HAL open science

The null labelling of k-uniform hypergraphs and the complexity of the 2-intersection graph decision problem

Niccoló Di, Andrea Frosini, William Kocay, Elisa Pergola, Lama Tarsissi

To cite this version:

Niccoló Di, Andrea Frosini, William Kocay, Elisa Pergola, Lama Tarsissi. The null labelling of k-uniform hypergraphs and the complexity of the 2-intersection graph decision problem. Algorithmica, In press. hal-03685794

HAL Id: hal-03685794

https://hal.science/hal-03685794v1

Submitted on 2 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The null labelling of k-uniform hypergraphs and the complexity of the 2-intersection graph decision problem.

Niccoló Di Marco[1*], Andrea Frosini[1], William Lawrence Kocay[2], Elisa Pergola[1] and Lama Tarsissi[3]

[1*]Department of Mathematics and Computer Science, University of Florence, Florence, Italy.
[2]Department of Computer Science and St. Pauls College, University of Manitoba, Manitoba, Canada.
[3]LIGM, CNRS, F-77454 Université Gustave Eiffel, Marne-la-Vallée, France .
[3]Sorbonne University Abu Dhabi, United Arab of Emirates.

*Corresponding author(s). E-mail(s): niccolo.dimarco@unifi.it; Contributing authors: andrea.frosini@unifi.it; bkocay@cs.umanitoba.ca; elisa.pergola@unifi.it; lama.tarsissi@esiee.fr;

Abstract

Given a $\mathbf{3}$-uniform hypergraph $\boldsymbol{H}$ having a set $\boldsymbol{V}$ of vertices, and a set of hyperedges $\boldsymbol{T} \subset \boldsymbol{\mathcal{P}(V)}$, whose elements have cardinality three each, a *null labelling* is an assignment of $\pm\mathbf{1}$ to the hyperedges such that each vertex belongs to the same number of hyperedges labelled $+\mathbf{1}$ and $-\mathbf{1}$. A sufficient condition for the existence of a null labelling of $\boldsymbol{H}$ (proved in [6]) is a Hamiltonian cycle in its $\mathbf{2}$-intersection graph. The notion of $\mathbf{2}$-intersection graph generalizes that of intersection graph of an (hyper)graph and extends its effectiveness. The present study first shows that this sufficient condition for the existence of a null labelling in $\boldsymbol{H}$ can not be weakened by requiring only the connectedness of the $\mathbf{2}$-intersection graph. Then some interesting properties related to their clique configurations are proved. Finally, the main result is proved, the NP-completeness of this characterization and, as a consequence, of the construction of the related $\mathbf{3}$-hypergraphs.

MSC Classification: 05C65 , 05C22 , 05C85 , 05C99

1 Introduction

The degree sequence of a graph or hypergraph provides useful information about its geometrical and topological properties. When a graph or hypergraph has been constructed for a given degree sequence, we say that it has been *reconstructed*. A longstanding problem concerns the use of degree sequences for the reconstruction of the related graphs or hypergraphs. In 1960, in a milestone paper, Erdös and Gallai [7] characterized the degree sequences of simple graphs and set the basis for the successive definition of fast reconstruction algorithms.

These results can not be easily generalized to hypergraphs, even in the simplest form of the k-uniform ones, with $k > 2$. In recent years, many different reconstruction strategies have been considered, most of them relying on a general result of Dewdney in [4], concerning relevant sub-classes of k-uniform hypergraphs (see [1, 3, 9, 10]).

Recently, Deza et al. proved the NP-completeness of the reconstruction problem of 3-uniform hypergraphs from their degree sequences. So, the study of the classes of hypergraphs whose reconstruction can be performed in polynomial time becomes more relevant, as does the characteristics of the solutions related to a single degree sequence.

As an example, in [15], the authors defined a general operator to move among all the 3-hypergraphs sharing the same degree sequence and they introduced the notion of null-labelling obtained as the symmetric difference of the edges of any two 3-hypergraphs. In [8], this notion, restricted to 3-hypergraphs, is studied through their intersection graph and a sufficient condition for its existence was provided.

Finally, in [6], it was shown that as the number of edges of the related 3-hypergraph increases, the usefulness of the intersection graph decreases, and the stronger notion of 2-intersection graph was introduced. In particular, the presence of an Hamiltonian cycle in the 2-intersection graph guarantees the existence of, and allows one to compute, the null labelling of the related 3-hypergraph.

Our study relies on these results and aims to deepen the knowledge of the structural properties of 2-intersection graphs in order to enhance their use for the null labelling problem. We are concerned with the question of when, given a graph G, it is possible to find a 3-hypergraph H such that $I_2(H) \cong G$. If it is possible, we say that G is *reconstructible* or equivalently, it has the 2-*intersection property*. In the final section of the paper we prove that the reconstruction problem is NP-complete.

2 Definitions and preliminary results

Let us recall some basic definitions for graph and hypergraphs, borrowing the notation from [2]. A graph G is defined as a pair $G = (V, E)$ such that $V = \{v_1, \ldots, v_n\}$ is the set of *vertices* and E is a subset of pairs of vertices called *edges*. The notion of graph admits a natural generalization to hypergraph by removing the constraint on the cardinality of the edges: $H = (V, E)$ is a hypergraph with $V = \{v_1, \ldots, v_n\}$ the set of *vertices* and E a collection of subsets of vertices called hyperedges, or briefly edges, when no misunderstanding may arise. We also choose to abbreviate the vertex notation v_i with its only index i, when possible.

In the sequel, we will consider only graphs and hypergraphs that are *simple*, i.e., they do not allow singleton (hyper)edges or (hyper)edges that are contained in or equal to other edges. The *degree* of a vertex $v \in V$ is the number of (hyper)edges containing v. The degree sequence $d = (d_1, d_2, \ldots, d_n)$ of a (hyper)graph is the list of its vertex degrees usually arranged in non-increasing order. We focus on a remarkable class of hypergraphs, i.e., k-uniform hypergraphs or simply k-hypergraphs, whose hyperedges have fixed cardinality k. In particular we are going to consider 3-hypergraphs.

Given a (hyper)graph, we can assign a $+1$ or -1 label to each (hyper)edge, resulting in positive and negative degrees of the vertices.

An assignment of ± 1 to the (hyper)edges of a (hyper)graph is a *null labelling* if, for each vertex v, the sum $d(v)$ of the labels of its incident edges equals 0. A (hyper)graph that admits a null labelling is said to be a *null (hyper)graph*.

The study of null (hyper)graphs is strictly related to their reconstruction from the degree sequences. Let $H_1 = (V, E_1)$ and $H_2 = (V, E_2)$ be two hypergraphs with the same vertex set V, and same degree sequence $(d_1, d_2, \ldots, d_n)$. Assign $+1$ to the triples of H_1 and -1 to the triples of H_2, and construct $H_1 \oplus H_2 = (V, (E_1 \cup E_2) \setminus (E_1 \cap E_2))$. It is a hypergraph with a null labelling. This raises the question of whether there is a characterization of null (hyper)graphs. Concerning graphs, the following lemma characterizes those with null labelling.

