton level in the sense that for every unordered pair $\{X, Y\} \subseteq V$ there exists at most one meta-edge $e \in E$ with $\{s(e), t(e)\} = \{X, Y\}$, then

$$|E| = n - 1.$$

Proof. Since $\text{Skel}(T)$ is a finite tree on n vertices, it has exactly $n - 1$ undirected edges. Each undirected skeleton edge $\{X, Y\}$ arises from at least one meta-edge $e \in E$ satisfying $\{s(e), t(e)\} = \{X, Y\}$ by the definition of the skeleton. Therefore, to realize all $n - 1$ distinct skeleton edges, the set E must contain at least $n - 1$ meta-edges; hence $|E| \geq n - 1$.

If, additionally, there is at most one meta-edge realizing each unordered pair, then each of the $n - 1$ skeleton edges is realized by exactly one meta-edge, and thus $|E| = n - 1$. $\square$

Proposition 2 (Skeleton-preserving minimal submetagraph). *Let $n := |V|$. There exists a subset $E_0 \subseteq E$ with $|E_0| = n - 1$ such that the restricted structure*

$$T_0 := (V, E_0, s|_{E_0}, t|_{E_0}, \lambda|_{E_0})$$

is again a Graph in Tree and satisfies

$$\text{Skel}(T_0) = \text{Skel}(T).$$

Proof. The tree $\text{Skel}(T)$ has exactly $n - 1$ undirected edges. For each skeleton edge $\{X, Y\}$, choose one meta-edge $e_{\{X, Y\}} \in E$ with $\{s(e_{\{X, Y\}}), t(e_{\{X, Y\}})\} = \{X, Y\}$ (such an edge exists by the definition of the skeleton). Let E_0 be the set of the chosen meta-edges. Then $|E_0| = n - 1$.

Restricting s, t, λ to E_0 preserves the incidence condition $(s(e), t(e)) \in \lambda(e)$ for all $e \in E_0$, since it holds for all $e \in E$. By construction, the set of unordered pairs $\{\{s(e), t(e)\} : e \in E_0\}$ equals the edge set of $\text{Skel}(T)$, so $\text{Skel}(T_0) = \text{Skel}(T)$, which is a tree. Hence T_0 is a Graph in Tree. $\square$

2.2. Tree in Tree

A *Tree in Tree* is a Graph in Tree whose embedded graphs are themselves trees.

Definition 8 (Tree universe). Let $\mathfrak{T} \subseteq \mathfrak{G}$ denote the class of all finite trees (viewed as graphs) that lie in the ambient universe $\mathfrak{G}$. Define the restricted relation family

$$\mathcal{R} \upharpoonright_{\mathfrak{T}} := \{ R \cap (\mathfrak{T} \times \mathfrak{T}) : R \in \mathcal{R} \}.$$

Definition 9 (Tree in Tree). A Tree in Tree over $(\mathfrak{G}, \mathcal{R})$ is a Graph in Tree $\mathsf{T} = (V, E, s, t, \lambda)$ over $(\mathfrak{G}, \mathcal{R})$ such that every meta-vertex is a tree, i.e. $V \subseteq \mathfrak{T}$.

Equivalently, in the explicit description of Definition 7, the node-labelling satisfies $\iota(N) \subseteq \mathfrak{T}$ and the labels take values in $\mathcal{R} \upharpoonright_{\mathfrak{T}}$.

Example 4 (A concrete Tree in Tree). Let $\mathfrak{G}$ be the class of all finite simple graphs, and let $\mathfrak{T} \subseteq \mathfrak{G}$ be the subclass of finite trees. Let $\mathcal{R} = \{R\}$ where

$$(T_1, T_2) \in R \iff \text{diam}(T_1) \leq \text{diam}(T_2),$$

and $\text{diam}(T)$ denotes the diameter of a tree T .

Define three trees:

$$T_1 := P_2 \text{ (a single edge)}, \quad T_2 := P_3 \text{ (a path on 3 vertices)}, \quad T_3 := K_{1,3} \text{ (a star on 4 vertices)}.$$

Their diameters are

$$\text{diam}(T_1) = 1, \quad \text{diam}(T_2) = 2, \quad \text{diam}(T_3) = 2.$$

Consider the metagraph

$$\mathsf{T} = (V, E, s, t, \lambda)$$

with

$$V = \{T_1, T_2, T_3\}, \quad E = \{e_{12}, e_{23}\},$$

and structure maps given by

$$s(e_{12}) = T_1, \quad t(e_{12}) = T_2, \quad \lambda(e_{12}) = R, \quad s(e_{23}) = T_2, \quad t(e_{23}) = T_3, \quad \lambda(e_{23}) = R.$$

Then T is a Tree in Tree: every meta-vertex is a tree (so $V \subseteq \mathfrak{T}$), and the skeleton

$$T_1 - T_2 - T_3$$

is a tree. Moreover, the incidence constraints hold because $\text{diam}(T_1) = 1 \leq 2 = \text{diam}(T_2)$ and $\text{diam}(T_2) = 2 \leq 2 = \text{diam}(T_3)$.

