Discrete Applied Mathematics 355 (2024) 232–246

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

The complexity of 2-intersection graphs of 3-hypergraphs recognition for claw-free graphs and triangulated claw-free graphs

N. Di Marco [a], A. Frosini [a], C. Picouleau [b,*]

[a] Dipartimento di Matematica e Informatica, Università di Firenze, Firenze, Italy
[b] Conservatoire National des Arts et Métiers, CEDRIC laboratory, Paris, France

ARTICLE INFO

Article history:
Received 13 October 2023
Received in revised form 20 March 2024
Accepted 7 May 2024
Available online 18 May 2024

Keywords:
Uniform hypergraph
Intersection graph
Triangulated graph
NP-complete

ABSTRACT

Given a 3-uniform hypergraph H, its 2-intersection graph G has as vertex set the hyperedges of H and ee' is an edge of G whenever e and e' have exactly two common vertices in H. Di Marco et al. prove in Di Marco et al. (2023) that deciding whether a graph G is the 2-intersection graph of a 3-uniform hypergraph is NP-complete. Following this result, we study the class of claw-free graphs. We show that the recognition problem remains NP-complete for that class, but becomes polynomial if we consider triangulated claw-free graphs.

1. Introduction

A hypergraph $H = (V, E)$ [2] is a generalization of the notion of graph consisting in a set of vertices $V = \{v_1, \ldots, v_n\}$ and a set of hyperedges $E \subset \mathcal{P}(V) \setminus \{\emptyset\}$ such that $e \not\subset e'$ for any pair e, e' of E. In the case in which $|e| = 1$ for all $e \in E$, we say that H is trivial. Similarly to the graph case, the degree of a vertex $v \in V$ is the number of hyperedges $e \in E$ such that v belongs to e. A hypergraph H is k-uniform if $|e| = k$ for all hyperedge $e \in E$. In this paper, we suppose that H has no parallel hyperedges, i.e., $e \neq e'$ for any pair e, e' of hyperedges.

Our study focuses on the recognition of intersection graphs of hypergraphs. In particular, given a k-uniform hypergraph H, its l-intersection graph $G = L_k^l(H)$, $1 \leq l < k$ is $G = (E, F)$ where the vertex set is E, and $ee' \in F$ if and only if $|e \cap e'| = l$ (i.e. two hyperedges of H intersect in exactly l elements). Note that, even if a similar definition could be given for general hypergraphs, here we focus only on the simpler uniform case.

Some previous results are present in the literature. In particular, in [10], Hliněný and Kratochvíl proved that deciding whether a graph G belongs to L_3^1 is NP-complete. On the other hand, using the characterization of line graphs by Beineke [1], the problem of deciding whether G belongs in L_2^1 is polynomial.

In [6,7] Di Marco et al. studied the null label problem on hypergraphs and they proved that it is $NP-$complete to decide whether a graph belongs to L_3^2.

Relying on the latter result, we are interested in studying some subclasses of L_3^2 where the problem turns out to be polynomially solvable. One can observe that a relevant property of the elements of L_3^2 is that they are $K_{1,4}$-free. So, it deserves interest to strengthen this property and focusing on the $K_{1,3}$-free subset of L_3^2, say claw-free L_3^2 graphs. Here, we

* Corresponding author.
E-mail addresses: niccolo.dimarco@uniroma1.it (N. Di Marco), andrea.frosini@unifi.it (A. Frosini), chp@cnam.fr (C. Picouleau).

https://doi.org/10.1016/j.dam.2024.05.009

show that deciding whether a claw-free graph belongs to L_3^2 remains a NP-complete problem, but, interestingly, it turns into a polynomial task if we further narrow the subclass considering only its triangulated graphs.

The paper is organized as follows: In Section 2 we give the basic notations and definitions of graph theory that we use in the sequel, focusing on the properties of the class L_3^2. In Section 3 we study the membership problem of the subclasses of claw-free and triangulated L_3^2 graphs. Finally, in Section 4, we provide some further properties and complexity results on graphs in L_k^l.

2. Definitions and notations

In this section, we provide some basic definitions used throughout the paper. The reader is referred to [5] for further definitions and main results on graph theory.

2.1. (Hyper)graphs notions

Let us first consider a simple undirected graph $G = (V, E)$, $|V| = n, |E| = m$. For $v \in V$, we define its open neighbourhood as $N(v) = \{w \in V \mid (v, w) \in E\}$. The degree of $v \in V$ is $d_G(v) = |N(v)|$ or simply $d(v)$ when the context is unambiguous. We denote with $\Delta(G)$, $\delta(G)$ the maximal, respectively minimal, degree of a vertex. A vertex v is a *leaf* if $d(v) = 1$. G is k-regular when $d(v) = k$ for any $v \in V$.

For $S \subseteq V$, let $G[S]$ denote the subgraph of G *induced* by S, which has vertex set S and edge set $\{uv \in E \mid u, v \in S\}$. For $v \in V$, we write $G - v = G[V \setminus \{v\}]$. Similarly, for $S \subsetneq V$ and $v \in V \setminus S$ we write $G[S] + v = G[S \cup \{v\}]$. For $e \in E$, we write $G - e = (V, E \setminus \{e\})$.

For $k \geq 1$, $P_k = (u_1, u_2, \ldots, u_k)$ is a *chordless path* if no two vertices are connected by an edge that is not in P_k, i.e. if $V_{P_k} = \{u_1, \ldots, u_k\}$ then $G(V_{P_k}) = P_k$. Similarly, for $k \geq 3$, it is possible to define a chordless cycle (or induced cycle) $C_k = (u_1, u_2, \ldots, u_k, u_1)$. It is common to call a *hole* a chordless cycle having $k \geq 4$ while a graph without a hole is *chordal* or, equivalently, *triangulated*.

We say that $S \subseteq V$ is called a *clique* if $G[S]$ is a *complete graph*, i.e., every pairwise distinct vertices $u, v \in S$ are adjacent. We denote with K_p the clique on p vertices and we say that $C_3 = K_3$ is a *triangle*. $K_{1,p}$ is the star on $p+1$ vertices, that is, the graph with vertices $\{u, v_1, v_2 \ldots, v_p\}$ and edges $uv_1, uv_2, \ldots, uv_p$. The graph obtained when $p = 3$ (i.e. $K_{1,3}$) is known as *claw* graph.

For $S \subset V$ the clique $G[S]$ is *maximal* if for any $v \in V \setminus S$ then $G[S] + v$ is not a clique. If not differently specified, in this work we always consider maximal cliques. In this paper a clique K is *small, medium, big* when $|K| \leq 2, 3 \leq |K| \leq 4, |K| \geq 5$, respectively. Given two cliques c, c', we denote $c \cap c'$ as the set of vertices belonging to both c and c' (i.e. the intersection of the two cliques).

A *cut-edge* in a connected graph G is an edge $e \in E$ such that $G - e$ is not connected.

For a fixed graph H we write $H \subseteq G$ whenever G contains an induced subgraph isomorphic to H. Instead, G is *H-free* if it has no induced subgraph isomorphic to H.

We now consider a general hypergraph $H = (V, E)$. Given a pair of vertices $u, v \in V$, their multiplicity is the number of edges in E containing both u and v. We denote with $m(H)$ the maximum multiplicity among all pairs of vertices. From this notion follows the notion of linear hypergraphs, which is defined as a hypergraph in which $m(H) = 1$.

In general, we define L_k^l as the class of graphs G that are the $l-$intersection graph of a k-uniform hypergraph H, i.e. $G = L_k^l(H)$. In such a case, we say that H is a preimage of G (note that a preimage is not necessarily unique). As an example, L_2^1 corresponds to the class of classic line graphs.

Since we are interested in detecting if a graph has a preimage, we assign to each vertex a specific label. In particular, for $G = L_k^l(H)$ a λ_k^l-labelling is a labelling of its vertices such that the label of each vertex is a k-set, i.e. a hyperedge of a $k-$uniform hypergraph (see Fig. 2 for an example of a λ_3^2-labelling of a graph).

2.2. Properties of graphs in L_3^2

When dealing with recognizing intersection graphs, a crucial role is played by the maximal cliques, since they may represent edges sharing a subset of vertices. For such reason, provided $G \in L_3^2$, we start giving some properties involving its maximal cliques.