Lemma 1 A graph G has a null labelling if and only if every connected component is an Eulerian graph with an even number of edges.

On the other hand, for k-hypergraphs with $k \geq 3$, it is shown in [8] that the problem of finding a null labelling even for the simplest case of 3-hypergraphs is NP-hard.

An obvious necessary condition for a (hyper)graph to have a null labelling is that each vertex must have even degree, i.e., it is an *even (hyper)graph*.

So, the characterization of subclasses of 3-hypergraphs having a null labelling acquires relevance. In [8], an exhaustive computation of the simplest cases supported the following

Conjecture 1. All connected even 3-hypergraphs on n vertices with more than $\frac{1}{2}\binom{n}{3}$ edges have a null labelling.

Note that $\binom{n}{3}$ is the maximum number of edges a (simple) 3-hypergraph on n vertices admits.

A valuable tool to inspect the characteristics of hypergraphs in general and, in our framework of 3-hypergraphs, is their intersection graph. The *intersection graph* of a 3-hypergraph H is denoted $I(H)$, its vertices are the hyperedges of H and two hyperedges are adjacent, i.e., they are joined by an edge, if their intersection is non-empty. This is an extension of the idea of a line graph to 3-hypergraphs. In [8], the following result has been proved

Theorem 1 Let H be a connected, even 3-hypergraph, in which every vertex has degree two. Then H has a null labelling if and only if $I(H)$ is bipartite.

However, the intersection graphs dramatically collapse into complete graphs when the number of edges of the related 3-hypergraph increases, so in [6] the notion of 2-*intersection graph* was introduced. The 2-intersection graph of a 3-hypergraph H is denoted $I_2(H) = (V_{2H}, E_{2H})$. Its vertices $V_{2H} = \{v_{e_1}, \ldots, v_{e_m}\}$ represents the hyperedges $E = \{e_1, \ldots, e_m\}$ of H. Two vertices $v_{e_i}, v_{e_j} \in V_{2H}$ are adjacent, i.e., $\{v_{e_i}, v_{e_j}\} \in E_{2H}$, if the related hyperedges e_i and e_j share two vertices in H (see Example 1 and the related Fig. 1). In the sequel, we label the edge $\{v_{e_i}, v_{e_j}\} \in E_{2H}$ with the indices of the vertices that are shared by e_i and e_j, if needed.

The idea of the 2-intersection graph is related to Steiner triple systems, in which any two triplets can intersect in at most one vertex. In the diagrams following, a 3-hypergraph is represented by its Levi graph, that is, a bipartite graph of vertices versus hyperedges, in which each hyperedge is represented by a vertex shaded grey, and of degree three.

Example 1. Consider the 3-hypergraph $H = (V, E)$, with $V = (v_1, \ldots, v_5)$ and $E = \{\{v_1, v_3, v_5\}, \{v_2, v_3, v_5\}, \{v_2, v_3, v_4\}, \{v_3, v_4, v_5\}\}$ as depicted in Fig. 1, top (a) (vertex v_i is indicated by its index i). The related $I_2(H)$ is in Fig. 1, bottom (a). It has $\mid E \mid = 4$ vertices, labelled by the corresponding triplets of the hyperedges of H, and five edges, labelled by the pair of vertices they contain. Furthermore, the common label $\{3, 5\}$ is assigned to the gray triangle clique.

The following property is straightforward:

Property 1. The edges of a 3-hypergraph H sharing the same pair of vertices determine a clique in the 2-intersection graph $I_2(H)$.

In the sequel, we extend the common label of the edges of a clique of a 2-intersection graph to label the clique itself.

Since each edge of H has three pairs of vertices to share, the following holds

Property 2. Given a 3-hypergraph H, each vertex of $I_2(H)$ belongs to at most three maximal cliques.

Our main focus is to study the properties of the 2−intersection graph and its connection with the null labelling problem. Moreover, we prove that, given a graph G, deciding if G is the 2-intersection graph of some 3-hypergraph is an NP-complete problem. Therefore, we consider the following questions.

Question 1: is there a subclass of null 3-hypergraphs whose null labelling relies on properties of the 2-intersection graphs (recall from [8] that the problem is NP-complete in its general form)?

Question 2: given a graph G, is it possible to decide in polynomial time whether there exists a 3-hypergraph H such that $G \cong I_2(H)$?

The next section presents a brief review of [6], i.e. Question 1, while Section 4 will answer Question 2 by providing a proof of its NP-completeness.

3 A subclass of 3-hypergraphs having null labelling

Recall that in [6] the authors proved the following theorem.

Theorem 2 Let H be a 3-hypergraph. If the 2-intersection graph $I_2(H)$ is Hamiltonian, then H admits a null labelling.

The proof is based on an iterative algorithm that, moving along the vertices of the 2-intersection graph, reduces step by step the signed degree of the vertices, considered by absolute value. The presence of an Hamiltonian cycle guarantees that all the vertices can be reached by the algorithm.

Define the n-intersection graph of a k-hypergraph H as the graph G in which the vertices are the hyperedges of H and two vertices form an edge if their hyperedges share at least n vertices, and denote it by $I_n(H)$. The following obvious generalization of Theorem 2 holds.

Corollary 1 Let H be a k-hypergraph. If $I_{k-1}(H)$ is Hamiltonian, then H admits a null labelling.

Theorem 2 is a step towards proving Conjecture 1. However, an easy check reveals that not all the 3-hypergraphs with more than half the possible edges and with a null labelling, have a Hamiltonian cycle in $I_2(H)$. On the other hand, after generating several 3-hypergraphs on 6, 7 and 8 vertices, we noticed that the connectedness of the graph $I_2(H)$ seems to be related to the presence

of a null labelling in H. This property could extend the results in Theorem 2. Unfortunately, successive computations show that $I_2(H)$ being connected is not sufficient to provide a null labelling of H as shown by the following

Example 2. Consider the following even 3-hypergraph

$$H = \{159, 269, 147, 369, 459, 146, 368, 157, 156, 189, 279, 246, 379, 239\}.$$

It is easy to check that H does not have a null labelling although its 2-intersection graph $I_2(H)$ is connected.

Remark 1. Connectedness of the 2-intersection graph does not characterize the set of 3-hypergraphs meeting the lower bound in Conjecture 1 as shown by the following example.

Example 3. The hypergraph H defined as

$$E = \{134, 245, 247, 125, 567, 167, 146, 157, 357, 135, 347, 467, 127, 236, 257, 457, 156, 345\}$$

has 7 vertices and 18 hyperedges. It satisfies Conjecture 1 as shown by the following null labelling

$$[1, 1, 1, 1, 1, -1, -1, 1, 1, -1, -1, -1, -1, -1]$$

However it has a disconnected $I_2(H)$.