2.3. Graph in Cycle

Replacing the tree-shaped skeleton by a cycle yields a cyclic variant.

Definition 10 (Graph in Cycle). A Graph in Cycle over $(\mathfrak{G}, \mathcal{R})$ is a metagraph $\mathsf{C} = (V, E, s, t, \lambda)$ over $(\mathfrak{G}, \mathcal{R})$ such that its skeleton $\text{Skel}(\mathsf{C})$ is a (finite) simple cycle graph. Equivalently, it is data (C_n, ι, Λ) where C_n is an n -cycle ($n \geq 3$), together with $\iota : V(C_n) \rightarrow \mathfrak{G}$ and $\Lambda : \vec{E}(C_n) \rightarrow \mathcal{R}$ satisfying $(\iota(u), \iota(v)) \in \Lambda(u, v)$ for every oriented edge (u, v) of C_n .

Example 5 (A concrete Graph in Cycle). Let $\mathfrak{G}$ be the class of all finite simple graphs, and let $\mathcal{R} = \{R\}$ where

$$(G, H) \in R \iff |E(G)| \leq |E(H)|.$$

Define three base graphs:

$$G_1 := P_3 \text{ (a path on 3 vertices)}, \quad G_2 := C_3 \text{ (a triangle)}, \quad G_3 := K_4 \text{ (the complete graph on 4 vertices)}.$$

Their numbers of edges are $|E(G_1)| = 2$, $|E(G_2)| = 3$, and $|E(G_3)| = 6$.

Consider the metagraph

$$C = (V, E, s, t, \lambda)$$

with meta-vertex set $V = \{G_1, G_2, G_3\}$ and meta-edge set $E = \{e_{12}, e_{23}, e_{31}\}$ given by

$$s(e_{12}) = G_1, t(e_{12}) = G_2, \lambda(e_{12}) = R,$$

$$s(e_{23}) = G_2, t(e_{23}) = G_3, \lambda(e_{23}) = R,$$

$$s(e_{31}) = G_1, t(e_{31}) = G_3, \lambda(e_{31}) = R.$$

Then C is a Graph in Cycle because its skeleton is the 3-cycle

$$G_1 - G_2 - G_3 - G_1,$$

and every meta-edge satisfies the incidence constraint: $|E(G_1)| = 2 \leq 3 = |E(G_2)|$, $|E(G_2)| = 3 \leq 6 = |E(G_3)|$, and $|E(G_1)| = 2 \leq 6 = |E(G_3)|$.

Let

$$C = (V, E, s, t, \lambda)$$

be a Graph in Cycle over $(\mathfrak{G}, \mathcal{R})$ in the sense of Definition 10, and let $\text{Skel}(C)$ be its skeleton (Definition 6).

Theorem 4 (Every Graph-in-Cycle is a MetaGraph). *Fix a nonempty universe $\mathfrak{G}$ of finite graphs and a nonempty family of binary relations $\mathcal{R} \subseteq \mathcal{P}(\mathfrak{G} \times \mathfrak{G})$. Let C be a Graph in Cycle over $(\mathfrak{G}, \mathcal{R})$ in the sense of Definition 10. Then C is a metagraph over $(\mathfrak{G}, \mathcal{R})$ (i.e. a MetaGraph in the sense of Definition [MetaGraph (graph of graphs)]).*

Proof. By Definition 10, a Graph in Cycle over $(\mathfrak{G}, \mathcal{R})$ is, by construction, a metagraph

$$C = (V, E, s, t, \lambda) \quad \text{over } (\mathfrak{G}, \mathcal{R})$$

whose skeleton $\text{Skel}(C)$ is a finite simple cycle graph.