Let $T = K_3$ be a maximal triangular clique in G with vertices a, b, c. Two hyperedges configurations can be detected in the preimage 3-uniform hypergraph (from now on numbers indicate the vertices' indexes in the hypergraph): either $a = \{1, 2, x\}, b = \{1, 2, y\}, c = \{1, 2, z\}, x \neq y \neq z$ or $a = \{1, 2, 3\}, b = \{1, 2, 4\}, c = \{1, 3, 4\}$. The first case is defined as *positive clique* and the second as *negative clique*.

A similar situation occurs with K_4. In fact, if we consider a clique with four vertices a, b, c, d, then again two cases appear: either $a = \{1, 2, x\}, b = \{1, 2, y\}, c = \{1, 2, z\}, d = \{1, 2, t\}, x \neq y \neq z \neq t$ or $a = \{1, 2, 3\}, b = \{1, 2, 4\}, c = \{1, 3, 4\}, d = \{2, 3, 4\}$. The first case is denoted as *positive clique* and the second *negative clique*.

For bigger cliques, the situation is easier. Let K_p, with $p \geq 5$, the clique with p vertices $a_1, \ldots, a_p$. Then it must be $a_i = \{1, 2, x_i\}$, $1 \leq i \leq p$. We indicate all these cases as *positive*. In general, we are referring to positive cliques whenever their labels are composed of only one sharing couple. The other cases are referred to as negative.

To set the notation, when a clique K is positive we denote it by K^+ and K^- otherwise. For convenience the clique of two vertices K_2 is both positive and negative. Fig. 1 shows an example of positive and negative cliques.

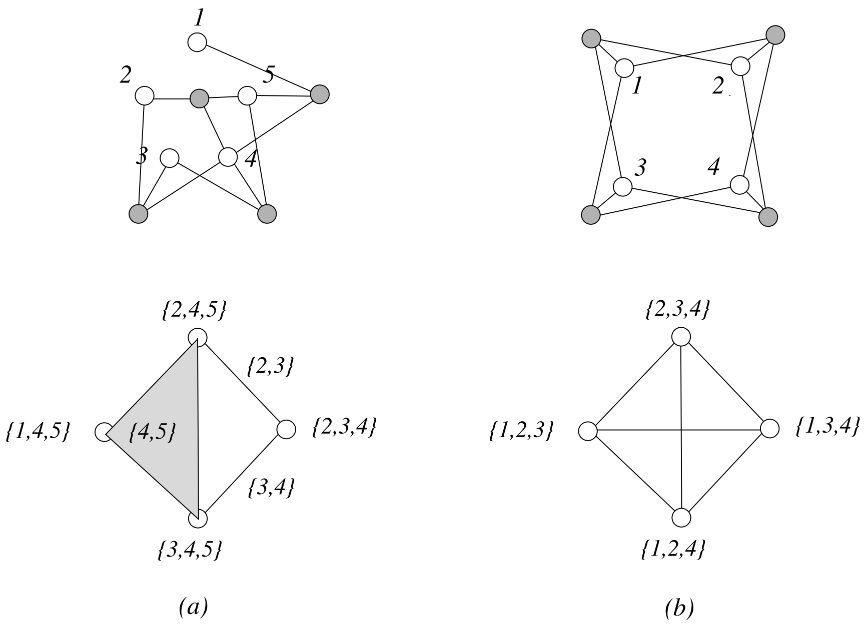

Fig. 1. (a) The shaded triangle is a positive clique since all the edges share $\{4,5\}$. On the other hand, the white triangle and (b) are negative cliques.

Property 2.1. *If K_i, K_j are two cliques of $G \in L_3^2$ such that $|K_i \cap K_j| = 2$. Then, we have K_i^+ and K_j^- (or vice versa).*

Proof. W.l.o.g. let us consider $K_i \cap K_j = \{u, v\}$ with $u = \{1, 2, 3\}$, $v = \{1, 2, 4\}$ and $s \in K_i \setminus K_j$, $t \in K_j \setminus K_i$ be such that $st \notin E$. For contradiction, we assume that K_i^+, K_j^+ or K_i^-, K_j^-. Two cases arises: if K_i^+, K_j^+, then it holds $s = \{1, 2, 5\}$, $t = \{1, 2, 6\}$ that is not possible. Lastly, if K_i^-, K_j^-, then w.l.o.g., it holds $s = \{1, 3, 4\}$, again a contradiction when labelling t. □

A simple consequence of the previous property is the following.

Corollary 2.1. *Let e be an edge of $G \in L_3^2$. Then e is an edge of at most two cliques.*

The following proposition holds.

Property 2.2. *Let K_i, K_j be two distinct cliques of $G \in L_3^2$. Then $|K_i \cap K_j| \leq 2$.*

Proof. Let us proceed by contradiction assuming that $\{x, y, z\} \subseteq K_i \cap K_j$. There exists $v_1 \in K_i \setminus \left(K_i \cap K_j\right)$, $v_2 \in K_j \setminus K_i \cap K_j$ such that $v_1 v_2 \notin E$.

If one of the two cliques, say K_i, is positive then $x = \{1, 2, 3\}, y = \{1, 2, 4\}, z = \{1, 2, 5\}$. Thus $v_1 = \{1, 2, 6\}, v_2 = \{1, 2, 7\}$, a contradiction.

Therefore, consider $K_i = K_4^-$. Thus, $x = \{1, 2, 3\}, y = \{1, 2, 4\}, z = \{1, 3, 4\}$ but $v_1 = \{2, 3, 4\}, v_2 = \{2, 3, 4\}$, another contradiction. Finally, the case in which $K_i = K_3^-$ is trivial. □

Based on that, for two (maximal) cliques K_i, K_j we say that they are *strongly intersecting* when $|K_i \cap K_j| = 2$ and they are *weakly intersecting* when $|K_i \cap K_j| = 1$.

3. Recognizing claw-free graphs in L_3^2

In this section, we deal with the recognition of claw-free graphs in L_3^2. We initially show that there exist claw-free graphs in L_3^2 and we provide some necessary conditions for their belonging to that class. Then, we prove that, in general, the recognition problem is *NP*-complete for claw-free graphs in L_3^2.

We define a *realization* of a graph $G = L_k^l(H)$ a λ_k^l-labelling of its vertices such that each of them represents a hyperedge of H. Therefore $G = L_3^2$ if and only it has a realization (see Fig. 2 for an example).

3.1. Claw-free graphs in L_3^2

Fig. 2 shows a realization of a claw-free graph in L_3^2. In general, not all graphs have a realization as witnessed in Fig. 3, that is an immediate consequence of Property 2.2.

We now show some relevant claw-free graphs in L_3^2 that will be used in the proofs of the main results of this section.

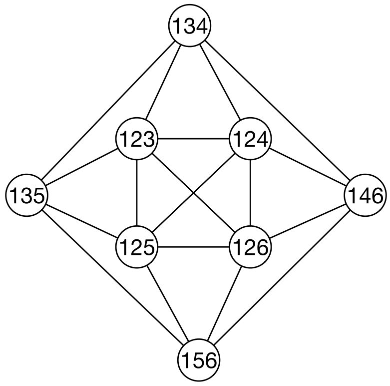

Fig. 2. A graph in L_3^2.

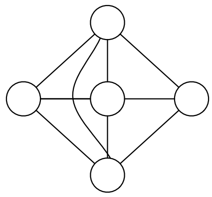

Fig. 3. $K_5 - e$ has no realization.

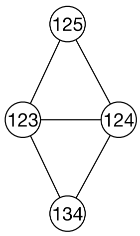

Fig. 4. A realization of the diamond.