Next, we show an extension of Theorem 2. The idea comes from the following observation: consider an even graph G with n connected components such that each component has an even number of edges. We can find a null labelling of G by considering, for each component, an Eulerian cycle, assigning alternatively ± 1 to its edges. Note that, if we have an intersection graph $I(G)$ with n components, each having an Hamiltonian cycle, in the same fashion, we can alternately assign ± 1 to the vertices of each Hamiltonian cycle to obtain a null labelling. This same property holds for 3-hypergraphs.

Theorem 3 Let H be a 3-hypergraph whose 2-intersection graph $G = I_2(H)$ has n connected components $G_1, \ldots, G_n$. If each connected component G_i, with $1 \leq i \leq n$, is Hamiltonian, and the sub-3−hypergraph $H_i \subset H$ such that $G_i = I_2(H_i)$ is even, then H admits a null labelling.

Proof Since each G_i is Hamiltonian and comes from an even sub-3-hypergraph H_i, we can apply Theorem 2 to it, obtaining a null labelling of H_i. The null labelling of H is obtained by the union of the null labellings of these sub-hypergraphs. □

Note that H may be also a connected 3−hypergraph, but the requirement of the previous theorem asks that its 2−intersection graph has to be disconnected.

Furthermore, each of its components comes from an even sub−3−hypergraph of H.

4 The 2-Intersection-Graph Problem

In this section we provide some structural properties of 2-intersection graphs that are related to 3-hypergraphs, with the aim of characterizing them and determining the computational complexity of this problem.

Recall that a graph G has the 2-intersection property if it is the 2-intersection graph of a 3-hypergraph H.

So, we approach the problem of the existence of a 3-hypergraph H whose 2-intersection graph is G, i.e., $G \cong I_2(H)$, by first observing that Property 1 does not characterize the cliques of $I_2(H)$. In fact, there may appear triangles (K_3 cliques) whose edges do not share a common label. In order to distinguish them from the triangles whose edges have a common label, we indicate the first and the second types as T-triangles and K-triangles, respectively.

More precisely, the vertices $\{v_1, v_2, v_3\}$ of a T-triangle have the general form $v_1 = \{x, y, z\}$, $v_2 = \{x, y, t\}$, and $v_3 = \{x, z, t\}$ — only four vertices of H are used; while those of a K triangle (with common label $\{x, y\}$) have the form $v_1 = \{x, y, z\}$, $v_2 = \{x, y, t\}$, and $v_3 = \{x, y, r\}$ — five vertices of H are used.

Figure 1, (a), shows a configuration of a 2-intersection graph where a T-triangle and a K-triangle live together and share an edge. The T-triangle is shaded gray.

t1t2.pdf

Fig. 1 (a) example of a T and a K sharing 2 vertices. (b) example of S.

With the previous definitions, a first obvious necessary condition for a graph to be reconstructible is described in the following lemma.

Lemma 2 If G is *reconstructible*, then every vertex of G must belong to at most three maximal cliques, such that the vertices of each of them have in common the same couple of labels. Furthermore, these maximal cliques can be joined by at most two non-adjacent T-triangles as shown in Fig. 2.

Proof A clique in $I_2(H)$ can derive from a shared pair or it can be a triangle (in which two pairs are used). In all cases, since a vertex of $I_2(H)$ has only three pairs, it cannot belong to more than three cliques. However, note that the configuration in which a point is shared by four triangles is reconstructible (see Fig. 2).

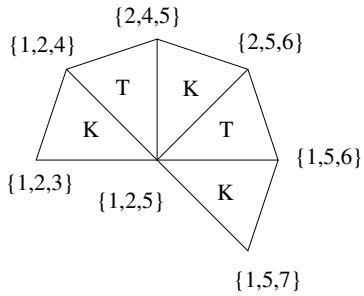

Fig. 2 Reconstruction of the configuration in which a point belongs to 4 triangles.

The reason is simple: the edges of the T triangles shared with the K triangles count as a unique pair containing the central point. Therefore, it shares exactly three pairs, as desired. Note that the configuration is reconstructible even if the K triangles are replaced by any clique of order at least four. □

We encounter a similar situation when dealing with K_4 cliques: Figure 1, (b), shows a clique that may be present in a 2-intersection graph and whose edges do not share a common label. This clique is composed of four intersecting T-triangles. It arises from four vertices of H for which all possible triples occur as hyperedges. So, maintaining the notation introduced for triangles, we denote such a square by S, and we differentiate it from the K-square, i.e., the 4-clique whose edges all have a common label.

Property 3. Let C_1 and C_2 be two cliques of $I_2(H)$. If $\mid C_1 \cap C_2 \mid \geq 3$ then $C_1 = C_2$.

Proof Let $C = \{v_1, v_2, v_3\} \subseteq C_1 \cap C_2$. We proceed by contradiction assuming that there exist two non adjacent vertices $x \in C_1$ and $y \in C_2$ such that $C_1' = C \cup x$ and $C_2' = C \cup y$ are two different cliques. The following two cases arise (see cases (a) and (b) in Fig. 1).

C is a T-triangle: we suppose w.l.g. $v_1 = \{1, 2, 3\}$, $v_2 = \{1, 2, 4\}$, and $v_3 = \{1, 3, 4\}$. Since C_1' is a clique, then $x = \{2, 3, 4\}$ is the only possibility for x. The same reasoning holds for y, so it follows $x = y$, against the hypothesis.

C is a K-triangle: we suppose w.l.g. $v_1 = \{1, 2, 3\}$, $v_2 = \{1, 2, 4\}$, and $v_3 = \{1, 2, 5\}$. Both x and y have to contain the pair $\{1, 2\}$, shared by all the elements of C, so there exists an edge joining them, against the hypothesis.

□

In the proof of Property 3, we note that if C is a K-triangle, then only two of its elements are enough to determine the common pair $\{1, 2\}$, so that:

Corollary 2 Let C_1 and C_2 be two cliques of $I_2(H)$ not containing any T-triangle. If $\mid C_1 \cap C_2 \mid \geq 2$ then $C_1 = C_2$.

Corollary 3 Let C_1 and C_2 be two (maximal) cliques of $I_2(H)$. If $|C_1 \cap C_2| = 2$, then either C_1 or C_2, but not both, are T-triangles.

We emphasize that, if both C_1 and C_2 are T-triangles such that $|C_1 \cap C_2| = 2$, then they are not maximal, i.e., they form an S-square as in Fig. 1, (b).

The following example shows that S is the biggest clique that does not admit a common pair in all of its edges.

Example 4. Consider a clique K_5 of a 2-intersection graph. It is not possible to label its vertices without a common pair. In fact, let us start by labelling a vertex with $\{1, 2, 3\}$ (see Fig. 3) and one of its neighbours with $\{1, 2, 4\}$. Suppose that we label a third vertex with a pair different from $\{1, 2\}$, say $\{1, 3, 4\}$. Then we have only one label possible for the fourth vertex, namely $\{2, 3, 4\}$. We observe that there does not exist any label for the fifth vertex, since all possible pairs have been already used. So a labelling of the vertices of K_5 without a common pair is not possible.

