www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 16(4), Sep-Dec 2025, 459-472. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2025.v16i4.586

On tractable subclasses of the class 2μ-3MON

Sonia Navarro Flores[1], Carlos Guillén Galván[1]
[1] Facultad de Ciencias Físico Matemáticas, Benemérita Universidad Autónoma de Puebla.
sonianavarroflores91@gmail.com, carlosguillen.galvan@gmail.com

Abstract. The SAT problem is important in the theory of computational complexity. It has been deeply studied because solutions for fragments of SAT can be transformed into solutions for several CSPs, including problems in areas such as Artificial Intelligence and Operations Research. Although SAT is an NP-complete problem, it is known that SAT is fixed-parameter tractable if we take any hypertree width as a parameter. In this work, we present several hypergraphs and countable classes of hypergraphs. For these classes of hypergraphs, we analyze their generalized hypertree width to prove that all the CSPs modeled with those hypergraphs are tractable.
Keywords: Parametrized tractability theory, Constraint Satisfaction Problems, Hypergraphs, Fractional hypertree width, Computational Complexity.

Article Info
Received November 29, 2024
Accepted December 4, 2024

1 Introduction

Constraint Satisfaction Problems (CSPs) consist of finitely many variables with a finite domain and finitely many constraints about the values variables can take simultaneously. These problems are important because several relevant problems from Artificial Intelligence, Operations Research, and Data Base Queries are CSPs.

Generally, CSPs are NP-complete problems and there are many search methods and heuristics to solve them (Tsang, 2014).
It is known that graph acyclicity often guarantees tractability for problems involving graphs. For this reason, several measures of acyclicity have been introduced. From those measures, the tree width is the optimal parameter (Downey & Fellows, 2013). Several problems that are not related to graphs can be modeled with graphs.

For problems modeled with graphs, it is known that having bounded tree width is equivalent to being tractable (Downey & Fellows, 2013). However, most CSPs are better modeled with hypergraphs. Many hypertree widths have been defined to find parameters to make CSPs fixed parameter tractable (Gottlob et al., 2014; Grohe & Marx, 2014).

SAT is a very important problem in computer science. This problem has been studied deeply because the solutions of SAT can be transformed into solutions for other important problems in areas such as real-time verification systems and planning in artificial intelligence. Although SAT is NP-complete, many tractable subclasses have been found by using Parametrized Complexity Theory (Szeider, 2003; Haan & Szeider, 2014).

In this work, we will present some non-trivial tractable subclasses of SAT, and we will analyze their generalized hypertree width to guarantee the tractability of all CSP problems modeled by those hypergraphs. In Section 2 we present preliminar concepts about hypergraphs, the sat problem and hypertree width. In Section 3 we present several hypergraphs and their hypertreewidth. In Subsection 3.1 we include several hypergraphs that were found with the software SageMath, we include their tree width that was computed with SageMath. In Subsection 3.2 we present different methods to define infinite classes of hypergraphs with bounded hypertree width. In Subsection 3.3 we define recursively the class of hypergraphs $\mathcal{B}$. To define the class $\mathcal{B}$, we introduce the operation $\oplus$ on hypergraphs and prove that $ghw(H_1 \oplus H_2) = max\{ghw(H_1), ghw(H_2)\}$ for 1-boundaried hypergraphs H_1 and H_2. This result is interesting because it allows us to produce infinite classes of hypergraphs with bounded hypertree width by gluing hypergraphs with bounded hypertree widths. We also prove that if H_1 and H_2 are 2-boundaried hypergraphs, $max\{ghw(H_1), ghw(H_2)\} \leq ghw(H_1 \oplus H_2) \leq max\{ghw(H_1), ghw(H_2)\} + 1$. This result has many potential applications to produce either infinite classes of hypergraphs with bounded hypertree width or infinite classes of hypergraphs with unbounded hypertree width.

2 Preliminaries

In this section, we present all the necessary concepts to follow the subsequent sections. We include relevant concepts about Hypergraphs, Constraint Satisfaction Problems, the SAT problem and Hypertree widths.

2.1 Hypergraphs

In this Subsection, we present basic definitions of hypergraphs and hypertree decompositions that are crucial to follow the work. The reader interested in knowing more about hypergraphs can see (Bretto, 2013).

Definition 1. A hypergraph is a pair H = (V(H), E(H)) where V(H) is the set of vertices and E(H) includes non-empty subsets of V(H) called edges. Also, every vertex must belong to some edge.

From now on, we identify hypergraphs with the set of its edges. Given a hypergraph H, the primal graph of H is the graph G that satisfies that V(G)= V(H) and a pair of vertices u, v are adjacent in G if and only if there is an edge e in H such that both u and v belong to e. Finally, the size of a hypergraph |H|=|V(H)|+|E(H)|*|V(H)|.

Example 1. Let H be the hypergraph satisfying that V(H)={0, 1, 2, 3, 4, 5, 6, 7, 8} and E(H)={{0, 1, 2}, {0, 3, 4}, {5, 6, 7}, {1, 5, 8}, {2, 3, 6}, {4, 7, 8}}. The primal graph of H is determined by the set of edges {{0, 1}, {0, 2}, {0, 3}, {0, 4}, {1, 2}, {1, 5}, {1, 8}, {2, 3}, {3, 4}, {3, 6}, {4, 7}, {4, 8}, {5, 6}, {5, 7}, {5, 8}, {6, 7}, {7, 8}}. Now we present visualizations of H (see Fig. 1) and its primal graph (see Fig. 2).

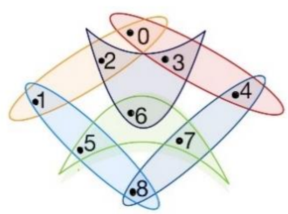

Fig. 1. Hypergraph H.

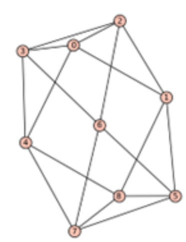

Fig. 2. Primal graph of H.

Definition 2. Let H = (V, E) be a hypergraph, a path P in H from x to y, is a sequence $x_1 e_1 x_1 e_2 \dots e_n x_{n+1}$ such that:

- $x_1 = x$ and $x_{n+1} = y$,
- $x_1, x_2, \dots, x_{n+1}$ are distinct vertices,
- $e_1, e_2, \dots, e_n$ are distinct edges,
- for every $i \in \{1, \dots, n\}$, $x_i, x_{i+1} \in e_i$.

Definition 3. A t-boundaried hypergraph is a hypergraph H with t distinguished vertices labeled by 1,...,t. The vertices labeled by 1,...,t are called the boundary, and we denote by ∂(H) the set of boundary vertices of H.

Definition 4. Let H_1 and H_2 t-boundaried graphs. We denote by $H_1 \oplus H_2$ the hypergraph obtained by taking the disjoint union of H_1 and H_2, and identifying each vertex of ∂(H_1) with the vertex of ∂(H_2) with the same label.