Property 3.1. *If $G \in L_3^2$ contains, as an induced subgraph, a diamond D (see Fig. 4) consisting of the two triangles T_1, T_2 then we have either T_1^-, T_2^+ or T_1^+, T_2^-.*

Proof. The result easily follows from Property 2.1. □

Property 3.2. *If $G \in L_3^2$ contains, as an induced subgraph, a 4-wheel W_4 (see Fig. 5) consisting of the four triangles $T_1 = \{a, b, c\}, T_2 = \{a, c, d\}, T_3 = \{a, d, e\}, T_4 = \{a, b, e\}$ then we have either $T_1^-, T_2^+, T_3^-, T_4^+$ or $T_1^+, T_2^-, T_3^+, T_4^-$.*

Proof. The results easily follows from Property 2.1. □

Corollary 3.1. *If $G \in L_3^2$ then it cannot contain the 5-wheel W_5 (see Fig. 6) as an induced subgraph.*

Property 3.3. *If $G \in L_3^2$ contains, as an induced subgraph, a butterfly B (see Fig. 7) consisting of the two triangles T_1, T_2 then we have either T_1^+, T_2^+ or T_1^-, T_2^+ or T_1^+, T_2^-.*

Proof. Let $T_1 = \{a, b, c\}, T_2 = \{a, d, e\}$. Fig. 7 shows two realizations with T_1^+, T_2^+ or T_1^-, T_2^+ or T_1^+, T_2^-. It remains to show that T_1^-, T_2^- is impossible. Let $a = \{1, 2, 3\}, b = \{1, 2, 4\}, c = \{1, 3, 4\}$. Therefore, the only couple that can be used to label d and e is $\{2, 3\}$, which contradicts the hypothesis that T_2 is negative. □

The prism P consists of two vertex disjoint triangles $T_1 = \{a, b, c\}, T_2 = \{d, e, f\}$ plus the three edges ad, be, cf. See Fig. 8 for an example.

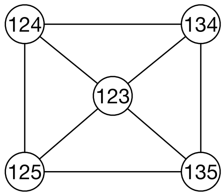

Fig. 5. A realization of the 4-wheel W_4.

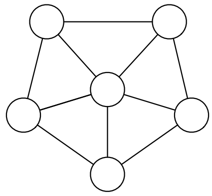

Fig. 6. The 5-wheel W_5.

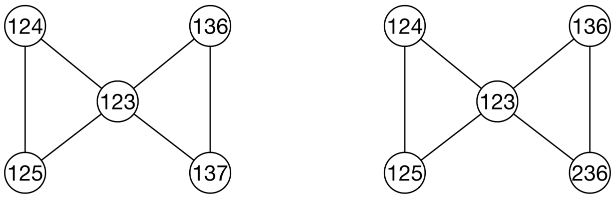

Fig. 7. Two realizations of the butterfly.

Fig. 8. Two realizations of the prism.

Property 3.4. *If $G \in L_3^2$ contains, as an induced subgraph, a prism P then we have T_1^+, T_2^+ or T_1^-, T_2^-.*

Proof. For contradiction, assume that T_1^+, T_2^-. Let $a = \{1, 2, 3\}, b = \{1, 2, 4\}, c = \{1, 2, 5\}$. Without loss of generality $d = \{1, 3, 6\}$. Then we have $e = \{1, 6, 4\}$. It follows that $f = \{1, 5, 6\}$, so T_2 is positive, a contradiction. □

Property 3.5. *If $G \in L_3^2$ contains, as an induced subgraph, a sun S_3 (see Fig. 9) consisting of the four triangles $T_1 = \{a, b, c\}, T_2 = \{a, b, d\}, T_3 = \{a, c, e\}, T_4 = \{b, c, f\}$ then we have $T_1^-, T_2^+, T_3^+, T_4^+$ with $a = \{1, 2, 3\}, b = \{1, 2, 4\}, c = \{1, 3, 4\}, d = \{1, 2, 5\}, e = \{1, 3, 6\}, f = \{1, 4, 7\}$.*

Proof. From Corollary 2.1 we have either $T_1^-, T_2^+, T_3^+, T_4^+$ or $T_1^+, T_2^-, T_3^-, T_4^-$. Fig. 9 shows a realization with $T_1^-, T_2^+, T_3^+, T_4^+$. Now we assume that $T_1^+, T_2^-, T_3^-, T_4^-$ with $a = \{1, 2, 3\}, b = \{1, 2, 4\}, c = \{1, 2, 5\}$. Then, w.l.o.g., $d = \{1, 3, 4\}$. It follows that $e = \{2, 4, 5\}$ but f cannot be labelled. □

Property 3.6. *If $G \in L_3^2$ contains, as an induced subgraph, a path on three vertices (u, v, w) with $u = \{a, b, c\}, w = \{d, e, f\}$ then $\{a, b, c\} \cap \{d, e, f\} \neq \emptyset$.*

As mentioned before, we denote with $K_4 + v$ the graph with $V = \{a, b, c, d, e\}$ where $\{a, b, c, d\}$ is complete and e is connected to exactly one vertex, say a.

Property 3.7. *If $G \in L_3^2$ contains $K_4 + v$, as an induced subgraph, then the clique K_4 of $K_4 + v$ is positive.*

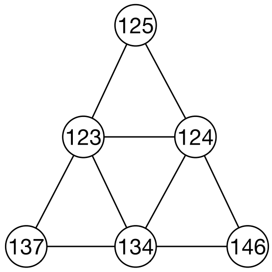

Fig. 9. A realization of the sun S_3.

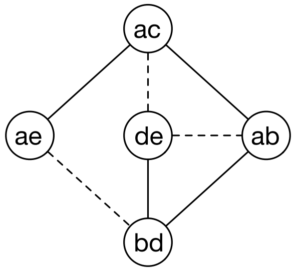

Fig. 10. A graph with a 2-labelling φ. Weak and strong edges are represented by dotted and straight lines, respectively, while the nodes show the pair of elements associated by φ.

Proof. For contradiction, assume that the clique on four vertices $G[\{a, b, c, d\}]$ is negative: $a = \{1, 2, 3\}$, $b = \{1, 2, 4\}$, $c = \{1, 3, 4\}$, $d = \{2, 3, 4\}$. Then, without loss of generality, $e = \{1, 2, 5\}$. So $be \in E$, a contradiction. □

Note that all the graphs we have considered in this subsection are claw-free.

3.2. Recognition for claw-free graphs in L_3^2

In [7], the authors proved that recognizing whether a graph G is in L_3^2 is NP-complete. Since a graph in L_3^2 is $K_{1,4}$-free, we are concerned with the subclass of claw-free graphs ($K_{1,3}$-free graphs) and we will prove that also in this case the membership problem to L_3^2 remains NP-complete. To reach our goal we need an intermediate problem that is defined and proved NP-complete below.

The 2-labelling intersection (2LI) problem

Let us consider a simple graph $G = (V, E)$ and a partition of its edge-set E into two subsets E_w and E_s, i.e. $E = E_w \cup E_s$ and $E_w \cap E_s = \emptyset$. We call *weak edges* the edges in E_w and *strong edges* those in E_s.

We define a function, say a 2-*labelling* φ, that associates to each vertex $v \in V$ a pair of labels $\{a_v, b_v\}$ such that:

(i) if $v_1 \neq v_2$, then $\varphi(v_1) \neq \varphi(v_2)$;

(ii) if $v_1v_2 \in E_w$, then $\varphi(v_1) \cap \varphi(v_2) = \emptyset$;

(iii) if $v_1v_2 \in E_s$, then $|\varphi(v_1) \cap \varphi(v_2)| = 1$.

See Fig. 10 for an example.

The 2LI problem, provided in its decision form, follows

2-LABELLING INTERSECTION (2LI) *Instance:* a simple graph $G = (V, E)$ and a partition of its edges into E_w and E_s. *Question:* does G admit φ a 2-labelling of its vertices?

We show the NP-completeness of 2LI by a reduction that involves the problem 3-SAT (LO2 in [8])

3-SAT *Instance:* a set U of variables, a collection C of clauses over U such that each clause $c \in C$ has $\lvert c\rvert = 3$. *Question:* Is there a satisfying truth assignment for C?

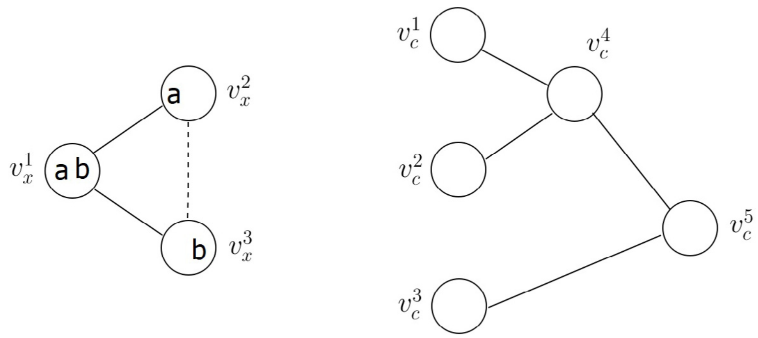

Fig. 11. (a) the graph defined for a variable $x \in U$; (b) the graph defined for a clause $c \in C$. In (a) the nodes show the pair of elements associated by φ.