In particular, the defining axioms of a metagraph are satisfied:

$$V \subseteq \mathfrak{G}, \quad s, t : E \rightarrow V, \quad \lambda : E \rightarrow \mathcal{R},$$

and for every meta-edge $e \in E$ the incidence condition holds,

$$(s(e), t(e)) \in \lambda(e).$$

These conditions are exactly those required for C to be a metagraph (MetaGraph) over $(\mathfrak{G}, \mathcal{R})$. The additional requirement that $\text{Skel}(C)$ is a cycle merely restricts the allowable incidence pattern at the meta-level and does not affect metagraphhood. Hence C is a MetaGraph. $\square$

Lemma 1 (Local structure of the skeleton). *Let $n := |V|$. Then $n \geq 3$ and, for every $X \in V$, there exist exactly two distinct vertices $X^-, X^+ \in V$ such that the neighbors of X in the skeleton are precisely $\{X^-, X^+\}$. Equivalently, in $\text{Skel}(C)$ every vertex has degree 2.*

Proof. By assumption, $\text{Skel}(C)$ is a finite simple cycle graph, hence it has at least 3 vertices and every vertex has degree 2. Translating this statement to the vertex set V of $\text{Skel}(C)$ yields the claim. $\square$

Theorem 5 (Cyclic ordering of meta-vertices). *Let $n := |V|$. There exists an enumeration*

$$V = \{X_0, X_1, \dots, X_{n-1}\}$$

such that the edge set of the skeleton is exactly

$$E(\text{Skel}(C)) = \{\{X_i, X_{i+1}\} : i = 0, 1, \dots, n-1\},$$

where indices are taken modulo n (so $X_n = X_0$). In particular, the only unordered pairs of distinct meta-vertices that can appear as $\{s(e), t(e)\}$ are consecutive pairs $\{X_i, X_{i+1}\}$.

Proof. Since $\text{Skel}(C)$ is a simple cycle on n vertices, it is isomorphic to the standard cycle graph C_n . Choose an isomorphism $\varphi : C_n \rightarrow \text{Skel}(C)$ and write $X_i := \varphi(i)$ for $i \in \{0, 1, \dots, n-1\}$. Under this identification, the edges of $\text{Skel}(C)$ are precisely the images of the cycle edges $\{i, i+1\}$, hence they are exactly $\{X_i, X_{i+1}\}$ (indices modulo n). $\square$

Proposition 3 (A lower bound on the number of meta-edges). *Let $n := |V|$. Then $|E| \geq n$. Moreover, if C has at most one meta-edge whose endpoints form a given unordered pair (i.e. for each $\{X, Y\} \subseteq V$ there is at most one $e \in E$ with $\{s(e), t(e)\} = \{X, Y\}$), then $|E| = n$.*

Proof. The skeleton $\text{Skel}(C)$ has exactly n undirected edges. By definition of the skeleton, each undirected edge $\{X, Y\}$ of $\text{Skel}(C)$ arises from at least one meta-edge $e \in E$ with $\{s(e), t(e)\} = \{X, Y\}$. Since a single meta-edge contributes to exactly one unordered pair $\{s(e), t(e)\}$, covering n distinct skeleton edges requires at least n meta-edges. Hence $|E| \geq n$.

If, in addition, there is at most one meta-edge realizing each unordered pair, then each of the n skeleton edges is realized by exactly one meta-edge, so $|E| = n$. $\square$

Theorem 6 (Weak connectivity). *Every Graph in Cycle C is weakly connected; equivalently, the underlying undirected adjacency graph on V induced by E is connected.*

Proof. The underlying undirected adjacency graph on V has an edge between X and Y precisely when there exists $e \in E$ with $\{s(e), t(e)\} = \{X, Y\}$, which is exactly the definition of the skeleton $\text{Skel}(C)$. Since $\text{Skel}(C)$ is a cycle graph, it is connected. Therefore C is weakly connected. $\square$

Proposition 4 (Skeleton-preserving minimal submetagraph). *Let $n := |V|$. There exists a subset $E_0 \subseteq E$ with $|E_0| = n$ such that the restricted structure*

$$C_0 := (V, E_0, s|_{E_0}, t|_{E_0}, \lambda|_{E_0})$$

is again a Graph in Cycle and satisfies $\text{Skel}(C_0) = \text{Skel}(C)$.

Proof. The skeleton $\text{Skel}(C)$ has exactly n edges. For each undirected skeleton edge $\{X, Y\}$, choose one meta-edge $e_{\{X, Y\}} \in E$ satisfying $\{s(e_{\{X, Y\}}), t(e_{\{X, Y\}})\} = \{X, Y\}$ (such an edge exists by the definition of the skeleton). Let E_0 be the set of these chosen edges; then $|E_0| = n$.