2.2 Constraint Satisfaction Problems

Constrain Satisfaction Problems consist of finitely many variables $x_1, x_2, \dots, x_n$ where each variable x_i has a finite domain, and a set of constraints $\{C_{x_i,\dots,x_k} | i, \dots, k \in 1,2, \dots, n\}$ where $C_{x_i,\dots,x_k}$ is the set of all allowed assignments for variables $x_i, \dots, x_k$. To solve the problem we must find an assignment for all variables satisfying all constraints. Generally, CSPs are NP-complete (Tsang, 2014).

Every CSP has an associated hypergraph, the nodes of the hypergraph are the variables of the CSP, and a set of variables is an edge if there is a constraint over that set of variables. $CSP(H)$ denotes the class of all CSP problems modeled by H.

We borrow the following example from (Gottlob et al., 2014). Solving a crossword puzzle is a CSP if we take every possible letter as a variable, the domain of each variable as the alphabet, and all horizontal or vertical columns correspond to a constraint.

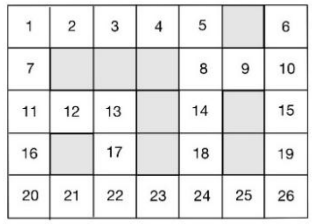

Fig. 3. Crossword puzzle.

In the crossword shown in Fig.3, the variables are $x_1, x_2, \ldots, x_{26}$. The domain of each variable is the alphabet {a, b, c,…,z}. The constraints of the problem are all vertical and horizontal words. Knowing the variables and constraints, we can obtain the hypergraph associated with the CSP.

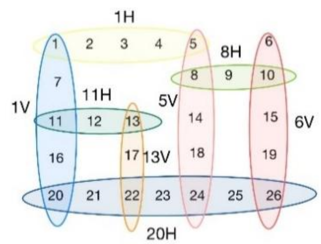

Fig. 4. The hypergraph associated with the crossword puzzle.

2.3 SAT problem

Now, we present some definitions of boolean formulas to define the SAT problem.
A boolean variable is a symbol that can be associated with the values 0 and 1. Boolean formulas are built recursively as follows:

- constants 0 and 1 are Boolean formulas,
- boolean variables are Boolean formulas,
- and conjunctions, disjunctions, and negations of Boolean formulas are Boolean formulas.

We say a boolean formula is satisfiable if an assignation for its variables makes the formula true. The SAT problem consists in deciding whether a quantifier-free boolean formula is satisfiable or not. A literal is whether a boolean variable or a negated Boolean variable. A clause is a disjunction of literals. A clause looks like this $C = l_1 \vee l_2 \vee \ldots \vee l_n$.

Finally, we say that a formula F is in conjunctive normal form (CNF) if it is the conjunction of a finite set of clauses. Those formulas take the form $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$. Every free-quantifier Boolean formula can be expressed as a formula in CNF. We consider all formulas in CNF to state the SAT problem. To do this, given a free-quantifier boolean formula, we consider the variables of the formula as the variables of the problem and the clauses of the formula as the constraints of the problem. The domain of each variable is the set $\{0,1\}$.

A given formula belongs to the class 2μ-3MON if in its FNC each clause has at most 3 variables, none of them are negated, and every variable appears at most in two clauses. The following formula belongs to the class 2μ-3MON, $F = (x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_3 \vee x_5)$. Therefore, hypergraphs associated to this syntactic class satisfy that every edge contains at most 3 vertices and every vertex belongs at most to 2 edges.

2.4 Some widths for hypergraphs

In this Subsection, we present some hypertree widths that have been defined to measure the acyclicity of hypergraphs. To avoid confusion, if T is a tree (decomposition), we will write N(T) to denote the set of vertices of T.

Definition 5. A tree decomposition of a hypergraph H is a pair $\langle T, (B_u)_{u \in N(T)} \rangle$ where T=(N(T), E(T)) is a tree and every node u ∈ N(T) has associated a set of vertices $B_u \subseteq V$ satisfying the following conditions:

1) for every e ∈ E(H) there exists u ∈ N(T) such that $e \subseteq B_u$,
2) for every v ∈ V(H), the set $\{u \in N(T) \mid v \in B_u\}$ is connected in T.

Sets B_u are called bags, and from now on we will identify u with B_u.

The width of a tree decomposition $\langle T, (B_u)_{u \in N(T)} \rangle$ is the cardinality of the bigger bar minus one, that is, $\max\{|B_u|-1 : u \in N(T)\}$. The tree width of H is the minimum of the widths of all the tree decompositions of H.

Example 2. The following tree (see Fig. 5) is a tree decomposition of H. We obtained the tree using SageMath. The tree consists of a path with 5 nodes, the bags of T are the sets {0, 2, 3, 4, 6}, {0, 1, 2, 4, 6}, {0, 1, 4, 6, 8}, {1, 4, 5, 6, 8} and {4, 5, 6, 7, 8}.

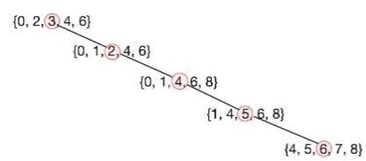

Fig. 5. Tree decomposition of H.

The tree width for hypergraphs is the immediate generalization of the tree width for graphs. For classes of hypergraphs with bounded size, having bounded tree width is the optimal parameter that guarantees tractability (Grohe, 2007).

Though there are classes of hypergraphs with unbounded size satisfying that they have unbounded tree width, but they are tractable. Therefore, the tree width does not capture the acyclicity of hypergraphs precisely. Trying to find better parameters to capture the acyclicity of hypergraphs, other widths for hypergraphs have been defined (Gottlob et al., 1999; Gottlob et al., 2014; Grohe and Marx, 2014).

Now we are going to introduce the concepts of edges cover and hypertree decompositions.

Definition 6. Given a hypergraph H and a set of vertices A ⊆ V(H), an edges cover of A in H is a function c : E(H) →{0,1} such that for every v ∈ A, there exists an edge e containing v such that c(e)=1.

The weight of c is the sum of all the values assigned to all edges by c, that is to say $\sum_{e \in E(H)} c(e)$.

Definition 7. A generalized hypertree decomposition of a hypergraph H is a triple $\langle T, (B_u)_{u \in N(T)}, (c_u)_{u \in N(T)} \rangle$ where:

1) $\langle T, (B_u)_{u \in N(T)} \rangle$ is a tree decomposition of H,
2) For every u ∈ N(T), c_u is an edges cover for B_u in H.

Example 3. For this example, we enumerate the edges of H, the hypergraph mentioned in the previous examples. e_1={0, 1, 2}, e_2={0, 3, 4}, e_3={5, 6, 7}, e_4={1, 5, 8}, e_5={2, 3, 6}, e_6={4, 7, 8}. A generalized hypertree decomposition for H consists of a path with two bags {0, 2, 3, 4, 5, 6, 7},{0, 1, 2, 4, 5, 7, 8}, and the edge covers for those bags {e_2, e_3, e_5}, {e_1, e_4, e_6}. In Figure 6, we present a hypertree decomposition for H.

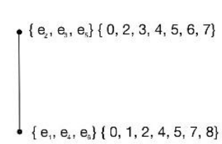

Fig. 6. Hypertree decomposition for H.