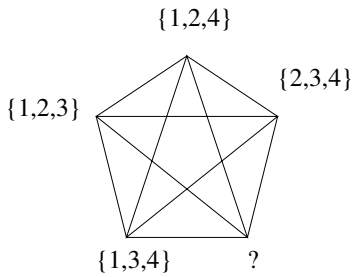

Fig. 3 The labelling of the clique K_5 used in Example 4.

The same argument can be used to prove that any clique of order greater than four has the edges labelled with the same pair of indices. Many other conditions can help to restrict the cases when G is a 2-intersection graph.

Theorem 4 Let G be a graph containing two distinct cliques of order n and m and intersecting in $2 \leq i \leq \min(m, n) - 1$ vertices. Moreover, assume that the condition of Lemma 2 holds. Then the following statements hold:

(i) if $n = 3$ and $m \geq 3$, then G is reconstructible;

(ii) let $n = 4$ and $m \geq 4$, then G is reconstructible if and only if $i = 2$;

(iii) if $n > 4$ and $m > 5$, then G is not reconstructible.

Proof (i): by hypothesis, $i = 2$, so the two cliques must intersect in two vertices. From Property 3 and the corollaries, a clique K_m, with $m \geq 3$ shares a common edge, say with label $\{x, y\}$, with a T-triangle. We proceed in labelling the

three vertices of T by $\{x, y, z_1\}, \{x, y, z_2\}, \{x, z_1, z_2\}$. Finally, we assign the labels $\{x, y, z_3\}, \{x, y, z_4\}, \ldots, \{x, y, z_m\}$ to the remaining vertices of K_m. The labels turn out to be the hyperedges of a 3-hypergraph H on $\{x, y, z_1, \ldots, z_n\}$ vertices such that $I_2(H) = G$, as desired.

(ii): from Property 3 and corollaries, if $i = 2$, then the edge shared by the two cliques must intersect a triangle T. Let T be contained in the first clique whose configuration turns out to be S. The other clique is forced to have a common label of the edges, since it can not contain a second T-triangle. So, a labelling similar to that defined in (i) is allowed, providing a 3-hypergraph whose 2-intersection graph is G.

Finally, the case $i = 3$ causes the collapse of the two cliques into one single clique.

(iii): consider two cliques of order greater than four. Property 3 states that if they share two or more vertices, then they are the same clique, so G is not a 2-intersection graph. □

The above result can be naturally extended to graphs having more than two cliques, by inspecting the intersections between pairs of them. A labelling of the edges and the determination of a 3-hypergraph compatible with them can be performed.

However, there are several classes of graphs that can be immediately detected as 2-intersection graphs due to their geometric properties. For them, some simple edge labellings lead to the determination of a related 3-hypergraph. One class is the cycle graphs, i.e. a sequence of consecutive 2-cliques.

First, we observe that a cycle of length k admits a labelling involving k integer indices at most. In fact, we start by labelling a randomly chosen vertex v_1 with $\{1, 2, 3\}$, then we visit the remaining vertices $v_2, \ldots, v_k$ of the cycle according to a chosen direction and we label the ith one, where $1 < i < k - 1$, with $\{i, i + 1, i + 2\}$, so that $\{i, i + 1\}$ is the pair shared by v_{i-1} and v_i, i.e., the label of the edge joining them. Finally, vertices v_{k-1} and v_k are labelled with $\{1, k, k + 1\}$ and $\{1, 2, k\}$, respectively. An example is shown in Fig. 4.

Note that the labelling of the first $k - 1$ vertices of the cycle defined above can be used to label a path of length $k - 1$ involving k integer indices.

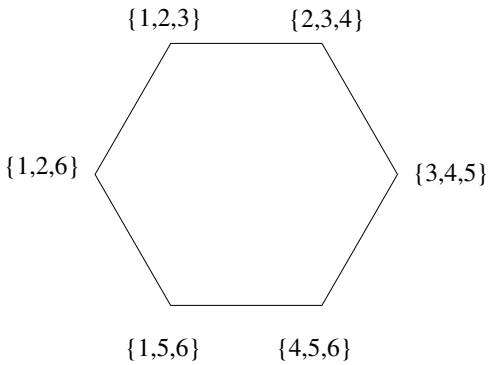

Fig. 4 The labelling of a cycle of length 6.

5 Deciding if a graph has the 2-intersection property is NP-hard

In this section it is shown that it is NP-complete to determine whether an arbitrary graph G is the 2-intersection graph of a 3-hypergraph. We refer to this problem as then *2INT* problem. We reduce from the following variant of the 3-SAT problem (L02 in [11])

MAX-3-SAT:

INSTANCE: a set U of *variables* and a set C of *clauses* over U such that each clause $c \in C$ has $\mid c \mid = 3$ *literals*, where a literal corresponding to a variable x is either x or $\overline{x}$. Each variable, as represented by a literal, appears at most three times in the clauses. Furthermore, no three occurrences of a variable are the same literal.
QUESTION: Is there a satisfying truth assignment for C?

First of all, note that this problem is $NP-$complete.

Lemma 3 MAX-3-SAT is NP-complete.

Proof We reduce 3-SAT to MAX-3-SAT. Consider an instance of 3-SAT, in which a variable x appears k times, either as x or $\overline{x}$. Replace the first instance with x_1, the second with x_2 and so on up to x_k (possibly the complements). Then add $(\overline{x}_1 \vee x_2) \wedge (\overline{x}_2 \vee x_3) \ldots (\overline{x_k} \vee x_1)$. This ensures that either all x_i are true, or all are false. Note that each x_i has two literals occurring in this expression, and one more literal occurring in the clauses of 3-SAT, giving three occurrences in total. Doing the same for the others literals we obtain an instance of 3-SAT, but in which each variable appears at most three times, and no three occurrences of a variable are the same literal. □

Given an instance C of MAX-3-SAT, we construct a graph G_C so that there is a solution of the MAX-3-SAT instance if and only if G_C is a 2-intersection graph. This will imply that *2INT* is NP-complete. We remark that each solution of C will be achieved by one of the possible labellings of G_C that determines a 3-hypergraph H such that $G = I_2(H)$.

To achieve the objective, we need to define some graph configurations that have the 2-intersection property and that are useful to model variables and clauses of 3-SAT. Each vertex of a 2-intersection graph G corresponds to a hyperedge of its related hypergraph H. So each vertex v of G contains three vertices of H. In order to avoid confusion with the use of the word vertex, we will use the term *indices* for the vertices of H contained in a vertex of G. When a hypergraph H is constructed from a 2-intersection graph G, by assigning indices to the vertices of G, we say that a *label* is assigned to v, eg., we assign $\{1, 2, 4\}$ as the label of a vertex of G. In the following properties, we consider subgraphs of G determined by triangles, and show that when G is a

2-intersection graph, the labels of G have to satisfy certain requirements, i.e., local properties of G extend to properties of H.