Restricting s, t, λ to E_0 preserves the metagraph incidence condition $(s(e), t(e)) \in \lambda(e)$ for all $e \in E_0$ (since it held for all $e \in E$). By construction, the set of unordered pairs $\{\{s(e), t(e)\} : e \in E_0\}$ equals the skeleton edge set, hence $\text{Skel}(C_0) = \text{Skel}(C)$, which is a cycle graph. Therefore C_0 is a Graph in Cycle. $\square$

Theorem 7 (A sufficient condition for strong connectivity). *Let $V = \{X_0, \dots, X_{n-1}\}$ be a cyclic ordering as in Theorem 5. Assume there exist meta-edges $e_i \in E$ ($i = 0, \dots, n-1$) such that*

$$s(e_i) = X_i, \quad t(e_i) = X_{i+1} \quad (\text{indices modulo } n).$$

Then C is strongly connected (as a directed multigraph).

Proof. The edges $e_0, e_1, \dots, e_{n-1}$ form a directed cycle

$$X_0 \rightarrow X_1 \rightarrow \dots \rightarrow X_{n-1} \rightarrow X_0.$$

In a directed cycle, every vertex can reach every other vertex by following the forward direction around the cycle. Hence, for any $X_i, X_j \in V$, there exists a directed path from X_i to X_j using a suitable subword of the cycle edges. Therefore C is strongly connected. $\square$

2.4. Cycle in Cycle

A further specialization is obtained by requiring each embedded graph to be a cycle.

Definition 11 (Cycle universe). Let $\mathfrak{C} \subseteq \mathfrak{G}$ denote the class of all finite simple cycle graphs (i.e. graphs isomorphic to C_m for some $m \geq 3$) that lie in $\mathfrak{G}$. Define

$$\mathcal{R} \upharpoonright_{\mathfrak{C}} := \{ R \cap (\mathfrak{C} \times \mathfrak{C}) : R \in \mathcal{R} \}.$$

Definition 12 (Cycle in Cycle). A Cycle in Cycle over $(\mathfrak{G}, \mathcal{R})$ is a Graph in Cycle $C = (V, E, s, t, \lambda)$ over $(\mathfrak{G}, \mathcal{R})$ whose meta-vertex set consists of cycles, i.e. $V \subseteq \mathfrak{C}$. Equivalently, in Definition 10 one requires $\iota(V(C_n)) \subseteq \mathfrak{C}$ and labels in $\mathcal{R} \upharpoonright_{\mathfrak{C}}$.

Example 6 (A concrete Cycle in Cycle). Let $\mathfrak{G}$ be the class of all finite simple graphs and let $\mathfrak{C} \subseteq \mathfrak{G}$ be the subclass of cycle graphs. Let $\mathcal{R} = \{R\}$ where

$$(X, Y) \in R \iff |V(X)| \leq |V(Y)|.$$

Choose three cycle graphs

$$C_3, \quad C_4, \quad C_5.$$

Consider the metagraph

$$C^\circ = (V, E, s, t, \lambda)$$

with

$$V = \{C_3, C_4, C_5\}, \quad E = \{e_{34}, e_{45}, e_{35}\},$$

and structure maps

$$\begin{aligned} s(e_{34}) &= C_3, \quad t(e_{34}) = C_4, \quad \lambda(e_{34}) = R, \\ s(e_{45}) &= C_4, \quad t(e_{45}) = C_5, \quad \lambda(e_{45}) = R, \\ s(e_{35}) &= C_3, \quad t(e_{35}) = C_5, \quad \lambda(e_{35}) = R. \end{aligned}$$

Then C° is a Cycle in Cycle: all meta-vertices are cycle graphs (so $V \subseteq \mathfrak{C}$), its skeleton is the 3-cycle

$$C_3 - C_4 - C_5 - C_3,$$

and the incidence constraints hold because $3 \leq 4$, $4 \leq 5$, and $3 \leq 5$.

3. Additional Result: Spiral Graph

The purpose of this section is to formalize a “spiral” as a graph obtained by iteratively *gluing* a prescribed family of finite *blocks* along designated boundary vertices (ports). A *type word* (finite or infinite) specifies which block type is used at each step. The PDCA cycle (Plan–Do–Check–Act) serves as a motivating example, but the construction applies to any finite alphabet of types.

3.1. Types and Semantics

This subsection fixes the type alphabet Σ and introduces an optional semantic interpretation map $\text{Sem} : \Sigma \rightarrow \mathcal{S}$, which allows one to declare when two types (and hence two blocks) represent the same meaning.