The width of a generalized hypertree decomposition is the maximum weight of all the edge covers c_u included in the decomposition. The generalized hypertree width of a hypergraph H is the minimum width of all generalized hypertree decompositions. Grohe and Marx proved that if a class of hypergraphs has bounded (Grohe & Marx, 2014).

It is known that determining the generalized hypertree decomposition of a given hypergraph is an NP-complete problem. Also, given a fixed k, it is NP-complete to determine if $ghw(H) \leq k$ (Gottlob et al., 2021). It follows from different works (Grohe, 2001, 2017; Grohe et al., 2011; Gottlob et al., 1999, 2000) that if a hypergraph H has bounded generalized hypertree width, the problems in CSP(H) are tractable. We are interested in finding tractable subclasses of the syntactic class 2μ-3MON, and for that reason, in this work, we analyze the generalized hypertree width of several hypergraphs and classes of hypergraphs.

3 Hypergraphs and their widths

In this section, we present some hypergraphs and their hypertree widths. Since we are interested in studying the syntactic class 2μ-3MON, all our hypergraphs satisfy the condition that their edges have at most 3 vertices, and each vertex is incident to at most 2 hyperedges.

To calculate the tree width of a hypergraph, we use the function treewidth from SageMath; it's worth mentioning that the algorithm used by this program is not optimal, and for hypergraphs with more than 30 vertices, the computation takes much time. To compute the hypertree width, we use the program provided in the following repository: https://github.com/daajoe/detkdecomp.

We want to determine whether the class 2μ-3MON is fixed-parameter tractable, taking the hypertree width as a parameter. Since we are dealing with a subclass of an NP-complete problem, and for each $k \geq 2$, determining whether the fractional width of a hypergraph is bounded by k is also an NP-complete problem (Gottlob et al., 2021), we have defined some hypergraphs to analyze the behavior of different widths on them. Despite being highly restrictive, the hypertree width (even the tree-width) of this class of hypergraphs is still unknown. The known software for determining tree-width for graphs experiences combinatorial explosion for a small number of vertices and edges. With the objective of contributing to a better understanding of the hypertree width, we include a visualization of the primal graph of each hypergraph. Even though these graphs do not have more than 30 vertices, their visualization becomes confusing.

3.1 Hypergraphs found using SageMath

The hypergraphs presented below were constructed to achieve the maximum possible width with that number of vertices and hyperedges. As we can observe, the maximum tree width we have achieved is 8, and the maximum hypertree width we have reached is 6. We aim to find hypergraphs with a tree width of 9 and a hypertree width of 7. For this purpose, we have defined several hypergraphs with 33 vertices and 22 hyperedges, but so far, we have not achieved the desired widths.

The hypergraph H_1 determined for the set of edges {{0, 1, 2}, {0, 3, 4}, {1, 5, 6}, {3, 7, 8 }, {5, 9, 10}, {7, 9, 14}, {11, 12, 13 }, {9, 7, 14}, {6, 8, 11}, {4, 10, 12}}. There is no other hypergraph with 15 vertices, 10 edges, and greater tree width than H_1. The treewidth of H_1 is 6, and the hypertree width of H_1 is 3. In Figure 7, we present a visualization of the primal graph of H_1.

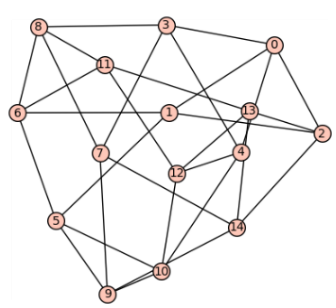

Fig. 7. Primal graph of H_1.

The hypergraph H_2 determined for the set of edges {{0, 1, 2}, {0, 3, 4}, {5, 6, 7}, {5, 8, 9}, {1, 10, 11}, {3, 12, 13}, {14, 15, 16 }, {17, 18, 19}, {20, 10, 14}, {12, 15, 17}, {6, 11, 13}, {2, 8, 18}, {4, 9, 16}, {19, 20, 7}}. There is no other hypergraph with 21 vertices, 14 edges, and greater tree width than H_2. The tree width of H_2 is 7, and the hypertree width of H_2 is 4. In Figure 8, we present a visualization of the primal graph of H_2.

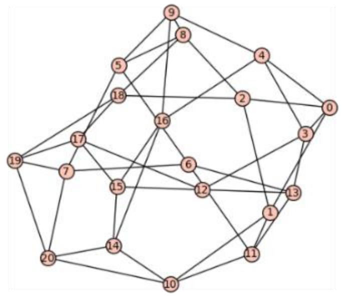

Fig. 8. Primal graph of H_2.

The hypergraph H_3 determined for the set of edges {{0, 1, 2}, {0, 3, 6}, {1, 4, 7}, {2, 5, 8}, {6, 9, 15}, {3, 10, 16}, {4, 12, 18 }, {7, 11, 17}, {5, 13, 20}, {8, 14, 19}, {9, 12, 19}, {11, 15, 20}, {10, 13, 18}, {14, 16, 17}}. H_3 has 21 vertices and 14 edges. The tree width of H_3 is 7, and the hypertree width of H_3 is 5. In Figure 9, we present a visualization of the primal graph of H_3.

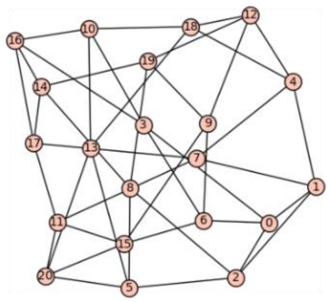

Fig. 9. Primal graph of H_3.

The hypergraph H_4 determined for the set of edges {{0, 1, 2}, {0, 3, 4}, {1, 3, 5}, {6, 7, 8}, {6, 9, 10}, {7, 11, 12}, {9, 13, 14}, {11, 15, 16}, {17, 18, 19}, {20, 21, 22}, {23, 24, 25}, {17, 20, 26}, {13, 18, 23}, {2, 8, 21}, {4, 12, 14}, {10, 15, 24}, {16, 19, 22}, {5, 25, 26}}. There is no other hypergraph with 27 vertices, 18 edges, and greater tree width than H_4. The tree width of H_4 is 8, and the hypertree width of H_4 is 5. In Figure 10, we present a visualization of the primal graph of H_4.

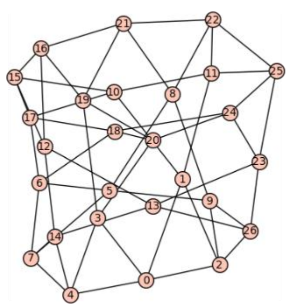

Fig. 10. Primal graph of H_4.

The hypergraph H_5 determined for the set of edges {{0, 1, 2}, {0, 3, 4}, {1, 5, 6}, {2, 7, 8}, {3, 11, 12}, {4, 9, 10}, {5, 13, 14}, {6, 15, 16}, {7, 17, 18}, {8, 19, 20}, {9, 21, 22}, {15, 23, 24}, {19, 25, 26}, {16, 18, 22}, {10, 23, 26}, {14, 21, 25}, {11, 13, 20}, {12, 14, 17}}. H_5 has 27 vertices and 18 edges. The tree width of H_5 is 7, and the hypertree width of H_5 is 5. In Figure 11, we present a visualization of the primal graph of H_5.