Given an instance A of 3-SAT, we construct a graph $G_A = (V_A, E_A)$ and a partition of its edges into weak and strong edges $E_{A,w}$ and $E_{A,s}$ such that the 3-SAT instance admits a solution if and only if G_A admits a 2-labelling. This will imply the NP-completeness of 2LI.

Hence, we start by providing two graphs' prototypes to model variables and clauses of the 3-SAT instance A, then we show how to use them to reach the graph G_A.

Representing the variables and the clauses of A

Let us define the graph $G_x = (V_x, E_x)$ that will be used to represent each variable $x \in U$. The set V_x consists of three vertices v_1^x, v_2^x, and v_3^x, while E_x is partitioned into the weak edges $E_{x,w} = \{v_2^x v_3^x\}$ and the strong edges $E_{x,s} = \{v_1^x v_2^x, v_1^x v_3^x\}$ (see Fig. 11, (a)).

Lemma 3.2. *Let G_x be the graph related to variable x and $\varphi(v_1^x) = \{a, b\}$. It holds, w.l.o.g., that $a \in \varphi(v_2^x)$ and $b \in \varphi(v_3^x)$.*

The proof is immediate by definition of weak and strong edges.

The gadget G_x will also be used to represent the dichotomy of the truth values in the final graph. In particular, from now on, we consider the truth values labels T and F. We define G_{TF} similarly to G_x with the further assumption that $a = T$ and $b = F$.

On the other hand, we associate each clause $c \in C$ with a graph $G_c = (V_c, E_c)$ having five vertices $V_c = \{v_c^1, v_c^2, v_c^3, v_c^4, v_c^5\}$ and four strong connections, i.e., $E_c = E_{c,s} = \{v_c^1 v_c^4, v_c^2 v_c^4, v_c^3 v_c^5, v_c^4 v_c^5\}$ (see Fig. 11, (b)).

Putting things together

So, starting from an instance A of 3-SAT, we proceed in defining the graph $G_A = (V_A, E_A)$. The reader can follow the construction in Fig. 12. We include in G_A a number $n = |U|$ of graphs $G_{x_1}, \dots, G_{x_n}$ representing the variables of U, a number $m = |C|$ of graphs $G_{c_1}, \dots, G_{c_m}$ representing the clauses of C, and the graph G_{TF}.

The connections between these graphs in G_A are set according to the following rules:

(1) the variables are connected by all the possible weak edges between the vertices v_x^1, i.e., for each couple of variable x and y in U, we set the weak edge $v_x^1 v_y^1 \in E_{A,w}$;

(2) let x be the ith variable involved in the clause c, with $1 \le i \le 3$. We set the weak edge $v_x^1 v_c^i \in E_{A,w}$, and the strong edge $v_x^3 v_c^i \in E_{A,s}$ if x is negated, $v_x^2 v_c^i \in E_{A,s}$ otherwise;

(3) for each variable x, we set the strong edges $v_x^2 v_{TF}^1, v_x^3 v_{TF}^1 \in E_{A,s}$ to connect the variable to the truth values in G_{TF}. Furthermore, we set three more strong edges $v_c^4 v_{TF}^1, v_c^5 v_{TF}^1, v_c^5 v_{TF}^2 \in E_{A,s}$ to connect also each clause c to the truth values in G_{TF}.

Theorem 3.3. *Given an instance A of 3-SAT, the graph G_A admits a 2-labelling if and only if the instance A has a solution.*

Proof. Suppose that a 2-labelling φ exists. Let $\{a_i, b_i\}$ be the label associated to the node $v_{x_i}^1$ of G_{x_i}. The connections in (1) assure that the labels $\{a_i, b_i\}$, $1 \le i \le n$, have no common elements.

By the edges in (2), for each variable x, one among v_x^1 and v_x^2 contains T, while F belongs to the other since their labels cannot intersect by definition of G_x. Let us consider a clause c involving literals of the (distinct) variables x, y and z. $\varphi(v_c^1)$ contains, by the edges in (2), the truth value either in $\varphi(v_x^2)$ or in $\varphi(v_x^3)$ of G_x to whom it is strongly connected. Note that $\varphi(v_c^1)$ and $\varphi(v_x^1)$ does not intersect since a weak edge is set between them in (2).

Now, consider the node v_c^4: it is strongly connected both with v_{TF}^1, v_c^1, and v_c^2, so its label contains T or F according to the values of $\varphi(v_c^1)$ and $\varphi(v_c^2)$. More precisely, the following three cases arise:

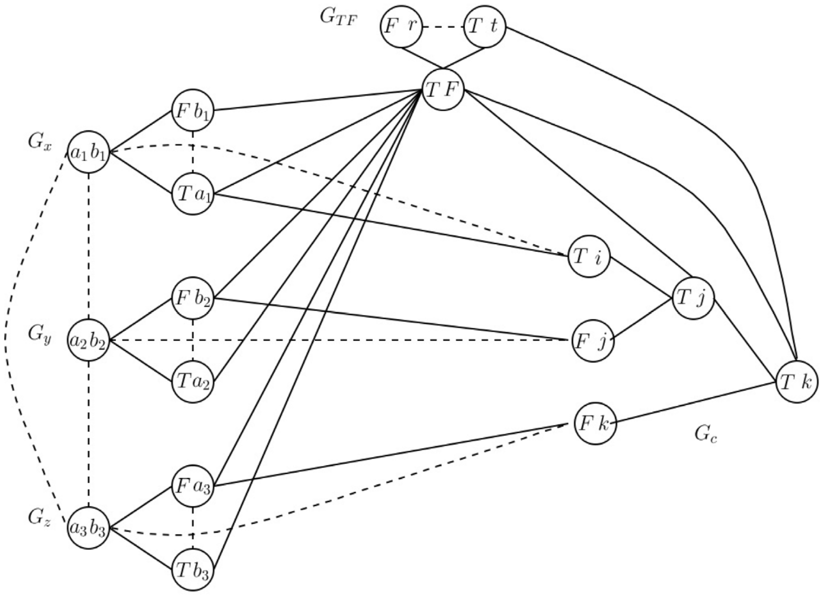

Fig. 12. The graph G_A for a clause $C = \bar{x} \vee y \vee z$.

- $T \in \varphi(v_c^1)$ and $T \in \varphi(v_c^2)$: it follows that $T \in \varphi(v_c^4)$;
- $T \in \varphi(v_c^1)$ and $F \in \varphi(v_c^2)$, or conversely: it follows that one among T or F, but not both, belongs to $\varphi(v_c^4)$;
- $F \in \varphi(v_c^1)$ and $F \in \varphi(v_c^2)$: it follows that $F \in \varphi(v_c^4)$.

Finally, consider the node v_c^5: since it is strongly connected to v_{TF}^1 and v_{TF}^2, its label contains, w.l.o.g. T. By the three cases above, if $T \in \varphi(v_c^5)$, then it holds $T \in \varphi(v_c^3)$ or $T \in \varphi(v_c^4)$. So, $F \in \varphi(v_c^1)$, $F \in \varphi(v_c^2)$, and $F \in \varphi(v_c^3)$ if and only if G_c does not admit a 2-labelling, and the same holds for G_A. Since the three truth values in $\varphi(v_c^1)$, $\varphi(v_c^2)$, and $\varphi(v_c^3)$ are the truth values of the literals in c, then a truth assignment for c exists if and only if a 2-labelling for G_c does.

The converse holds as the construction is reversible. □

Theorem 3.4. *Let $G = (V, E)$ be a claw-free graph. Deciding whether there exists a 3-uniform hypergraph H such that $G = L_3^2(H)$ is a NP-complete problem.*

Proof. The proof has two parts. Firstly, we define $\mathcal{C}$ a subclass of claw-free graphs we are interested in here. Then we show that the problem of deciding whether $G \in \mathcal{C}$ is such that $G \in L_3^2$ is equivalent to the problem 2*LI* which is *NP*-complete from Theorem 3.3.

First: the definition of $\mathcal{C}$. A graph $G \in \mathcal{C}$ consists of components $C_1, \ldots, C_k$ each of the C_i's being a clique of size at least five, the C_i's form a partition of the vertex set of G. When two components C_i, C_j are linked there are connected by either a *strong link* or a *weak link*. A strong link consists of a C_4 of G with its two non-adjacent vertices $i_1, i_2 \in C_i$ and its two other non-adjacent vertices $j_1, j_2 \in C_j$. A weak link consists of a K_4 of G with two vertices $i_1, i_2 \in C_i$ and the two other vertices $j_1, j_2 \in C_j$. The links, weak or strong, have no common vertices. It follows that G is claw-free. Moreover, since $|C_i| \geq 5$, $C_i \cup C_j$ the union of two distinct components cannot be a clique.