Property 4. If two triangles intersecting in one vertex have the 2-intersection property, and there are no edges joining the triangles, then they are not both T-triangles.

Proof Observe that the three vertices of a T-triangle use just four different indices from H. If the two triangles are T-triangles, then the common vertex has to share four different pairs of indices with the four vertices with which it is adjacent, and this is not possible if there are no edges joining the triangles (see Fig. 5). □

We will call two triangles sharing one vertex, with no edges between them, a *ribbon* configuration. Figure 5 shows the labels of two one-vertex intersecting triangles when at least one of them is a K-triangle.

Fig. 5 Two possible labels of a ribbon configuration. In each of them at most one triangle is a T-triangle.

Property 5. Let T_1 and T_2 be two triangular cliques. Suppose there are just two edges joining a vertex of T_1 to a vertex of T_2, and that these edges have no common endpoints. Then the obtained configuration has the 2-intersection property. Furthermore, T_1 and T_2 cannot both be T-triangles. It is possible for T_1 and T_2 to be both K-triangles.

Proof Let x_1 and y_1 be the vertices of T_1 adjacent to the vertices x_2, and y_2 of T_2 by the edges e_x and e_y, respectively. We first show that that T_1 and T_2 cannot be both T-triangles. Suppose that T_1 is a T-triangle whose vertices have labels $x_1 = \{1, 2, 3\}$, $y_1 = \{2, 3, 4\}$, and $z_1 = \{1, 2, 4\}$ (see Fig. 6). Since x_2 is adjacent to only x_1 in T_1, then the label of e_x must be $\{1, 3\}$. With the same argument, the label of e_y is $\{3, 4\}$. Since x_2 and y_2 belong to T_2, they have a common pair of labels, say $\{3, 5\}$. So their labels must be $x_2 = \{1, 3, 5\}$ and $y_2 = \{3, 4, 5\}$. If we assume T_2 to be a T-triangle, then $z_2 = \{1, 4, 5\}$. This leads to a contradiction since z_2 is not adjacent to $z_1 = \{1, 2, 4\}$. Therefore the only possible label is $z_2 = \{3, 5, 6\}$, showing that T_2 is a K-triangle.

To show that T_1 and T_2 can both be K-triangles, we set $x_2 = \{1, 3, 5\}$ and $y_2 = \{3, 4, 5\}$ and $z_2 = \{3, 5, 6\}$ in T_2 (see Fig. 6). We can then take $x_1 = \{1, 2, 3\}$, $y_1 = \{2, 3, 4\}$ and $z_1 = \{2, 3, 7\}$ in T_1. □

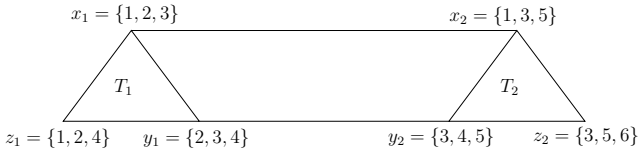

Fig. 6 The configuration obtained by two triangles with joined by two edges. One possible labelling is shown.

Representing the variables of U

Define a *variable gadget*, denoted G_x, to represent a variable x in U. The gadget is a 2-intersection graph obtained by the union of different configurations and its definition can be checked in Fig. 7.

The graph G_x consists of two gadgets G_x^1 and G_x^2 that grow around two K_5 cliques C_1 and C_2 that are connected by a ribbon configuration, say R.

The gadget G_x^1 includes the clique C_1 whose vertices are in common with five different K_3 triangular cliques, counter-clockwise denoted by T_i^1, with $1 \leq i \leq 5$, and starting from the triangle facing the ribbon that joins the two gadgets. These triangles are then connected in pairs by their two remaining free vertices, forming a 10-cycle, as shown in Fig. 7.

The triangles T_2^1 and T_5^1 will be associated with the two (at most) occurrences of the variable x, while T_3^2 will be associated with the single occurrence of the variable $\overline{x}$.

We show that the graph G_x has the 2-intersection property. Furthermore, it allows very few possibilities when setting the triangles as K or T.

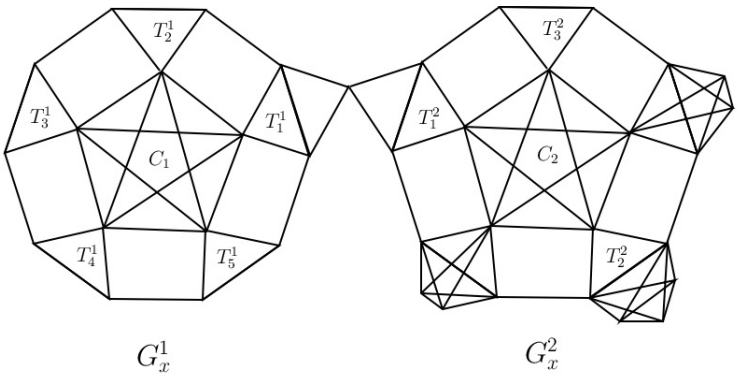

Fig. 7 The graph G_x. It is used to model the occurrences of the literals x and $\overline{x}$ in the clauses of an instance of MAX-3-SAT. The two parts G_x^1 and G_x^2 are indicated together with their connection through a ribbon configuration.

Property 6. The variable gadget G_x is a 2-intersection graph.

In Fig. 8 and 9 we provide samples of labellings of G_x.

Corollary 4 A labelling of the vertices of the graph G_x only produces the following possible types (K or T) of the triangles T_2^1, T_5^1 and T_3^2:

i) if $T_3^2 = T$ then $T_2^1 = T_5^1 = K$;

ii) if $T_5^1 = T$ (resp. $T_2^1 = T$) then $T_3^2 = K$.

Proof i) let us assume $T_3^2 = T$. Property 5 assures that triangle T_1^2 is a K-triangle. So, by Corollary 3, the triangle adjacent to T_1^2 is a T-triangle, and the triangle adjacent to T_1^1 is a K-triangle by Property 4. Continuing with the configurations, we get $T_1^1 = T$, and, again by Property 5, we obtain that both T_2^1 and T_5^1 are K-triangles as desired (see a possible labelling in Fig. 8).

ii) let us assume $T_5^1 = T$ (resp. $T_2^1 = T$). Property 5 assures that $T_1^1 = K$. So, by Corollary 3, the triangle adjacent to T_1^1 is a T-triangle, and the triangle adjacent to T_1^1 is a K-triangle by Property 4, and finally $T_1^2 = T$. Again Property 5 assures that T_3^2 is a K-triangle as desired (see a possible labelling in Fig. 9). □