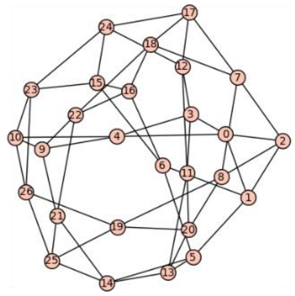

Fig. 11. Primal graph of H_5.

The hypergraph H_6 determined for the set of edges {{0, 1, 2}, {0, 3, 4}, {1, 5, 6}, {2, 7, 8}, {3, 11, 12}, {4, 9, 10}, {5, 13, 14}, {6, 15, 16}, {7, 17, 18}, {8, 19, 20}, {9, 21, 22}, {15, 23, 24}, {19, 25, 26}, {10, 27, 28}, {14, 29, 80}, {20, 31, 32}, {16, 21, 32}, {22, 26, 30}, {11, 29, 31}, {12, 13, 25}, {17, 23, 27}, {18, 24, 28}}. H_6 has 33 vertices and 22 edges. The tree width of H_6 is 6, and the hypertree width of H_6 is 5. In Figure 12, we present a visualization of the primal graph of H_6.

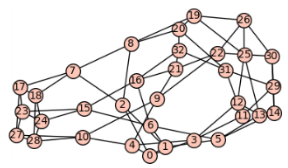

Fig. 12. Primal graph of H_6.

The hypergraph H_7 determined for the set of edges {{0, 1, 2}, {0, 3, 4}, {1, 5, 6}, {2, 7, 8}, {3, 11, 12}, {4, 9, 10}, {5, 13, 14}, {6, 15, 16}, {7, 17, 18}, {8, 19, 20}, {9, 21, 22}, {15, 23, 24}, {19, 25, 26}, {10, 27, 28}, {14, 29, 80}, {20, 31, 32}, {16, 21, 32}, {22, 26, 30}, {11, 29, 31}, {12, 13, 25}, {17, 23, 27}, {18, 24, 28}}. H_7 has 33 vertices and 22 edges. The tree width of H_7 is 8, and the hypertree width of H_7 is 5. In Figure 13, we present a visualization of the primal graph of H_7.

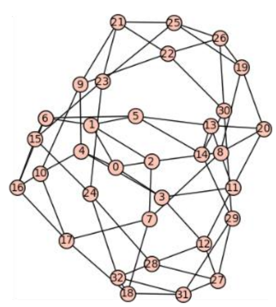

Fig. 13. Primal graph of H_7.

The hypergraph H_8 determined for the set of edges {{0, 1, 2}, {0, 3, 4}, {1, 5, 6}, {2, 7, 8}, {3, 11, 12}, {4, 9, 10}, {5, 13, 14}, {6, 15, 16}, {7, 17, 18}, {8, 19, 20}, {10, 21, 22}, {12, 23, 24}, {14, 25, 26}, {16, 27, 28}, {18, 29, 80}, {20, 31, 32}, {9, 27, 17}, {11, 13, 29}, {15, 19, 21}, {22, 25, 30}, {23, 26, 31}, {24, 28, 32}}. H_8 has 33 vertices and 22 edges. The tree width of H_8 is 8, and the hypertree width of H_8 is 6. In Figure 14, we present a visualization of the primal graph of H_8.

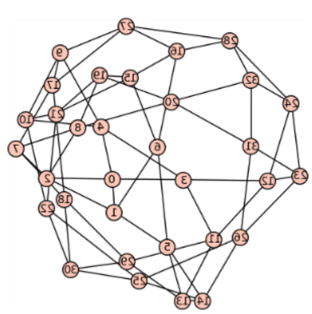

Fig. 14. Primal graph of H_8.

The hypergraph H9 determined for the set of edges {{0, 1, 2}, {0, 3, 4}, {1, 5, 6}, {2, 7, 8}, {3, 11, 12}, {4, 9, 10}, {5, 13, 14}, {6, 15, 16}, {7, 17, 18}, {8, 19, 20}, {10, 21, 22}, {12, 23, 24}, {13, 23, 31}, {14, 25, 26}, {22, 27, 28}, {24, 29, 80}, {26, 31, 32}, {9, 15, 29}, {18, 21, 32}, {16, 19, 27}, {20, 28, 30}, {11, 17, 25}}. H9 has 33 vertices and 22 edges. The tree width of H9 is 7, and the hypertree width of H9 is 5. In Figure 15, we present a visualization of the primal graph of H9.

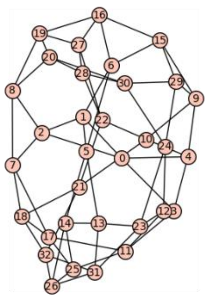

Fig. 15. Primal graph of H9.

3.2 Classes of hypergraphs

Studying different hypergraphs, we found it difficult to define huge hypergraphs by writing all their hyperedges. This is why we found ways to define arbitrary large hypergraphs using an analytic description. In this Subsection, we present some countable families of hypergraphs and prove their tractability. We will start defining the class $\mathcal{HB}$.

For every $n \geq 2$, let H_n be the hypergraph satisfying that $V(H_n) = \{1,2,\dots,6n\}$ and $E(H_n)$ consists of all sets $e_k = \{2k+1, 2k+2, 2k+3\}$ with $k < 3n-1$, also all sets $f_k = \{2k+1, 2k+2n+1, 2k+4n+1\}$ with $k < n-1$. Notice that every hypergraph H_n has $6n$ vertices and $4n$ edges. Let $\mathcal{HB}$ the family of all the hypergraphs H_n. In Figure 16, we present a visualization of the hypergraph H8.

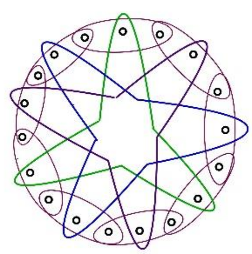

Fig. 16. H_8.

Proposition 1. The family of hypergraphs $\mathcal{HB}$ has bounded generalized hypertree width.

Proof. Fix some $n \geq 3$. We will build a tree decomposition for H_n. Let B_0 be the bag $\{0, 1, 2, 2n, 1+2n, 2+2n, 4n, 1+4n, 2+4n\}$. For every $0 < i < n$, let

$$A_i=\{2i, 2i+1, 2i+2, 2i+2n, 2i+1+2n, 2i+2+2n, 2i+4n, 2i+1+4n, 2i+2+4n\},$$

$C_i= \{2(n-i), 2(n-i)+1, 2(n-i)+2, 2(n-i)+2n, 2(n-i)+1+2n, 2(n-i)+2+2n, 2(n-i)+4n, 2(n-i)+1+4n, 2(n-i)+2+4n\}$. Define $B_{u_i} = A_i \cup C_i$. Notice that for every $i < n$, B_{u_i} meets $B_{u_{i+1}}$. We define the tree T as the path with nodes u_i satisfying that for every $i < n$, u_i is adjacent to u_{i+1}. Notice that edges $e_k, e_{k+n}, e_{k+2n}, f_k \subseteq A_i$ and $e_{n-k}, e_{(n-k)+n}, e_{(n-k)+2n}, f_{n-k} \subseteq C_i$. Therefore, T is a three decomposition of H_n, and its tree width is 6. Therefore $ghw(H_n) \leq 6$.