When $G \in L_3^2$ the components satisfy the following: Since each component $C_i = \{v_1^i, v_2^i, \ldots, v_p^i\}$ has at least five vertices, we necessarily have $v_1^i = \{i, i', 1\}$, $v_2^i = \{i, i', 2\}, \ldots, v_p^i = \{i, i', p\}$ and C_i is associated with its pair of common labels $\{i, i'\}$. For two distinct components C_i, C_j we have $|\{i, i'\} \cap \{j, j'\}| \leq 1$. When C_i, C_j are connected with a strong link then we have $|\{i, i'\} \cap \{j, j'\}| = 1$. When C_i, C_j are connected with a weak link then we have $\{i, i'\} \cap \{j, j'\} = \emptyset$.

Second: equivalence with the problem 2*LI*. Given $G \in \mathcal{C}$ we define the graph G' as follows: to the vertices v_i of G' correspond the components C_i of G, and vice versa; to a strong (resp. weak) link of G corresponds a strong (resp. weak) edge of G', and vice versa.

We assume that there exists a 3-uniform hypergraph H such that $G = L_3^2(H)$. Since $|C_i| \geq 5$ the intersection of the labels of the pairs of vertices in the same component C_i is the same two labels says $\{i, i'\}$. Now, for two distinct C_i, C_j, since $C_i \cup C_j$ is not a clique we have that $|\{i, i'\} \cap \{j, j'\}| \leq 1$. When C_i, C_j are strongly connected then $|\{i, i'\} \cap \{j, j'\}| = 1$, when they are weakly connected then $|\{i, i'\} \cap \{j, j'\}| = 0$. Thus for each vertex v_i of G' when assigning the two labels i, i to v_i we obtain a positive answer for the problem 2*LI*.

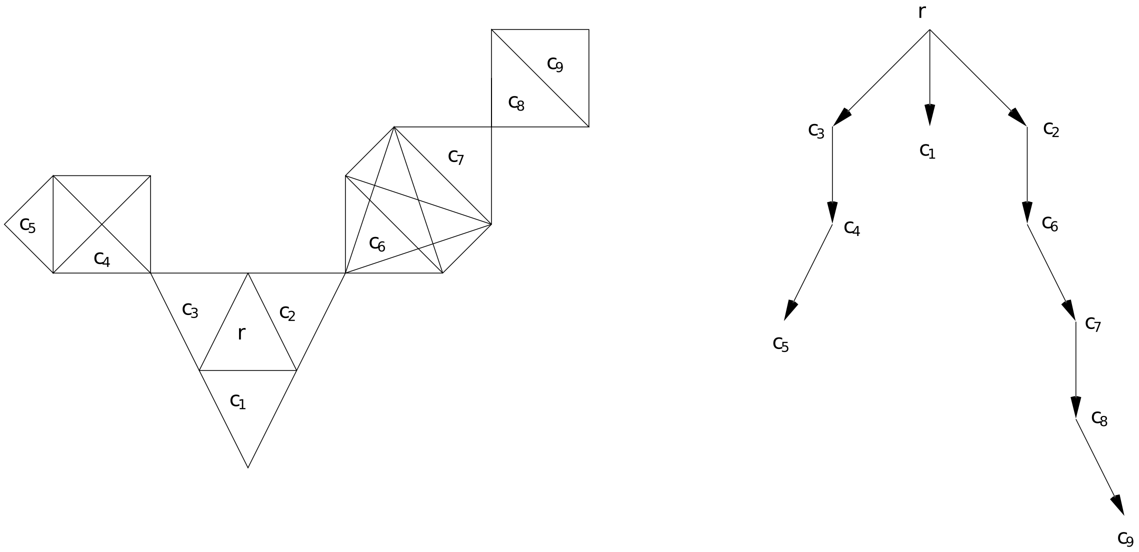

Fig. 13. Example of a claw-free triangulated graph and its associated directed tree. We consider the clique r as the root of the tree. The three leaves are highlighted.

Now, we assume that the problem $2LI$ has a positive answer. Let i, i' be the two labels assigned to v_i in G'. We assign $\{i, i'\}$ to the component C_i of G. Let v_i, v_j be two vertices linked with a strong edge. Then the labels associated to C_i, C_j are $\{i, i'\}, \{i, j\}$, respectively. Let w_1^i, w_2^i and w_1^j, w_2^j be respectively the two vertices of the strong link between C_i and C_j. Then $w_1^i = \{i, i', a\}, w_2^i = \{i, i', b\}, w_1^j = \{i, j, a\}, w_2^j = \{i, j, b\}$. Let v_i, v_j be two vertices linked with a weak edge. Then the labels associated to C_i, C_j are $\{i, i'\}, \{j, j'\}$, respectively. Let w_1^i, w_2^i and w_1^j, w_2^j be respectively the two vertices of the weak link between C_i and C_j. Then $w_1^i = \{i, i', j\}, w_2^i = \{i, i', j'\}, w_1^j = \{j, j', i\}, w_2^j = \{j, j', i'\}$. Thereafter, when a vertex $w_k^i \in C_i$ is not contained in a link, weak or strong, we take $w_k^i = \{i, i', k\}$. Thus there exists H a 3-uniform hypergraph H such that $G = L_3^2(H)$. □

3.3. Recognition for triangulated claw-free graphs in L_3^2

Recall that a graph G is *triangulated* (or *chordal*) if it is C_k-free, $k \geq 4$.

It is known that to each triangulated graph $G = (V, E)$, we can associate, in linear time [3], a maximal clique tree $T = (C, S)$ where each maximal clique of G corresponds to a vertex $c \in C$ and $cc' \in S$ if $c \cap c' \neq \emptyset$ and $c \cap c'$ is a minimal separator of G which is a clique. An example is shown in Fig. 13

From Property 2.2 if $G \in L_3^2$, then an edge $cc' \in S$ corresponds either to a strong intersection when $|c \cap c'| = 2$ or to a weak intersection when $|c \cap c'| = 1$. Moreover, when $cc' \in S, |c|, |c'| \geq 3$ we can easily obtain the following:

- c and c' weakly intersect: let $c \cap c' = \{a\}$; there exists $\{u, v\} \in c, \{u', v'\} \in c'$ such that $G[\{a, u, v, u', v'\}]$ is a butterfly, otherwise a cannot be a separator of G;
- c and c' strongly intersect: let $c \cap c' = \{a, b\}$; there exists $u \in c, u', \in c'$ such that $G[\{a, b, u, v\}]$ is a diamond, otherwise $\{a, b\}$ cannot be a separator of G.

In the sequel, differently from what indicated before, we denote as K_i the generic cliques, without referring to the index i as its cardinality, unless otherwise specified. The following Proposition holds.

Property 3.8. *Let $G = (V, E)$ be a triangulated claw-free graph and K_t be a clique of G such that $t \geq 4$. If K_i, K_j are two (distinct) cliques that strongly intersect K_t then $K_i \cap K_j \cap K_t = \emptyset$.*

Proof. For contradiction, we assume $v \in K_i \cap K_j \cap K_t$. Let $K_i \cap K_t = \{v, v_i\}, K_j \cap K_t = \{v, v_j\}, v_i \neq v_j$. Since $t \geq 4$ there exists $k \in K_t \setminus \{v, v_i, v_j\}$. Let $w_i \in K_i \setminus \{K_j \cup K_t$ and $w_j \in K_j \setminus \{K_i \cup K_t\}$. We have $w_i w_j \in E$, otherwise $G[\{v, k, w_i, w_j\}]$ is a claw. We also have $v_i v_j \in E$ for a similar reason. But then $G[\{v_i, w_i, w_j, v_j\}] = C_4$ which is not possible since G is triangulated. □

Fig. 14 shows the situation described in Property 3.8.

The following theorem states the polynomiality of the reconstruction of a 3-uniform hypergraph from a triangulated claw-free graphs G such that $G = L_3^2(H)$. To help the reader, the proof is divided into small steps that lead to the final result.