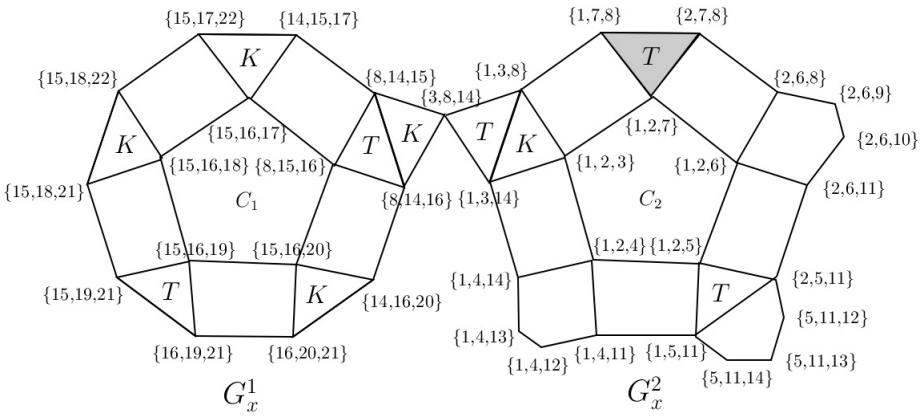

Fig. 8 One of the labellings of G_x related to case *i*) of Corollary 4. The starting triangle $T_3^2 = T$ is shaded. The type (T or K) of each triangle is also shown.

Remark 2. We point out that in case *ii*) of Corollary 4 no assumption involves the type of T_2^1, that can be either T or K, when $T_5^1 = T$ and vice versa. Figure 9 shows an example of a labelling where $T_1^2 = K$. An easy check reveals that exchanging the types K and T between T_1^3 and T_1^2 produces a new admissible labelling.

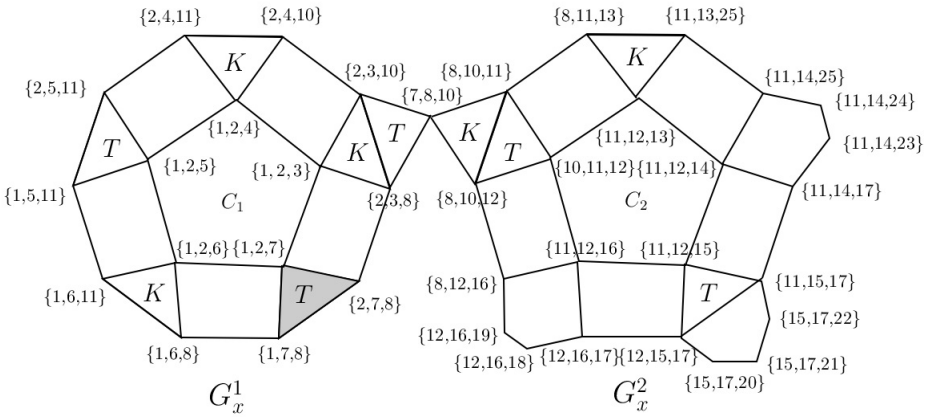

Fig. 9 One of the labellings of G_x related to case *ii*) of Corollary 4. The starting triangle $T_3^2 = T$ is shaded. The type (T or K) of each triangle is also shown

Representing the clauses of U

The *clause gadget* G_c, represents a single clause $c \in C$. It consists of a central clique K_6 whose vertices also belong to six different K_3 cliques, called *boundary triangles*, as shown in Fig. 10. The boundary triangles are then connected in pairs via the two remaining free vertices, as shown in Fig. 12, forming a 12-cycle. In order to have the 2-intersection property, the clause gadget admits a maximum number of T-triangles among the six boundary ones, as stated in the following property

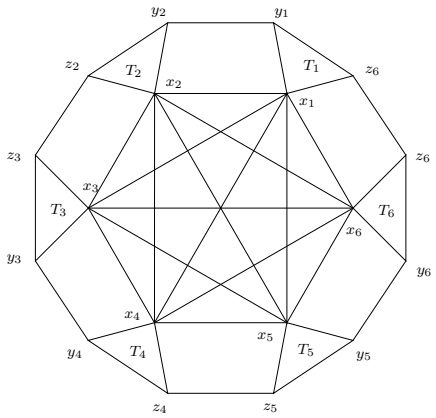

Fig. 10 The clause gadget G_c with its triangles. By Property 5 no more that three T-triangles are allowed in the boundary.

Lemma 4 A clause gadget G_c is a 2-intersection graph if and only if the boundary triangles do not contain either exactly three or exactly one T-triangles.

Proof W.l.g. let $T_1, \ldots, T_6$ be the boundary triangles (labelled counter-clockwise) of a clause gadget. Without loss of generality, let T_i have vertices $\{x_i, y_i, z_i\}$, such that x_i is in common with the central clique K_6, while y_i and z_i connect T_i with the neighbour triangles T_{i-1} and T_{i+1}. Note that the indices $i-1$ and $i+1$ are reduced to the range $1 \ldots 6$. We first show that there is no labelling when three (non consecutive, according with Property 5) boundary triangles are T-triangles. Let T_1, T_3 and T_5 be T-triangles that alternate with three K-triangles T_2, T_4 and T_6. We show that such a configuration does not have the 2-intersection property, i.e., it does not allow any labelling. Assume that the edges of the central clique K_6 have the common label $\{1, 2\}$. Therefore, label every vertex $\{x_1, \ldots, x_6\}$ of K_6 with $\{1, 2, i\}$, with $3 \leq i \leq 8$, respectively, and let x_1 belong to a T-triangle.

W.l.g, its two remaining vertices have labels $y_1 = \{1, 3, 9\}$ and $z_1 = \{2, 3, 9\}$. Suppose T_1 is connected by the edges (y_1, y_2) and (z_1, z_6) with the triangles T_2 and T_6, respectively. So, the edge $\{y_1, y_2\}$ has label $\{1, 9\}$, and the edge $\{z_1, z_6\}$ has label $\{2, 9\}$ (check the labelling in Fig.11, left graph).

So, the labels $y_2 = \{1, 8, 9\}$ and $z_6 = \{2, 4, 9\}$ are determined. Since T_2 is a K-triangle, the label of z_2 requires the introduction of a new index, i.e., $z_2 = \{1, 8, 10\}$. The same holds in T_6, where the label $y_6 = \{2, 4, 11\}$ can be assumed.

Continuing labelling the boundary triangles, we observe that the edge $\{z_2, z_3\}$ has two possible labels remaining, i.e., either $\{1, 10\}$ or $\{8, 10\}$. Since T_3 is a T-triangle, then z_3 has either label $\{1, 7, 10\}$ or $\{2, 7, 10\}$; the first one only being compatible with one of the possible labels of the edge $\{z_2, z_3\}$. As a consequence, $y_3 = \{2, 7, 10\}$.

Acting similarly on the T-triangle T_5, we get $y_5 = \{2, 5, 11\}$ and $z_5 = \{1, 5, 11\}$.

Finally, a problem occurs in the labelling of the K-triangle T_4: the edges $\{z_3, z_4\}$ and $\{y_4, y_5\}$ have labels $\{2, 10\}$ and $\{1, 11\}$, respectively, preventing any connection between the vertices y_5 and z_5.