Theorem 1. For every $H \in \mathcal{HB}$, $CSP(H)$ are tractable.

Proof. It follows from the fact that the family $\mathcal{HB}$ has bounded generalized hypertree width.

Now, we will define another countable class of hypergraphs with bounded generalized hypertree width.

If Σ is an alphabet, Σ^* is the set of all finite strings of elements of Σ. Let Σ_1={0,1,2}, Σ_2 ={0,1} and define Σ= $\Sigma_1\Sigma_2^*$ the concatenation of the sets Σ_1 and Σ_2^*.

For every natural number n with n ≥ 2, we define $V_n = \{w: |w| \leq n\}$. Notice that

$$|V_n|=\sum_{i=1}^{n} |\{ w \in \Sigma : |w| = i\}| = \sum_{i=1}^{n} 3(2^{i-1}) = 3(\sum_{j=0}^{n-1} 2^j) = 3(\frac{1-2^n}{1-2}) = 3(2^n - 1). \quad \textbf{(1)}$$

Now fix some $n \geq 2$. We define the edges for our hypergraph as follows. Recall that ϵ denotes the empty string, let e_ϵ ={0,1,2}. For every $v \in V_n$ with $|v| < n$ let $e_v = \{v, v0, v1\}$. Notice that there are $3(2^{n-1}$-1) many different sets e_w. Now, for every $w \in \Sigma_2^{n-1}$, let $f_w = \{0w, 1w, 2w\}$. Notice that there are 2^{n-1} many different sets f_w.

Let E_n be the collection of all edges $\{e_\epsilon\} \cup \{e_v: v \in V_n, |v| < n\} \cup \{f_w: w \in \Sigma_2^{n-1}\}$.

Notice that there are $1 + 3(2^{n-1} - 1) + 2^{n-1}$ edges, and

$$1 + 3(2^{n-1} - 1) + 2^{n-1} = 1 - 3 + 3(2^{n-1}) + 2^{n-1} = 4(2^{n-1}) - 2 = 2(2^n - 1). \quad \textbf{(2)}$$

So we define $HT_n = (V_n, E_n)$. First, we will see that every vertex belongs to exactly two edges. Take $v \in V_n$. If $|v| = 1$, $v \in \{0,1,2\}$ and $v \in e_v$. If $1 < |v| < n$, we can write $v = wi$ with $i \in \Sigma_2$. Then $v \in e_v$ and $v \in e_w$. If $|v| = n$, we can write $v = wi = jt$ with $i \in \Sigma_2$ and $j \in \Sigma_1$. Then $v \in e_w$ and $v \in f_t$.

Let $\mathcal{HT}$ be the family of all hypergraphs HT_n where n is a natural number greater than 2.

Proposition 2. The family $\mathcal{HT}$ has bounded hypertree width. Furthermore, for every $HT_n \in \mathcal{HT}$, the respective generalized hypertree width is at most 3.

Proof. We will build an hypertree decomposition for HT_n. Let $B_\epsilon = \{v \in V_n: |v| \leq 2\}$ and notice that $B_\epsilon = e_\epsilon \cup e_0 \cup e_1 \cup e_2 = e_0 \cup e_1 \cup e_2$. Therefore we define $c_\epsilon = \{e_0, e_1, e_2\}$. Now, for every $i \in \{1, \dots, n-2\}$ and for every $t \in \Sigma_2^i$,

$$B_t = \{0t, 0t0, 0t1, 1t, 1t0, 1t1, 2t, 2t0, 2t1\} = e_{0t} \cup e_{1t} \cup e_{2t}.$$

Therefore we define $\lambda_t = \{e_{0t}, e_{1t}, e_{2t}\}$. We must verify that every hyperedge f_w is contained in some bag. To do this, fix some $w \in \Sigma_2^{n-1}$. So, $f_w = \{0w, 1w, 2w\} \subset B_w$. Then we define T as the tree with nodes $N = \Sigma_2^{\leq n}$ satisfying that $s, t \in N$ are adjacent in T if s is a prefix of t or t is a prefix of s.

To see that $\langle T, (B_u)_{u \in N(T)} \rangle$ is a tree decomposition of HT_n we must prove that for every $v \in V_n$, $\{t \in N: v \in B_t\}$ is connected in T. To do this, fix $v \in V_n$.

- If $|v| = 1$, v only belongs to the bag B_ϵ and v does not belong to other B_t.
- If $|v| = 2$, we can write $v = ij$ with $i \in \Sigma_1, j \in \Sigma_2$. Then v only belongs to the bags B_ϵ and B_j.
- If $2 < |v| < n$, $v = it = isj$ with $i \in \Sigma_1$ and $j \in \Sigma_2$. Then $v \in B_t$ and $v \in B_s$. Since *s* is a prefix of *t*, *s* and *t* are adjacent.
- If $|v| = n$, we can write $v = itj$ with $i \in \Sigma_1$, $t \in \Sigma_2^{n-2}$ and $j \in \Sigma_2$. Then v only belongs to the bag B_t.

Therefore, the triple $\langle T, (B_t)_{t \in N(T)}, (\lambda_t)_{t \in N(T)} \rangle$ is a generalized hypertree decomposition and its hypertree width is equal to 3. So, for every $n > 2$, $ghw(HT_n) \leq 3$.

Theorem 2. For every $H \in \mathcal{HT}$, *CSP(H)* are tractable.
Proof. It follows from the fact that the family $\mathcal{HT}$ has bounded generalized hypertree width.

3.3 Building a class of hypergraphs recursively

Now, we will present another family of hypergraphs. These hypergraphs are built by gluing some basic hypergraphswith the operation $\oplus$, and with this procedure, we can obtain countably many different hypergraphs with bounded hypertree width. First we prove some connections between the hypertreewidth of $H_1 \oplus H_2$ and hypertree widths of H_1 and H_2.

Theorem 3. Given H_1 and H_2 1-boundaried hypergraphs, $ghw(H_1 \oplus H_2) = max\{ghw(H_1), ghw(H_2)\}$.

Proof. Let H_1 and H_2 1-boundaried hypergraphs with $\partial(H_1) = \{u\}$ and $\partial(H_2) = \{v\}$. Let T_1, T_2 the disjoint hypertree decompositions of hypegraphs H_1 and H_2 $satisfyng\ that$ the width of T_1 is $ghw(H_1)$ and the width of T_2 is $ghw(H_2)$. There exist $s \in T_1$ and $t \in T_2$ satisfying that $u \in B_s$ and $v \in B_t$.