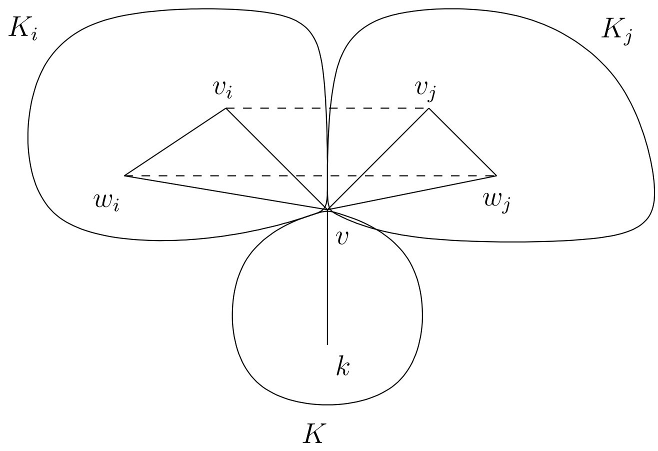

Fig. 14. Visual example of the proof of Property 3.8. Curves represent nodes inside the same clique. Dashed edges are forced by claw-free property.

Theorem 3.5. *Let $G = (V, E)$ be a triangulated claw-free graph. Deciding whether a 3-uniform hypergraph H exists such that $G = L_3^2(H)$ and constructing H when it exists can be done in polynomial time.*

Proof. Let $G = (V, E)$ be a triangulated claw-free graph. We can assume that G is connected and $|V| \geq 3$. We first consider two cases. *G has a cut-edge*

Recall that $G[S]$ denote the subgraph of G induced by $S \subseteq V$. Let $e = v_1v_2 \in E$ be a cut-edge of G. We call $\widetilde{G}_1, \widetilde{G}_2$ the two components of $\widetilde{G} = G - e$, with $v_1 \in \widetilde{G}_1, v_2 \in \widetilde{G}_2$.

We will denote $G_1 = \widetilde{G}_1 + v_2$ and $G_2 = \widetilde{G}_2 + v_1$.

Since G is claw-free, v_1 is a vertex of at most two cliques: the clique $G[\{v_1, v_2\}]$ and one clique K_n^1 of $\widetilde{G}_1$. The same holds for v_2, which can belong respectively to one clique K_m^2 of $\widetilde{G}_2$ or to $G[\{v_1, v_2\}]$.

First, we suppose that v_1 is a leaf of G (i.e. $deg_G(v_1) = 1$). In such a case, $G = G_2$.

Property 3.9. *G has a realization if and only if $\widetilde{G}_2$ has a realization such that K_m^2 is positive when $m = 4$.*

Proof. From Property 3.7 if G has a realization then K_m^2 is positive when $m = 4$.

Conversely, assume that $\widetilde{G}_2$ such that K_m^2 is positive when $m = 4$. The following cases arise:

- if $m = 4$ suppose that $K_m^2 = \{v_2, v_3, v_4, v_5\}$ with $v_2 = \{1, 2, 3\}$, $v_3 = \{1, 2, 4\}$, $v_4 = \{1, 2, 5\}$, $v_5 = \{1, 2, 6\}$. Then we can take $v_1 = \{1, 3, 0\}$;
- if $m \geq 5$ then K_m^2 is positive and we do as before.
- if $m = 3$ we have two further cases. If K_m^2 is positive, we do as above. If instead it is negative, let $K_m^2 = \{v_2, v_3, v_4\}$ with $v_2 = \{1, 2, 3\}$, $v_3 = \{1, 2, 4\}$, $v_4 = \{1, 3, 4\}$. Then we take $v_1 = \{2, 3, 0\}$;
- Finally $K_m^2 = \{v_2, v_3\}$ with $v_2 = \{1, 2, 3\}$, $v_3 = \{1, 2, 4\}$. We take $v_1 = \{1, 3, 0\}$. □

We now suppose both v_1 and v_2 are not leaves.

Property 3.10. *G has a realization if and only if G_1 and G_2 have a realization.*

Proof. Obviously, if G has a realization also G_1 and G_2 have it. Therefore we focus on sufficiency.

The following cases arise:

- suppose that K_n^1 and K_m^2 are positive and that the vertices of K_n^1 have the labels $\{1, 2, a_k\}, 1 \leq k \leq n$ with $v_1 = \{1, 2, a_1\}$, while the vertices of K_m^2 have the labels $\{2, 3, b_k\}$, $1 \leq k \leq m$ with $v_2 = \{2, 3, b_1\}$. Taking $a_1 = b_1$ and $a_i \neq b_j$ with $1 < i \leq n$ and $1 < j \leq m$, we obtain a realization for G;
- suppose K_n^1 is positive and K_m^2 is negative. Suppose also that K_n^1 vertices have labels $\{1, 2, a_k\}$, $1 \leq k \leq n$ with $v_1 = \{1, 2, a_1\}$. From Property 3.7 we have $m = 3$. W.l.o.g., the vertices of K_m^2 have the labels $\{2, 3, 4\}, \{2, 3, 5\}, \{3, 4, 5\}$ with $v_2 = \{2, 3, 4\}$. Taking $a_1 = 4$ we obtain a realization for G;
- suppose K_n^1 is positive and $m = 2$. Suppose also that the vertices of K_n^1 have labels $\{1, 2, a_k\}$, $1 \leq k \leq n$ with $v_1 = \{1, 2, 3\}$. W.l.o.g., for the realization of G_1 we have $v_2 = \{1, 3, 4\}$, then up to a renaming of the labels for the realization of G_2' we have a realization for G;
- suppose K_n^1 and K_m^2 are negative. From Property 3.7 we have $n = m = 3$. For the realization of G_1 let $v_1 = \{1, 2, 3\}$, $v_2 = \{2, 3, 5\}$ and let $\{1, 2, 4\}, \{1, 3, 4\}$ be the labels of the two other vertices of K_n^1. Then up to a renaming

of the labels for the realization of G'_2 with $\{3, 5, 6\}, \{2, 5, 6\}$ the labels of the two other vertices of K^2_m, we have a realization for G;
- Suppose $n = m = 2$. For the realization of G_1 let $v_1 = \{1, 2, 3\}$, $v_2 = \{1, 3, 5\}$ and let $\{1, 2, 4\}$ be the label of the other vertex of K^1_n. Up to a renaming of the labels for the realization of G'_2 we have a realization for G. □

G has no cut-edge:
If G has no cut-edge, then the following Proposition is readily obtained.

Property 3.11. *G has no small cliques.*

Proof. For contradiction we assume that $K = G[\{v_i, v_j\}]$ is a clique with $v_i \in K_i$, $v_j \in K_j$, where $K_i \neq K$, $K_j \neq K$, are two distinct cliques. Since v_iv_j is not a cut-edge there exists a path $(v_i, v_k, \dots, v_j)$ where $v_k \neq v_j$. Assume that it is one of the shortest paths. Then $(v_i, v_k, \dots, v_j, v_i)$ is an induced cycle of length greater than three, a contradiction. □

Therefore, we suppose without loss of generality that G contains only medium or big cliques.
Using the algorithm given in [3] we obtain $T = (C, S)$, a maximal clique tree of G in time $O(n)$ where a vertex $c \in C$ corresponds to a big or a medium clique of G. We show how to construct a representation for it.
First, we need the following proposition.

Property 3.12. *Let $c \in C$ with $|c| = 3$. Then, c has at most 3 neighbours in T.*

Proof. Suppose that c has more than 3 neighbours. Then, considering our previous results, only two cases are possible: at least two cliques weakly intersect c in the same node v or one strongly intersects c in $\{v, w\}$ and the other weakly intersects c, without loss of generality, in $\{v\}$. We consider these cases separately:

- assume that c has two neighbours c_1, c_2 that weakly intersect it in $v \in c$. Let $w \in c, w \neq v$. Since the cliques are distinct and of size greater than one, there must exist $v_i \in c_i, v_i \neq v, i \in \{1, 2\}$. Remember that, if c_1, c_2 are neighbours of c, it means that $\{v\}$ is a separator in G. Therefore, $wv_1, wv_2, v_1v_2 \notin E$, but $G[\{v, w, v_1, v_2\}] = K_{1,3}$, a contradiction;
- we assume that c has two neighbours c_1, c_2 such that $c_1 \cap c = \{v, v'\}$ and $c_2 \cap c = \{v\}$. Let $w \in c, w \neq v, v'$. Since $\{v, v'\}$ and $\{v\}$ are two separators $wv_1, wv_2, v_1v_2 \notin E$, but $G[\{v, w, v_1, v_2\}] = K_{1,3}$, a contradiction.