Definition 13 (Type alphabet and semantics). *Fix a finite alphabet (set of types) Σ . A semantic interpretation is a function*

$$\text{Sem} : \Sigma \rightarrow \mathcal{S},$$

where $\mathcal{S}$ is an arbitrary set of “meanings” (e.g. phases, labels, actions, roles). Two types $\sigma, \tau \in \Sigma$ are said to have the same meaning if $\text{Sem}(\sigma) = \text{Sem}(\tau)$.

Example 7 (PDCA as an interpretation). *Let $\Sigma = \{P, D, C, A\}$ and $\mathcal{S} = \{\text{Plan}, \text{Do}, \text{Check}, \text{Act}\}$. Define $\text{Sem}(P) = \text{Plan}$, $\text{Sem}(D) = \text{Do}$, $\text{Sem}(C) = \text{Check}$, and $\text{Sem}(A) = \text{Act}$. This matches the standard PDCA meaning; other applications may use different alphabets and interpretations.*

3.2. Blocks with Entrance/Exit Ports

This subsection defines the building blocks used in the construction. We formalize a block as a finite graph equipped with two distinguished vertices (entrance and exit ports), and record special cases such as rooted path blocks and Σ -indexed families of blocks.

Definition 14 (Two-terminal (ported) block). *A two-terminal block is a triple*

$$\mathcal{B} = (B, \alpha, \omega),$$

where B is a finite graph and $\alpha, \omega \in V(B)$ are distinguished vertices called the entrance port and exit port, respectively. We do not require $\alpha \neq \omega$, although in most applications $\alpha \neq \omega$. (One may additionally assume B is connected; the definition itself does not require this.)

Definition 15 (Rooted path block). *A rooted path block is a two-terminal block $\mathcal{B} = (B, \alpha, \omega)$ such that B is a finite simple path graph and its endpoints are α and ω .*

Definition 16 (Type-indexed block family). *Fix a type alphabet Σ . A Σ -indexed block family is a choice of two-terminal blocks*

$$\{\mathcal{B}_\sigma = (B_\sigma, \alpha_\sigma, \omega_\sigma)\}_{\sigma \in \Sigma}.$$

If every B_σ is a path, we call it a path-block family.

3.3. Words and Periodicity

This subsection specifies how a finite or infinite word over Σ prescribes an ordered sequence of block types. In particular, we define periodic infinite words (e.g. u^ω), which will be used to model repeating patterns such as PDCA.

Definition 17 (Words and periodic words). *A word over Σ is a function $w : I_w \rightarrow \Sigma$ where either*

$$I_w = \{0, 1, \dots, n-1\} \quad (n \in \mathbb{N}) \quad (\text{finite word})$$

or

$$I_w = \mathbb{N}_0 = \{0, 1, 2, \dots\} \quad (\text{one-sided infinite word}).$$

We write $w(i) = \sigma_i$ and denote the word by $\sigma_0\sigma_1 \cdots$ when convenient.

An infinite word $w : \mathbb{N}_0 \rightarrow \Sigma$ is periodic with period $m \geq 1$ if $\sigma_{i+m} = \sigma_i$ for all $i \geq 0$. For a finite word $u = \tau_0 \cdots \tau_{m-1} \in \Sigma^m$, we write u^ω for the infinite periodic word

$$u^\omega := \tau_0 \cdots \tau_{m-1} \tau_0 \cdots \tau_{m-1} \cdots.$$

3.4. Spiral Graphs by Gluing

This subsection gives the core construction of the spiral graph $S(w)$. We define $S(w)$ as a quotient graph obtained by gluing consecutive blocks along their ports according to the word w , and we introduce the resulting junction vertices.

Definition 18 (Spiral graph induced by a word). Fix a Σ -indexed block family $\{\mathcal{B}_\sigma\}_{\sigma \in \Sigma}$ and a word $w : I_w \rightarrow \Sigma$. The spiral graph $S(w)$ is constructed as follows.