Define T as follows, $N(T) = N(T_1) \cup N(T_2)$ and $E(T) = E(T_1) \cup E(T_2) \cup \{st\}$. Notice that vertices s and t are adjacent in T, we add this edge to guarantee that in T, all the bags containing $u \sim v$ are connected. Since $\langle T_1, (B_u)_{u\in N(T_1)}\rangle$ and $\langle T_2, (B_u)_{u\in N(T_2)}\rangle$ are tree decomposicions for H_1 and H_2 respectively, $\langle T, (B_u)_{u\in N(T_1)\cup N(T_2)}\rangle$ is a tree decomposition of $H_1 \oplus H_2$. Hence, $\langle T, (B_u)_{u\in N(T_1)\cup N(T_2)}, (c_u)_{u\in N(T_1)\cup N(T_2)}\rangle$ is a generalized hypertree decomposition for $H_1 \oplus H_2$. Since all the bags for T are bags in T_1 or T_2, the width of T is $max\{ghw(H_1), ghw(H_2)\}$. Therefore $ghw(H_1 \oplus H_2) \leq max\{ghw(H_1), ghw(H_2)\}$.

Finally, since $H_1 \oplus H_2$ contains the hypergraphs H_1 and H_2, $ghw(H_1) \leq ghw(H_1 \oplus H_2)$ and $ghw(H_2) \leq ghw(H_1 \oplus H_2)$. Therefore, $ghw(H_1 \oplus H_2) = max\{ghw(H_1), ghw(H_2)\}$.

Theorem 4. Given H_1 and H_2 2-boundaried hypergraphs, it holds that $max\{ghw(H_1), ghw(H_2)\} \leq ghw(H_1 \oplus H_2) \leq max\{ghw(H_1), ghw(H_2)\} + 1$.

Proof. Let H_1 and H_2 2-boundaried hypergraphs with $\partial(H_1) = \{u, v\}$ and $\partial(H_2) = \{w, z\}$. Let T_1, T_2 the disjoint hypertree decompositions of hypegraphs H_1 and H_2 satisfying that the width of T_1 is $ghw(H_1)$ and the width of T_2 is $ghw(H_2)$. There exist $s \in T_1$ and $t \in T_2$ satisfying that $u \in B_s$ and $w \in B_t$.

Define T as follows, $N(T) = N(T_1) \cup N(T_2)$ and $E(T) = E(T_1) \cup E(T_2) \cup \{st\}$. Notice that t is a tree. Now take $p \in T_1$ and $q \in T_2$ satisfying that $v \in B_p$ and $z \in B_q$. Since T is a tree, there exists a path $ve_1v_1 \dots v_{n-1}e_nz$ in T. Every edge e_i is contained in a bag B_{r_i} with $r_i \in N(T_1) \cup N(T_2)$. Define $B'_{r_i} = B_{r_i} \cup \{v\}$. We do this to guarantee that all the bags is T that contains $v \sim z$ are connected. For every $r \in (N(T_1) \cup N(T_2)) \setminus \{r_i: 1 \leq i \leq n\}$ define $B'_r = B_r$.

Notice that vertices s and t are adjacent in T, we add this edge to guarantee that in T, all the bags containing $u \sim w$ are connected. Since $\langle T_1, (B_u)_{u\in N(T_1)}\rangle$ and $\langle T_2, (B_u)_{u\in N(T_2)}\rangle$ are tree decomposicions for H_1 and H_2 respectively, $\langle T, (B'_u)_{u\in N(T_1)\cup N(T_2)}\rangle$ is a tree decomposition of $H_1 \oplus H_2$. Now, for every $1 \leq i \leq n$, let $c'_{r_i} = c_{r_i} \cup \{e_1\}$ and notice that c'_{r_i} is an edge cover for B_{r_i}. Also, for $r \in (N(T_1) \cup N(T_2)) \setminus \{r_i: 1 \leq i \leq n\}$ define $c'_r = c_r$. Hence, $\langle T, (B'_u)_{u\in N(T_1)\cup N(T_2)}, (c'_u)_{u\in N(T_1)\cup N(T_2)}\rangle$ is a generalized hypertree decomposition for $H_1 \oplus H_2$. Since all the bags for T are bags in T_1 or T_2 that could have an additional edge, the width of T is $max\{ghw(H_1) + 1, ghw(H_2) + 1\}$. Therefore $ghw(H_1 \oplus H_2) \leq max\{ghw(H_1) + 1, ghw(H_2) + 1\}$.

Finally, since $H_1 \oplus H_2$ contains the hypergraphs H_1 and H_2, $ghw(H_1) \leq ghw(H_1 \oplus H_2)$ and $ghw(H_2) \leq ghw(H_1 \oplus H_2)$. Therefore, $max\{ghw(H_1), ghw(H_2)\} \leq ghw(H_1 \oplus H_2) \leq max\{ghw(H_1), ghw(H_2)\} + 1$.

It follows from Theorem 4 that $ghw(H_1 \oplus H_2)$ could be either $max\{ghw(H_1), ghw(H_2)\}$ or $max\{ghw(H_1), ghw(H_2)\} + 1$. This is very interesting because we could use the operation $\oplus$ to produce bigger hypergraphs with either bounded hypertree width or unbounded hypertree width. Hence, it is desirable to know which properties on H_1 and H_2 guarantee that $ghw(H_1 \oplus H_2)$ is equal to $max\{ghw(H_1), ghw(H_2)\}$ and wich conditions guarantee that $ghw(H_1 \oplus H_2)$ is equal to $max\{ghw(H_1), ghw(H_2)\} + 1$. The following theorems give us conditions to gruarantee that $ghw(H_1 \oplus H_2) = max\{ghw(H_1), ghw(H_2)\}$.

Theorem 5. Given H_1 and H_2 2-boundaried hypergraphs such that $\partial(H_1) = \{u, v\} \subseteq e_1$ and $\partial(H_2) = \{w, z\} \subseteq e_2$ with $e_1 \in E(H_1)$ and $e_2 \in E(H_2)$. Then $ghw(H_1 \oplus H_2) = max\{ghw(H_1), ghw(H_2)\}$.

Proof. Let H_1 and H_2 2-boundaried hypergraphs with $\partial(H_1) = \{u, v\}$ and $\partial(H_2) = \{w, z\}$ $as\ above$. Let T_1, T_2 the disjoint hypertree decompositions of hypegraphs H_1 and H_2 $satisfyng\ that$ the width of T_1 is $ghw(H_1)$ and the width of T_2 is $ghw(H_2)$. There exist $s \in T_1$ and $t \in T_2$ satisfying that $e_1 \subseteq B_s$ and $e_2 \subseteq B_t$.

Define T as follows, $N(T) = N(T_1) \cup N(T_2)$ and $E(T) = E(T_1) \cup E(T_2) \cup \{st\}$. Notice that vertices s and t are adjacent in T, we add this edge to guarantee that in T, all the bags containing $u \sim w$ are connected, the same for all the bags containing $v \sim z$. Since $\langle T_1, (B_u)_{u \in N(T_1)} \rangle$ and $\langle T_2, (B_u)_{u \in N(T_2)} \rangle$ are tree decomposicions for H_1 and H_2 respectively, $\langle T, (B_u)_{u \in N(T_1) \cup N(T_2)} \rangle$ is a tree decomposition of $H_1 \oplus H_2$. Hence, $\langle T, (B_u)_{u \in N(T_1) \cup N(T_2)}, (c_u)_{u \in N(T_1) \cup N(T_2)} \rangle$ is a generalized hypertree decomposition for $H_1 \oplus H_2$. Since all the bags for T are bags in T_1 or T_2, the width of T is $max\{ghw(H_1), ghw(H_2)\}$. Therefore $ghw(H_1 \oplus H_2) \leq max\{ghw(H_1), ghw(H_2)\}$.