A similar reasoning reveals that the presence of exactly one boundary T-triangle does not allow a labelling of the clause gadget. The failure of a labelling in this case is shown in Fig. 11, right (T_1 is the T-triangle).

□

Finally, Fig. 12 shows the possible labellings when either two or no T-triangles are present as boundary triangles.

The final NP-completeness reduction.

Let us consider the instance $C = \{c_1, \ldots, c_n\}$ of MAX-3-SAT involving the variables in the set $U = \{x_1, \ldots x_m\}$. Based on the gadgets already defined, we construct a graph G_C whose labels determine its 2-intersection property and express the desired valuations of C. The reader can follow an example of the construction of G_C in the case $C = (x_1 \vee \overline{x}_1 \vee x_2) \wedge (x_1 \vee x_2 \vee \overline{x}_3)$ in Fig. 13.

First, from Lemma 3, we can suppose that each variable must appear at most three times: one in a form and two in the opposite form. For each variable $x_i \in U$, we define a variable gadget G_{x_i}, and associate the triangle T_3^2 with

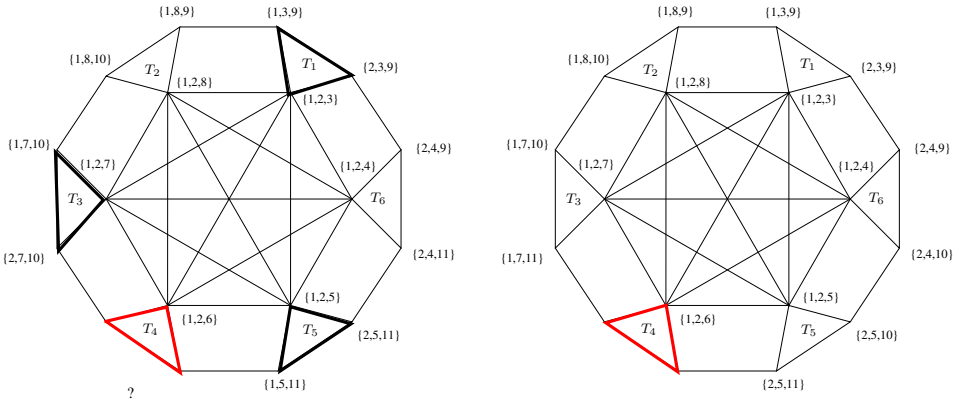

Fig. 11 An attempt to define a labelling of a clause gadget with three (on the left) or one (on the right) boundary T-triangles. Both configurations prevent any labelling of the K-triangle T_4.

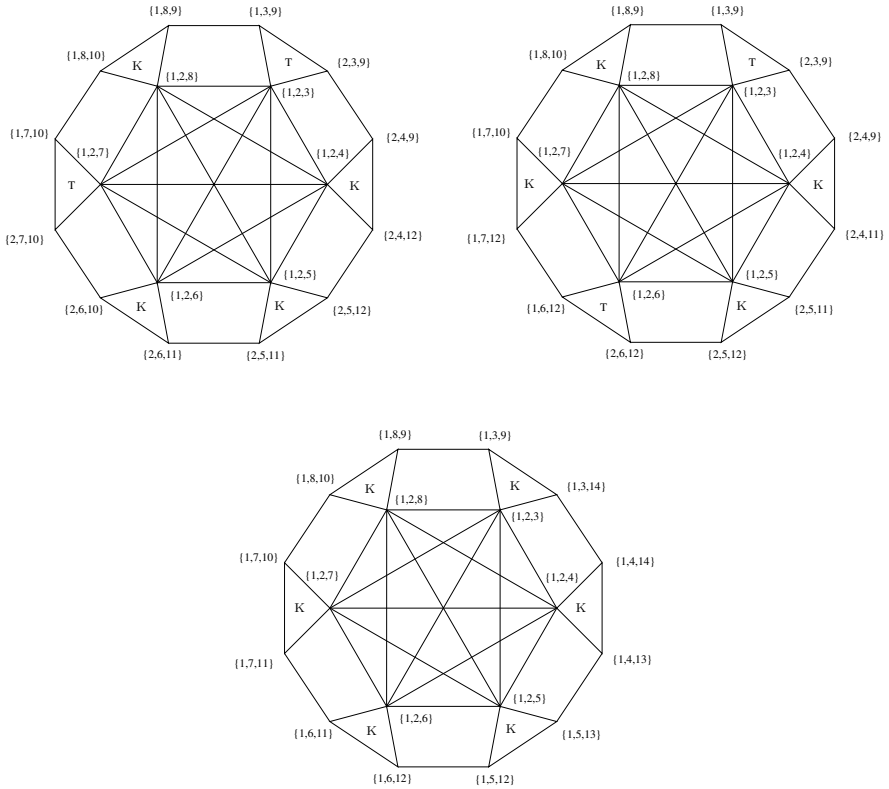

Fig. 12 Reconstruction in the case of zero or two T-triangles. The bold triangles are the boundary triangles.

the single occurrences of a literal x_i. The at-most-two remaining occurrences of the opposite literal are associated with the triangles T_2^1 and T_5^1.

For each clause $c_j \in C$, we construct a clause gadget G_{c_j} and label its boundary triangles $T_1 \dots T_6$ as in Lemma 4.

Finally, we connect variable gadgets and clause gadgets together as follows: for each clause c_j, with $1 \leq j \leq n$, we use a ribbon to the triangle T_{2i-1} of the clause gadget G_c, to the corresponding triangle associated with the ith literal of c in the G_x gadget of its variable, as in Fig. 13.

Now, we prove the main theorem of this section.

Theorem 5 Given an instance C of MAX-3-SAT, the graph G_C has the 2-intersection property if and only if the instance C has a solution.

Proof Let us assume that there exists a valuation for the MAX-3-SAT instance C. Given a variable $x \in U$, for each literal with value *true* associated with x, we assign the triangles associated with it to be T-triangles, and we assign the triangles associated with its negation to be K-triangles. Corollary 4 assures that the variable gadget G_x has a labelling. Then, for each clause $c_j \in C$, in its clause gadget G_{c_j}, we assign the triangles associated with the literals having valuation *true* to be K-triangles, but T-triangles for the literals having valuation *false*. Since three or more T-triangles are not allowed in G_{c_j} by Lemma 4, then also the clause gadget has a labelling. The labelling of the connecting ribbons is straightforward. The reader can benefit from an example of such a construction in Fig. 13.

On the other hand, suppose G_C has the 2-intersection property, i.e., there is a labelling of G_C that determines a 3-hypergraph H such that $I_2(H) = G_C$. For each clause gadget G_{c_j}, with $c_j \in C$, there exists at least one triangle among T_1, T_3 and T_5, say T' of type K, by Lemma 4. Property 4 assures that T' having type K leads to a T-triangle in the variable gadget G_x to which a literal, say l, is associated (note that the opposite does not hold, as shown by the red triangle in Fig. 13). We assign such a literal the truth value *true*. The opposite literal $\bar{l}$ is then associated with *false*. Due to Corollary 4 the triangles (one or two) in G_x associated with $\bar{l}$ are of type K.