For each $i \in I_w$, take a disjoint copy

$$\mathcal{B}_{\sigma_i}^{(i)} = (\mathcal{B}_{\sigma_i}^{(i)}, \alpha_{\sigma_i}^{(i)}, \omega_{\sigma_i}^{(i)}), \quad \sigma_i := w(i).$$

Let G_0 be the disjoint union of the graphs $\mathcal{B}_{\sigma_i}^{(i)}$ over all $i \in I_w$. Let $\sim$ be the smallest equivalence relation on $V(G_0)$ such that

$$\omega_{\sigma_i}^{(i)} \sim \alpha_{\sigma_{i+1}}^{(i+1)} \quad \text{for every } i \in I_w \text{ with } i+1 \in I_w.$$

Define $S(w)$ to be the quotient graph $G_0 / \sim$, i.e. the graph obtained by identifying vertices in the same $\sim$ -class. The identified vertices are called junction vertices. We set

$$v_0 := [\alpha_{\sigma_0}^{(0)}]_\sim, \quad v_i := [\omega_{\sigma_{i-1}}^{(i-1)}]_\sim = [\alpha_{\sigma_i}^{(i)}]_\sim \quad (i \geq 1, i \in I_w),$$

whenever these indices are defined.

Remark 2 (On simplicity vs. multigraphs). The quotient construction above is canonical at the level of vertex identifications. If one insists on working only with simple graphs, one may additionally simplify the result by deleting loops and merging parallel edges. In the examples below, no such simplification is necessary.

Remark 3 (Path concatenation case). If each $\mathcal{B}_\sigma$ is a rooted path block, then each block has a well-defined first vertex α and last vertex ω , and Definition 18 identifies the last vertex of block i with the first vertex of block $i+1$. Thus $S(w)$ is literally a concatenation of paths (more generally, of two-terminal graphs).

Example 8 (Finite spiral with nontrivial path lengths). Let $\Sigma = \{A, B\}$ and choose a path-block family

$$\mathcal{B}_A = (P_2, \alpha_A, \omega_A) \quad (\text{one edge}), \quad \mathcal{B}_B = (P_3, \alpha_B, \omega_B) \quad (\text{two edges}).$$

Take the finite word $w : \{0, 1, 2, 3\} \rightarrow \Sigma$ given by

$$w = ABBA.$$

Then $S(w)$ is obtained by gluing four blocks in sequence:

$$\omega_A^{(0)} \equiv \alpha_B^{(1)}, \quad \omega_B^{(1)} \equiv \alpha_B^{(2)}, \quad \omega_B^{(2)} \equiv \alpha_A^{(3)}.$$

The resulting graph is a single path. Its number of edges equals the sum of the block lengths,

$$|E(S(w))| = 1 + 2 + 2 + 1 = 6,$$

so $S(ABBA) \cong P_7$ (a path on 7 vertices). The junction vertices are

$$v_0 = \alpha_A^{(0)}, \quad v_1 = \omega_A^{(0)} = \alpha_B^{(1)}, \quad v_2 = \omega_B^{(1)} = \alpha_B^{(2)}, \quad v_3 = \omega_B^{(2)} = \alpha_A^{(3)}.$$

Hence the word ABBA records a segmentation of the long path into blocks of edge-lengths 1, 2, 2, 1.

Example 9 (Infinite spiral from repeating triangle blocks). Let $\Sigma = \{T\}$ and let $w = T^\omega = TTT \dots$ be the constant infinite word. Define the block $\mathcal{B}_T = (B_T, \alpha_T, \omega_T)$ by letting B_T be a triangle C_3 with vertices $\{u_0, u_1, u_2\}$, and choose distinct ports

$$\alpha_T = u_0, \quad \omega_T = u_1.$$

For each $i \geq 0$, take a disjoint copy $\mathcal{B}_T^{(i)} = (C_3^{(i)}, \alpha_T^{(i)}, \omega_T^{(i)})$ and glue them by identifying

$$\omega_T^{(i)} \equiv \alpha_T^{(i+1)} \quad (i \geq 0).$$

Then $S(T^\omega)$ is an infinite connected graph in which consecutive triangles share exactly one vertex. If x_i denotes the third (non-port) vertex of the i -th triangle, then for each $i \geq 0$ the i -th triangle has vertex set

$$\{v_i, v_{i+1}, x_i\}$$

and edge set

$$\{v_i, v_{i+1}\}, \{v_{i+1}, x_i\}, \{x_i, v_i\}.$$

Thus $S(T^\omega)$ can be viewed as a “chain of triangles” extending indefinitely, each new triangle attached at the previous exit vertex.

3.5. Meaning Equivalence and Variants

This subsection formalizes the notion of “same meaning” between steps using Sem, and introduces common variants of the construction. In particular, we define finite truncations $S_n(w)$ of an infinite spiral and cyclic spirals $S^\circ(w)$ obtained by closing a finite spiral.