Finally, since $H_1 \oplus H_2$ contains the hypergraphs H_1 and H_2, $ghw(H_1) \leq ghw(H_1 \oplus H_2)$ and $ghw(H_2) \leq ghw(H_1 \oplus H_2)$. Therefore, $ghw(H_1 \oplus H_2) = max\{ghw(H_1), ghw(H_2)\}$.

Now we define the following hypergraphs.

Let I be the hypergraph satisfying that $V(I) = \{1,2,3,4,5,6,7,8\}$ and $E(I) = \{\{1,2,3\}, \{2,4,6\}, \{3,5,7\}, \{4,5,8\}, \{6,7,8\}\}$. Notice that I has order 8, $d_I(1) = 1$ and for $i > 1$, $d_I(i) = 2$. In Figure 17, we present a visualization of the hypergraph I.

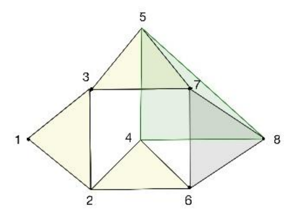

Fig. 17. Hypergraph I.

Let us denote $e_1 = \{1,2,3\}$, $e_2 = \{2,4,6\}$, $e_3 = \{3,5,7\}$, $e_4 = \{4,5,8\}$, $e_5 = \{6,7,8\}$. Let T_I be the tree with nodes s_1, s_2, s_3, and $B_{s_1} = \{1,2,3\} = e_1$, $B_{s_2} = \{2,3,4,5,6,7\} = e_2 \cup e_3$, and $B_{s_3} = \{4,5,6,7,8\} = e_4 \cup e_5$.

Notice that B_{s_1} and $B_{s_3} are\ adjacent\ to\ B_{s_2}$ so T_I is a path. Also, T_I is a hypertree decomposition of I and since every bag contains at most two edges, $ghw(I) \leq 2$. Furthermore, since I contains cycles, $ghw(I) = 2$.

Let J be the hypergraph satisfying that $V(J) = \{1,2,3,4,5,6,7,8,9,10,11\}$, and $E(J) = \{\{1,2,3\}, \{2,4,5\}, \{3,4,6\}, \{5,7,9\}, \{6,8,10\}, \{7,8,11\}, \{9,10,11\}\}$. Notice that J has order 11, $d_J(1) = 1$ and for $i > 1$, $d_J(i) = 2$. In Figure 18, we present a visualization of the hypergraph J.

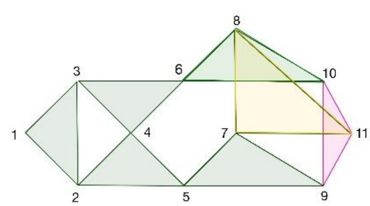

Fig. 18. Hypergraph J.

Let us denote $e_1 = \{1,2,3\}$, $e_2 = \{2,4,5\}$, $e_3 = \{3,4,6\}$, $e_4 = \{5,7,9\}$, $e_5 = \{6,8,10\}$, $e_6 = \{7,9,11\}$, $e_7 = \{9,10,11\}$. Let T_J be a path with nodes t_1, t_2, t_3, t_4 which is a path and $B_{t_1} = \{1,2,3\} = e_1$, $B_{t_1} = \{1,2,3\} = e_1$, $B_{t_2} = \{2,3,4,5,6\} = e_2 \cup e_3$, $B_{t_3} = \{5,6,7,8,9,10\} = e_4 \cup e_5$, $B_{t_3} = \{5,6,7,8,9,10\} = e_4 \cup e_5$, $B_{t_4} = \{7,8,9,10,11\} = e_6 \cup e_7$.

Notice that T_J is a hypertree decomposition of J and since every bag contains at most two edges, $ghw(J) \leq 2$. Furthermore, since J contains cycles, $ghw(J) = 2$.

Let K be the hypergraph satisfying that $V(K) = \{1,2,3,4,5,6,7\}$ and $E(K) = \{\{1,2,3\}, \{2,4,5\}, \{3,4,6\}, \{5,6,7\}\}$. Notice that K has order 7, $d_K(1) = 1$, $d_K(7) = 1$ and for $1 < i < 7$, $d_K(i) = 2$. In Figure 19, we present a visualization of the hypergraph K.

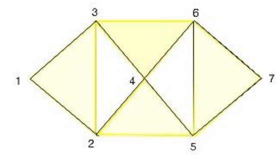

Fig. 19. Hypergraph K.

Let us denote $e_1 = \{1,2,3\}$, $e_2 = \{2,4,5\}$, $e_3 = \{3,4,6\}$, $e_4 = \{5,6,7\}$. Let T_K be a path with nodes u_1, u_2, u_3 and define $B_{u_1} = \{1,2,3\} = e_1$, $B_{t_2} = \{2,3,4,5,6\} = e_2 \cup e_3$, and $B_{t_3} = \{5,6,7\} = e_4$. Notice that T_K is a hypertree decomposition of K and since every bag contains at most two edges, $ghw(K) \leq 2$. Furthermore, since K contains cycles, $ghw(K) = 2$.

Let L be the hypergraph satisfying that $V(L) = \{1,2,3,4,5,6,7,8,9,10\}$ and

$$E(L) = \{\{1,2,3\}, \{2,4,7\}, \{3,5,6\}, \{4,5,8\}, \{6,7,9\}, \{8,9,10\}\}.$$

Notice that L has order 10, $d_L(1) = 1$, $d_L(10) = 1$, and for $1 < i < 10$, $d_L(i) = 2$. In Figure 20, we present a visualization of the hypergraph L.

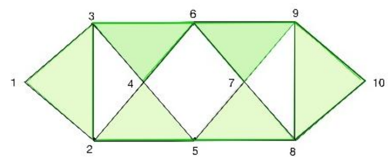

Fig. 20. Hypergraph L.

Let us denote $e_1 = \{1,2,3\}$, $e_2 = \{2,4,7\}$, $e_3 = \{3,5,6\}$, $e_4 = \{4,5,8\}$, $e_5 = \{6,7,9\}$, $e_6 = \{8,9,10\}$. Let T_L be a path with nodes v_1, v_2, v_3, v_4 and define $B_{v_1} = \{1,2,3\} = e_1$, $B_{v_2} = \{2,3,4,5,6,7\} = e_2 \cup e_3$, $B_{v_3} = \{4,5,6,7,8,9\} = e_4 \cup e_5$, $B_{v_4} = \{8,9,10\} = e_6$. Notice that T_L is a hypertree decomposition of L and since every bag contains at most two edges, $ghw(L) \leq 2$. Furthermore, since L contains cycles, $ghw(L) = 2$.