We emphasize the following situation: it may happen that there exists a G_x labelling where all three triangles associated with the literals are of type K. In such a case, the value assigned to the variable does not affect the truth value of the clauses of C, so it can be arbitrarily assigned.

So, the valuation defined is a solution of the MAX-3-SAT instance C: each clause gadget G_{c_j} has at most one triangle among T_1, T_3 and T_5 of type K, so at least one literal in the clause has value *true*. Furthermore, in each variable gadget, if one literal is set to *true*, i.e. at least one of the corresponding triangles is of type T, then the the opposite literal has the logical value *false*. □

Corollary 5 The $2INT$ Problem is NP-complete.

The proof directly follows from Theorem 5, after checking that the construction of the graph G_C associated with the instance C can be performed in polynomial time.

Remark 3. In Fig.13, the two highlighted triangles in different colors indicate the following situation:

• the red triangle in the ribbon connecting T_1 of G_{c_1} to T_2^1 of G_{x_3} shows that the T-triangle related to the value *false* in a clause may produce the truth value *true* in the associated literal. This is not a contradiction: it simply means that in the clause, here c_1, there exists at least a second triangle of type K, i.e. the clause is already satisfied by at least one different literal;
• the blue triangle T_1^5 in G_{x_2} is of type K which is different from the T-triangle T_2^1 although they are associated with the same literal l. This means that l may have a value *false* in a second clause in which it is possibly present. As a consequence, a different literal having truth value *true* is present in that clause. What is relevant in G_{x_2} is that the triangle associated with $\bar{l}$, i.e. T_3^2, has to be of type K, assuring that $\bar{l}$ has truth value *false* in all its occurrences.

Example 5. Consider the formula $F = (x_1 \vee \overline{x}_1 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee \overline{x}_3)$. Figure 13 shows the construction of its corresponding gadget. The labelling shown in the same figure corresponds to the valuation $x_1 =$ *false*, $x_3 =$ *false*. The value of x_2 doesn't matter since, in any case, the assignment is a solution for F.

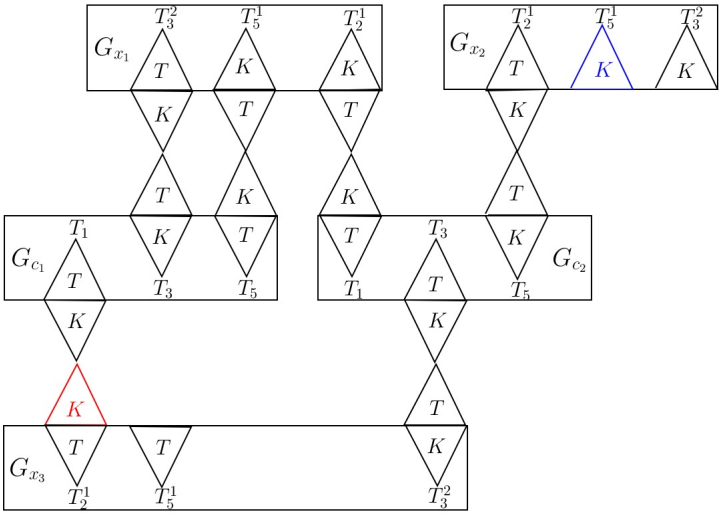

Fig. 13 Example of the construction of the gadget for $C = \{c_1, c_2\}$, $c_1 = (x_3, x_1, \overline{x}_1)$, $c_2 = (\overline{x}_1, \overline{x_3}, x_2)$. One of the valuations obtained by the labelling of the corresponding G_C graph is $x_1 =$*false*[Andrea: true], $x_2 =$*true* and $x_3 =$*true*. The valuation can be obtained by the position of (at least one of) the T configurations in the triangles T_2^1, T_5^1 and T_3^2 of the variable gadget.

References

[1] Behrens, S., Erbes, C., Ferrara, M., Hartke, S.G., Reiniger, B., Spinoza, H., Tomlinson, C.:*New Results on Degree Sequences of Uniform Hypergraphs*, Electron. J. Comb., 20(4), # $P14$ (2013).

[2] Berge, C.: *Hypergraphs*, North-Holland, Amsterdam, (1989).

[3] Brlek, S., Frosini, A.: *A Tomographical Interpretation of a Sufficient Condition for h-Graphical Sequences* DGCI 2016, Lect. Notes in Comput. Sc., 9647, pp. 95 − 104 (2016).

[4] Dewdney, A.K.: *Degree sequences in complexes and hypergraphs*, Proc. Amer. Math. Soc., 53(2), pp. 535 − 540 (1975).

[5] Deza, A., Levin, A., Meesum, S.M., Onn, S.: *Optimization over degree sequences*, SIAM J. Disc. Math. 32(3), pp. 2067 − 2079 (2018).

[6] Di Marco, N., Frosini, A., Kocay, W.L.: *A Study on the Existence of Null Labelling for 3-Hypergraphs* IWOCA 2021, Lect. Notes in Comput. Sc. 12757, pp. 282 − 294 (2021).

[7] Erdös, P., Gallai, T.: *Graphs with prescribed degrees of vertices* (Hungarian), Math. Lapok. 11 pp. 264 − 274 (1960).

[8] Frosini, A., Kocay, W.L., Palma, G., Tarsissi, L.: *On null 3-hypergraphs*, Discrete Appl. Math. 303, pp. 76 − 85 (2021).

[9] Frosini, A., Picouleau, C., Rinaldi, S.: *On the degree sequences of uniform hypergraphs* Lect. Notes in Comput. Sc. 7749, pp. 300 − 311 (2013).

[10] Frosini, A., Picouleau, C., Rinaldi, S.: *New sufficient conditions on the degree sequences of uniform hypergraphs* Theoret. Comput. Sc. 868(8), pp. 97 − 111 (2021).

[11] Garey, M.R., Johnson, D.S.: *Computers and Intractability - A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, NY, USA (1990).

[12] Harary, F.: *Graph Theory*, Addison Wesley Publishing Company (1972).

[13] Herman, G.T., Kuba, A. (Eds.): *Discrete tomography: Foundations algorithms and applications*, Birkhauser, Boston (1999).

[14] Herman, G.T., Kuba, A. (Eds.): *Advances in Discrete Tomography and Its Applications (Applied and Numerical Harmonic Analysis)*, Birkhauser, Boston (2007).

[15] Kocay, W., Li, P.C.: *On* 3-*Hypergraphs with Equal Degree Sequences*, Ars Combinatoria, 82, pp. 145 – 157 (2006)

[16] Kocay, W.: *A Note on Non-reconstructible* 3-*Hypergraphs*, Graphs Combinator. 32, pp. 1945 – 1963 (2016)