Definition 19 (Type, meaning, and step equivalence). Let $S(w)$ be as in Definition 18, and fix a semantic interpretation $\text{Sem} : \Sigma \rightarrow \mathcal{S}$. For each $i \in I_w$, define

$$\text{type}(i) := \sigma_i = w(i), \quad \text{meaning}(i) := \text{Sem}(\sigma_i).$$

We say that steps (blocks) i and j have the same meaning if

$$i \sim_{\text{Sem}} j \iff \text{meaning}(i) = \text{meaning}(j) \iff \text{Sem}(\sigma_i) = \text{Sem}(\sigma_j).$$

In particular, if $\text{Sem} = \text{id}_\Sigma$, then $i \sim_{\text{Sem}} j$ is equivalent to $\sigma_i = \sigma_j$.

Remark 4 (PDCA meaning and periodicity). For $w = (\text{PDCA})^\omega$ and Sem as in Example 7, one has

$$\text{meaning}(4k) = \text{Plan}, \quad \text{meaning}(4k+1) = \text{Do}, \quad \text{meaning}(4k+2) = \text{Check}, \quad \text{meaning}(4k+3) = \text{Act} \quad (k \geq 0),$$

and therefore $i \sim_{\text{Sem}} j$ holds exactly when $i \equiv j \pmod{4}$.

Definition 20 (Finite truncation). If $w : \mathbb{N}_0 \rightarrow \Sigma$ is infinite and $n \in \mathbb{N}$, the n -step truncation $S_n(w)$ is the spiral graph induced by the restriction $w \upharpoonright_{\{0,1,\dots,n-1\}}$. Equivalently, it is obtained by gluing blocks $0, 1, \dots, n-1$ and performing identifications $\omega_{\sigma_i}^{(i)} \equiv \alpha_{\sigma_{i+1}}^{(i+1)}$ for $0 \leq i \leq n-2$.

Definition 21 (Cyclic spiral). Let $w : \{0, 1, \dots, n-1\} \rightarrow \Sigma$ be a finite word. The cyclic spiral $S^\circ(w)$ is obtained from $S(w)$ by additionally identifying the last exit with the first entrance:

$$\omega_{\sigma_{n-1}}^{(n-1)} \equiv \alpha_{\sigma_0}^{(0)}.$$

Example 10 (A concrete cyclic spiral). Let $\Sigma = \{P, D, C, A\}$ and take the finite word $w = PDCA$. For each $\sigma \in \Sigma$, let B_σ be the one-edge rooted path block. Then $S(PDCA)$ is a path on 5 vertices (with 4 edges), whose edges are typed in order by P, D, C, A .
The cyclic spiral $S^\circ(PDCA)$ is obtained by identifying the last exit with the first entrance,

$$\omega_A^{(3)} \equiv \alpha_P^{(0)}.$$

In this (degenerate) block case, the identification closes the chain and yields a 4-cycle:

$$S^\circ(PDCA) \cong C_4,$$

where the four consecutive edges inherit the types P, D, C, A around the cycle.

4. Conclusion

In this paper, we defined a new class of metagraphs. We expect that future work will explore extensions based on Fuzzy Sets[10], Neutrosophic Sets[11,12], Hesitant Neutrosophic Sets[13,14], Uncertain Sets[15,16], and Plithogenic Sets[17,18], as well as applications to methods such as neural networks. In addition, we expect that studies on extended frameworks based on HyperGraphs [19,20] and SuperHyperGraphs [21,22] will be pursued in parallel.

Use of Computational Tools

All proofs and derivations were performed manually; no computational software (e.g., Mathematica, SageMath, Coq) was used.

Code Availability

No code or software was developed for this study.

Ethical Approval

This research did not involve human participants or animals, and therefore did not require ethical approval.

Use of Generative AI and AI-Assisted Tools

We use generative AI and AI-assisted tools for tasks such as English grammar checking, and We do not employ them in any way that violates ethical standards.

Disclaimer

The ideas presented here are theoretical and have not yet been validated through empirical testing. While we have strived for accuracy and proper citation, inadvertent errors may remain. Readers should verify any referenced material independently. The opinions expressed are those of the authors and do not necessarily reflect the views of their institutions.

Funding: No external funding was received for this work.

Informed Consent Statement: The author confirms that this manuscript is original, has not been published elsewhere, and is not under consideration by any other journal.

Data Availability Statement: This paper is theoretical and did not generate or analyze any empirical data. We welcome future studies that apply and test these concepts in practical settings.