Thus c has at most three neighbours. □

Note that for the case where c has three neighbours, either c is the central triangle of the sun or it weakly intersects three cliques c_1, c_2, c_3 in each of its vertices.
Returning to the main problem, we will associate a label, $\lambda(c) \in \{+, -\}$ to each vertex c of T. The idea is that the label associated with a clique represents its negativity or positivity.
In the first stage, partial labelling is obtained with the Algorithm 1. In this step only the *local* constraints based on big cliques, Properties 3.7 and 3.5 are taken into account. The last instruction of the algorithm consists of checking Property 3.3.
Then, in the second stage, Algorithm 2, Labelling, propagates, from the bottom to the top, the labelling in the neighbours of the cliques already labelled. The last instruction of the algorithm consists of checking Property 3.3. Note that we are not considering the case in which a clique does not have a label, although that may happen. We consider this case in the last stage.
In fact, when some vertices are not labelled, we can fix their labels to be either + or − in such a way that the labels alternate for the cliques c, c' that strongly intersect. Note that the labelling of T_r terminates in time $O(n)$.
Then we find a 3-uniform hypergraph H such that $G = L^2_3(H)$.
Construction of the 3-uniform hypergraph

Property 3.13. *Consider the labelled tree T of a claw-free triangulated hypergraph. Then there exists H such that $G = L^2_3(H)$.*

Proof. Given T with the labelling of its vertices, we construct a labelling of the vertices of G from the root c_r to the leaves of T_r. We assume w.l.o.g. that $\lambda(c_r) = +$. We label the vertices of c_r as follows: since $\lambda(c_r) = +$, c_r it is labelled positively with the labels $(1, 2, 3), (1, 2, 4), \dots, (1, 2, k_r)$. Let $c \in C, c \neq c_r$. We assume that the vertices of its predecessor c' in T_r are labelled.
Since T is a tree, up to a permutation of the labels, we assume that the labels of c' are taken into $\{1, 2, \dots, k\}$. Consider the following two cases:

- Suppose that c' is labelled positively with $(1, 2, 3), (1, 2, 4), \dots, (1, 2, k)$. If c strongly intersect c' then $\lambda(c) = -$. We set $(1, 2, 3), (1, 2, 4)$ the labels of $c \cap c'$. If c contains three vertices, it is labelled as $(1, 3, 4)$. If it contains four vertices, the last node is labelled as $(2, 3, 4)$.
If c weakly intersect c' from Property 3.7 we have $|c| = 3$. Let $(1, 2, 3)$ be the label of $c \cap c'$. If $\lambda(c) = -$ then the other labels of c are $(1, 3, a), (2, 3, a)$, with a different from any value in c and c'. If $\lambda(c) = +$ then the other labels

Algorithm 1 Preprocessing Labelling

for all $c \in T$ **do**
  if $|c| \geq 5$ **then**
    $\lambda(c) = +$ % a big clique is positive
  end if
  if $|c| = 4$ and there exists c' that weakly intersects c **then**
    $\lambda(c) = +$ % by Property 3.7
  end if
  if $|c| = 3$ and there are c_1, c_2, c_3 such that $|c_1| = |c_2| = |c_3| = 3$ and c_1, c_2, c_3 strongly intersects c in different arcs **then**
    $\lambda(c) = +$ % by Property 3.5 noticing that c, c_1, c_2, c_3 induce a sun
    for i=1**to** 3 **do**
      $\lambda(c_i) = -$
    end for
  end if
end for
for all $cc' \in S$ **do**
  if $\lambda(c) = \lambda(c') = -$ **then**
    return $G \notin L_3^2$
  end if % by Property 3.3
end for

of c are $(1, 3, a), (1, 3, b)$, with a, b different from any value in c and c'. In any case, we obtain a valid label for the two cliques;

- Suppose that c' is labelled negatively. Let us consider the case in which $|c'| = 4$ and their vertices are $(1, 2, 3), (1, 2, 4), (1, 3, 4), (2, 3, 4)$: from Property 3.7 c strongly intersect c' and $\lambda(c) = +$ must holds. Let $(1, 2, 3), (1, 2, 4)$ be the labels of $c \cap c'$. The other labels of c are $(1, 2, 5), \ldots, (1, 2, k)$.
  Consider now the case in which $|c'| = 3$ and their vertices are $(1, 2, 3), (1, 2, 4), (1, 3, 4)$. If c strongly intersect c', let $(1, 2, 3), (1, 2, 4)$ be the labels of $c \cap c'$. The other labels of c are $(1, 2, 5), \ldots, (1, 2, k)$.
  On the other hand, if c weakly intersects c', let $(1, 2, 3)$ be the label of $c \cap c'$. The other labels of c are $(2, 3, 5), \ldots, (2, 3, k)$.
  If c' has a second successor c'', such that the label of $c \cap c''$ is $(1, 2, 4)$, then the other labels of c'' are $(2, 4, k+1), \ldots, (2, 4, k+l)$. Recall that $c' \neq c_r$ so c' has at most two successors in T_r. □

To conclude, we estimate the complexity of the whole procedure. When G has no cut edge the procedure takes a time $O(n)$. The algorithm in [12] gives the cut-edges and the biconnected components in time $O(m)$ when G is connected. Hence the complexity of the algorithm is $O(m)$.
The proof is completed. □

Consider the graph depicted G in Fig. 13 and its associated tree cliques. Using the construction in the proof of Property 3.13, we obtain the labelling shown in Fig. 15. From that label, it is easy to find a label of the vertices (i.e. a realization) and conclude that $G \in L_3^2(H)$.

4. Further properties of L_k^l class

In this section, we prove some further results involving graphs belonging to certain L_k^l classes. After giving some properties of the class L_k^1, we move to L_3^2 providing a complexity result for the Hamiltonian cycles detection problem. In particular, Di Marco et al. proved in [7] that Hamiltonian cycles in L_3^2 are useful to assure the existence of a null label in the relative hypergraph. It holds the following

Property 4.1. *If $G \in L_k^l$ then G is $K_{1,p+1}$-free, with $p = \binom{l}{k}$.*

Proof. Let $G \in L_k^l$ and H a preimage of G. Suppose that G contains $K_{1,p+1}$ with a central vertex v and $N(v) = \{e_1 \ldots, e_{p+1}\}$. Then two edges e_i, e_j, $1 \leq i < j \leq p+1$ exit such that $|e_i \cap e_j| \geq p$, so $e_ie_j \in E$, a contradiction. □

4.1. NP-complete problems in the class L_k^1

Given a linear k-uniform hypergraph $H = (V, E)$ one can define a linear $(k+1)$-uniform hypergraph $H' = (V', E')$ as follows. For each hyperedge $e_i \in E$, with $1 \leq i \leq m$, we create the hyperedge $e'_i \in E'$, such that $e'_i = e_i \cup \{a_i\}$. Then we take $V' = V \cup_{1 \leq i \leq m} \{a_i\}$.

Algorithm 2 Labelling

```
choose c_r ∈ C as the root of the directed tree T_r
 % for each c ≠ c_r the unique path c_r − ··· − c is a directed path starting from c_r
for all c ∈ T_r and λ(c) = ∅, from the leaves to c_r do
  Let c′ be the predecessor of c in T_r
  if c is a leaf then
    if c and c′ strongly intersect then
      if λ(c′) = + then
        λ(c) = −
      end if
      if λ(c′) = − then
        λ(c) = +
      end if
    end if
    if c and c′ weakly intersect then
      if λ(c′) = − then
        λ(c) = +  %in this case |c′| = 3 must hold
      end if
      if λ(c′) = + then
        λ(c) = −
      end if
    end if
  end if
  if c is not a leaf then
    if (c has two successors c_1, c_2 that are strongly connected with c and λ(c_1) ≠ λ(c_2)) or (c has a successor c_1 that is
    strongly connected with c, λ(c_1) = + and c has a successor c_2 that is weakly connected with c, λ(c_2) = − ) then
      return G ∉ L_3^2
    end if
    if c has a successor c_1 strongly connected with c then
      if λ(c_1) = + then
        λ(c) = −
      end if
      if λ(c_1) = − then
        λ(c) = +
      end if
    end if
    if c has a successor c_1 weakly connected with c and λ(c_1) = − then
      λ(c) = +
    end if
    if λ(c) = λ(c′) = − then
      return G ∉ L_3^2
    end if
  end if
end for
```

Using this construction and some NP-complete results, we can obtain the following properties.