Acknowledgments: We thank all colleagues, reviewers, and readers whose comments and questions have greatly improved this manuscript. We are also grateful to the authors of the works cited herein for providing the theoretical foundations that underpin our study. Finally, we appreciate the institutional and technical support that enabled this research.

Conflicts of Interest: The authors declare no conflicts of interest regarding the publication of this work.

References

1. Diestel, R. *Graph theory*; Springer (print edition); Reinhard Diestel (eBooks), 2024.
2. Gross, J.L.; Yellen, J.; Anderson, M. *Graph theory and its applications*; Chapman and Hall/CRC, 2018.
3. Azevedo, R.; Lohaus, R.; Paixão, T. Networking networks. *Evol Dev* **2008**, *10*, 514–515.
4. Donnat, C.; Holmes, S. Tracking network dynamics: A survey using graph distances. *The Annals of Applied Statistics* **2018**, *12*, 971–1012.
5. Martin, O.C.; Wagner, A. Multifunctionality and robustness trade-offs in model genetic circuits. *Biophysical journal* **2008**, *94*, 2927–2937.
6. Fujita, T. MetaHyperGraphs, MetaSuperHyperGraphs, and Iterated MetaGraphs: Modeling Graphs of Graphs, Hypergraphs of Hypergraphs, Superhypergraphs of Superhypergraphs, and Beyond. *Current Research in Interdisciplinary Studies* **2025**, *4*, 1–23. <https://doi.org/10.58614/cris451>.
7. Fujita, T. Some Meta-Graph Structures: Mixed Graph, DiHyperGraph, Knowledge Graph, Intersection Graph, and Chemical Graph, 2025.
8. Fujita, T. MetaStructure, Meta-HyperStructure, and Meta-SuperHyper Structure. *Journal of Computers and Applications* **2025**, *1*, 1–22. Received: 05.10.2025; Accepted: 30.10.2025; Published: 07.11.2025, <https://doi.org/10.58613/jca111>.
9. Cao, J.; Zhang, S.; Chen, Q.; Wang, H.; Wang, M.; Liu, N. Network-wide task offloading with leo satellites: A computation and transmission fusion approach. *arXiv preprint arXiv:2211.09672* **2022**.
10. Zadeh, L.A. Fuzzy sets. *Information and control* **1965**, *8*, 338–353.
11. Wang, H.; Smarandache, F.; Zhang, Y.; Sunderraman, R. *Single valued neutrosophic sets*; Infinite study, 2010.
12. Broumi, S.; Talea, M.; Bakali, A.; Smarandache, F. Single valued neutrosophic graphs. *Journal of New theory* **2016**, *10*, 86–101.
13. Zhao, H.; Zhang, H. On hesitant neutrosophic rough set over two universes and its application. *Artificial Intelligence Review* **2019**, *53*, 4387 – 4406.
14. Zheng, Y.; Zhang, L. A Novel Image Segmentation Algorithm Based on Hesitant Neutrosophic and Level Set. In Proceedings of the 2021 International Conference on Electronic Information Technology and Smart Agriculture (ICEITSA). IEEE, 2021, pp. 370–373.
15. Fujita, T.; Smarandache, F. *A Dynamic Survey of Fuzzy, Intuitionistic Fuzzy, Neutrosophic, , and Extensional Sets*; Neutrosophic Science International Association (NSIA), 2025.
16. Fujita, T.; Smarandache, F. 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* **2025**, *91*, 337–380.
17. Smarandache, F. *Plithogenic set, an extension of crisp, fuzzy, intuitionistic fuzzy, and neutrosophic sets-revisited*; Infinite study, 2018.
18. Azeem, M.; Rashid, H.; Jamil, M.K.; Gütmen, S.; Tirkolaee, E.B. Plithogenic fuzzy graph: A study of fundamental properties and potential applications. *Journal of Dynamics and Games* **2024**, pp. 0–0.
19. Feng, Y.; You, H.; Zhang, Z.; Ji, R.; Gao, Y. Hypergraph neural networks. In Proceedings of the AAAI conference on artificial intelligence, 2019, pp. 3558–3565.
20. Cai, D.; Song, M.; Sun, C.; Zhang, B.; Hong, S.; Li, H. Hypergraph Structure Learning for Hypergraph Neural Networks. In Proceedings of the IJCAI, 2022, pp. 1923–1929.
21. Smarandache, F. *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.
22. Smarandache, F. *Introduction to the n-SuperHyperGraph-the most general form of graph today*; Infinite Study, 2022.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.