Property 4.2. *For any fixed $k \geq 3$, deciding whether a graph is n L_k^1 is NP-complete.*

Proof. We know that recognizing whether a graph $G \in L_3^1$ is *NP*-complete [10]. Using the previous construction we obtain the result. □

Property 4.3. *For any fixed $k \geq 2$, the problems Hamiltonian cycle, 3-coloring and Minimum domination are NP-complete in L_k^1.*

Proof. The problems Hamiltonian cycle [4], 3-coloring [11], Minimum domination [13] are NP-complete in L_2^1. Using the previous construction, we obtain the result. □

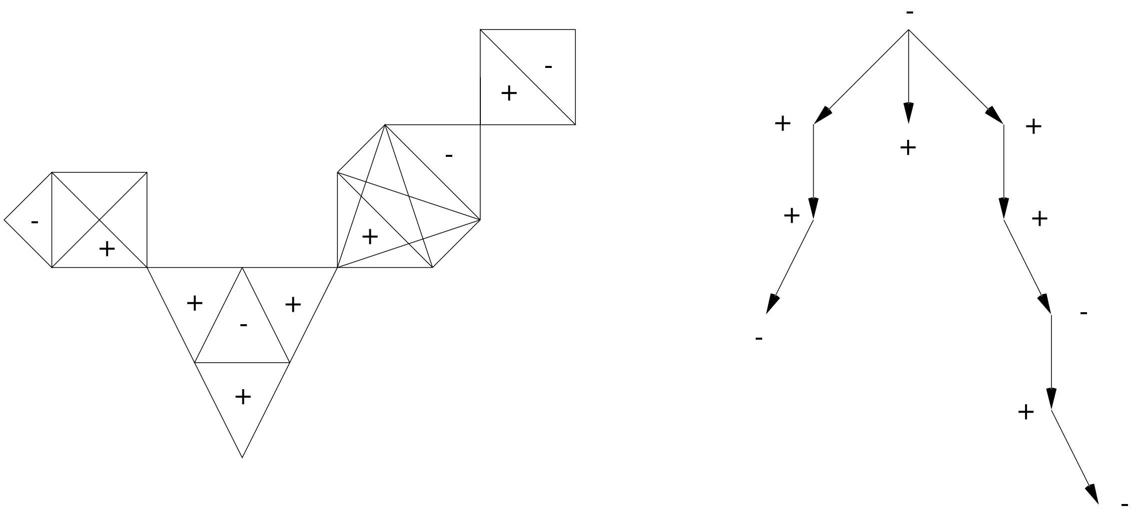

Fig. 15. Labelling of the graph G and its associated tree.

4.2. Hamiltonian cycle detection in L_3^2

In [6], the authors studied the null label problem and prove a sufficient condition for a 3−hypergraph to be null. In particular, the result uses Hamiltonian graphs in L_3^2. Here, we show that deciding if $G \in L_3^2$ is Hamiltonian is NP-complete, limiting the possible application of the result.

Theorem 4.1. *The Hamiltonian cycle problem is NP-complete in L_3^2 even for graphs $G = L_3^2(H)$ where $m(H) = 3$.*

Proof. We give a polynomial transformation from the Hamiltonian cycle problem in cubic graphs which is NP-complete [8]. From a cubic graph $G' = (V', E')$, we define $G \in L_3^2$ as follows: to each vertex $v \in V'$ with neighbours u, w, and t corresponds K_v, i.e., the complete graph with the three vertices $(v, \overline{v}, u), (v, \overline{v}, w), (v, \overline{v}, t)$. For each edge $uv \in E'$, we add the edge $(v, \overline{v}, u)(u, \overline{u}, v)$. It is straightforward to verify that $G \in L_3^2$, and that H, the hypergraph such that $G = L_3^2(H)$, satisfies $m(H) = 3$. Moreover, G' has a Hamiltonian cycle if and only if G has one. □

Remark 4.1. In [9], it is proved that the Hamiltonian cycle problem remains NP-complete for cubic planar graphs. Since K_3, the subgraph replacing each vertex in our reduction is planar, it is straightforward, using the same transformation, that the Hamiltonian cycle problem is NP-complete in L_3^2 even for planar graphs.

4.3. Recognition problem for trees

We finish this section addressing the recognition problem for trees in L_3^2.

Property 4.4. *Let T be a tree. $T \in L_3^2$ if and only if $\Delta(T) \leq 3$.*

Proof. Let T be a tree. If $\Delta(T) \geq 4$ then T contains $K_{1,4}$ as an induced subgraph and so $T \notin L_3^2$. Now $\Delta(T) \leq 3$. We use induction on n, the number of vertices of T. The cases $n = 1$ and $n = 2$ are trivial. Let v be a leaf of T. By our induction hypothesis, $T - v$ has a λ_3^2-labelling. Let w be the neighbour of v in T. In $T - v$, w has degree at most two. Without loss of generality, let $w = \{1, 2, 3\}$ and $w' = \{1, 2, 4\}$ be the labelling of a neighbour w' of w in $T - e$. When w' is the unique neighbour of w in T_e then $v = \{2, 3, 5\}$. Else w'' is the second neighbour of w in T_e. Let $w'' = \{1, 3, 5\}$. Then $v = \{2, 3, 6\}$. □

5. Conclusions

Basing on some results in literature stating that the membership problem related to the class of L_3^2 graphs is NP-complete, we consider the subclass of claw-free graphs in L_3^2 and we prove that the membership problem still remains non polynomially solvable. On the other hand, if we further restrict the class to triangulated claw-free L_3^2 graphs, then, for each element, we are able to provide a possible realization in polynomial time. Consequently, the related membership problem turns out to be polynomially solvable. These results set the path to a deeper analysis of L_3^2 graphs to constrain the subset where the membership problem preserves its non polynomiality.

Pushing things further, we suppose that similar tools can be used to study the membership problem both on different L_k^l classes and on intersection graphs of general hypergraphs, so relaxing the uniformity constrain.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

This research was partially supported by the INdAM-GNCS Project 2023, CUP_E53C22001930001, *"Combinatorial and enumerative aspects of discrete structures: strings, hypergraphs and permutations"*

References

[1] L.W. Beineke, Characterization of derived graphs, J. Combin. Theory 9 (1970) 129–135.
[2] C. Berge, Hypergraphs, North Holland, 1989.
[3] A. Berry, G. Simonet, Computing a clique tree with the algorithm maximal label search, Algorithms 10 (1) (2017) 20.
[4] A.A. Bertossi, The edge Hamiltonian path problem is NP-complete, Inform. Process. Lett. 13 (1981) 157–159.
[5] J.A. Bondy, U.S.R. Murty, Graph Theory, Springer, 2008.
[6] N. Di Marco, A. Frosini, W.L. Kocay, A study on the existence of null labelling for 3—hypergraphs, in: P. Flocchini, L. Moura (Eds.), Combinatorial Algorithms, IWOCA 2021, in: Lecture Notes in Computer Science, vol. 1275, Springer, Cham, 2021, pp. 282–294, http://dx.doi.org/10.1007/978-3-030-79987-8_20.
[7] N. Di Marco, A. Frosini, W. Kocay, E. Pergola, L. Tarsissi, Structure and complexity of 2-intersection graphs of 3-hypergraphs, Algorithmica 85 (2023) 745–761.
[8] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, 1979.
[9] M.R. Garey, D.S. Johnson, R.E. Tarjan, The planar Hamiltonian circuit problem is NP-complete, SIAM J. Comput. 5 (1976) 704–714.
[10] P. Hliněný, J. Kratochvíl, Lect. notes in comput. sci, in: Computational Complexity of the Krausz Dimension of Graphs, 1335, 1997, pp. 214–228.
[11] I. Holyer, The NP-completeness of edge-coloring, SIAM J. Comput. 10 (1981) 718–720.
[12] J. Hopcroft, R.E. Tarjan, Algorithm 447: efficient algorithms for graph manipulation, Commun. ACM 16 (6) 372–378.
[13] M. Yannakakis, F. Gavril, Edge dominating sets in graphs, SIAM J. Appl. Math. 38 (3) (1980) 364–372.