Let M be the hypergraph satisfying that $V(M) = \{1,2,3,4,5,6,7,8,9,10\}$ and $E(M) = \{\{1,2,3\}, \{2,4,5\}, \{3,5,6\}, \{4,7,8\}, \{6,7,9\}, \{8,9,10\}\}$. Notice that M has order 10, $d_M(1) = 1$, $d_M(10) = 1$ and for $1 < i < 10$, $d_M(i) = 2$. In Figure 21, we present a visualization of the hypergraph M.

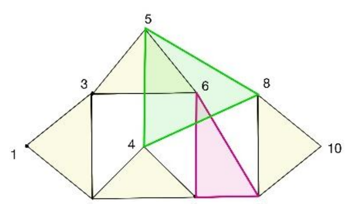

Fig. 21. Hypergraph M.

Let us denote $e_1 = \{1,2,3\}$, $e_2 = \{2,4,5\}$, $e_3 = \{3,5,6\}$, $e_4 = \{4,7,8\}$, $e_5 = \{6,7,9\}$, $e_6 = \{8,9,10\}$. Let T_M be the tree with nodes w_1, w_2, w_3, w_4 which is a path and $B_{w_1} = \{1,2,3\} = e_1$, $B_{w_2} = \{2,3,4,5,6,7\} = e_2 \cup e_3$, $B_{w_3} = \{4,5,6,7,8,9\} = e_4 \cup e_5$, $B_{w_4} = \{8,9,10\} = e_6$.

Notice that T_M is a hypertree decomposition of M and since every bag contains at most two edges, $ghw(M) \leq 2$. Furthermore, since M contains cycles, $ghw(M) = 2$.

We will use hypergraphs I, J, K, L, M to build countable many hypergraphs with bounded hypertree width. We will consider hypergraphs I, J, K, L, M as 1-boundary hypergraphs where $\partial(I) = \{1\}$, $\partial(J) = \{1\}$, $\partial(K) = \{1\}$, $\partial(L) = \{1\}$, $\partial(M) = \{1\}$. When we glue two hypergraphs $H_1 \oplus H_2$ where $H_1 \in \{I, J\}$ we identify 1^{H_1} and 1^{H_2} so 1 has degree 2 in $H_1 \oplus H_2$. Hence, if $H_2 \in \{I, J\}$ all the vertices in $H_1 \oplus H_2$ have degree 2. Otherwise, there exists exactly one vertex in $H_1 \oplus H_2$ with degree 1. In the last case we enumerate the vertices such that the only vertex with degree 1 is labeled with numer 1 and take $\partial(H_1 \oplus H_2) = \{1\}$.

Let $\mathcal{B}$ the class of all hypergraphs $H = H_1 \oplus \ldots \oplus H_n$ where $H_1 \in \{I, J\}$ and and $H_i \in \{I, J, K, L, M\}$ for $1 < i$. Notice that every $H \in \mathcal{B}$ contains zero or one vertex with degree 1 and all other vertices has degree 2.

Proposition 3. For every $H \in \mathcal{B}$, it holds that $ghw(H) = 2$.

Proof. We will prove it for induction on the length of H. Since $H \in \mathcal{B}$, $H = H_1 \oplus \ldots \oplus H_n$.

If n =2. Since H_1 and H_2 are 1-boundaried hypergraphs, it follows from Theorem 3 that $ghw(H) = max\{ghw(H_1), ghw(H_2)\} = 2$.

Now we assume that the conclusion holds for *n* and we will prove that the conclusion holds for *n+1*. Let $H = H_1 \oplus \ldots \oplus H_n \oplus H_{n+1}$. By inductive hypothesis $\partial(H_1 \oplus \ldots \oplus H_n) = \{1\}$, and $ghw(H_1 \oplus \ldots \oplus H_n) = 2$. Since $H_1 \oplus \ldots \oplus H_n$ and H_{n+1} are 1-boundaried hypergraphs, if follows from Theorem 3 that $ghw(H) = max\{ghw(H_1 \oplus \ldots \oplus H_n), ghw(H_{n+1})\} = 2$.

Theorem 6. For every $H \in \mathcal{HT}$, *CSP(H)* are tractable.

Proof. It follows from the fact that the family $\mathcal{HT}$ has bounded generalized hypertree width.

Figure 22 illustrates a member of $\mathcal{B}$.

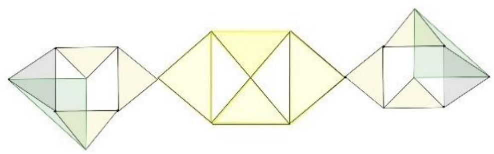

Fig. 22. Hypergraph $L \oplus K \oplus L$.

4 Conclusions

Observing the behavior of tree width in the analyzed examples, we conjecture that none of the widths are bounded on the class of 2μ-3MON. As we have dealt with NP-complete problems, the known algorithms that calculate tree-width and hyper-tree-width are not useful when dealing with very large hypergraphs. For this reason, the hypergraphs for which we can compute their tree width with software have at most 33 vertices. Since it is very difficult to define large hypergraphs by listing all their hyper-edges, in this work, we present different ways to define infinite classes of hypergraphs. To build the class of hypergraphs $\mathcal{B}$, we use the operation $\oplus$ to build new hypergraphs, gluing other hypergraphs. This technique is very useful to produce classes of hypergraphs with unbounded size. We do not know yet conditions on hypergraphs H_1 and H_2 guarantee that $ghw(H_1 \oplus H_2)$ is equal to $max\{ghw(H_1), ghw(H_2)\} + 1$. We are interested in finding those conditions because that information will allow us to build tractable classes of hypergraphs.

References

Bretto, A. (2013). *Hypergraph theory. An introduction.* Springer. https://doi.org/10.1007/978-3-319-00080-0

Courcelle, B., & Engelfriet, J. (2012). Graph structure and monadic second-order logic: a language-theoretic approach. *Cambridge University Press*, 138.

de Haan, R., & Szeider, S. (2014). Fixed-parameter tractable reductions to SAT. In *International Conference on Theory and Applications of Satisfiability Testing*, 85–102. Springer International Publishing. https://doi.org/10.1007/978-3-319-09284-3_8

Diestel, R. (2017). *Graph theory*. Springer.

Downey, R. G., & Fellows, M. R. (2013). *Fundamentals of parameterized complexity* (4). Springer. https://doi.org/10.1007/978-1-4471-5559-1

Fischl, W., Gottlob, G., & Pichler, R. (2018). General and fractional hypertree decompositions: Hard and easy cases. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, 17–32. ACM. https://doi.org/10.1145/3196959.3196962

Gottlob, G., Greco, G., & Scarcello, F. (2014). Treewidth and hypertree width. In Lucas Bordeaux, Youssef Hamadi & pushmeet Kohli (Eds.), *Tractability: Practical Approaches to Hard Problems*, (pp. 3-34). Cambridge University Press.

Gottlob, G., Lanzinger, M., Longo, D. M., Okulmus, C., & Pichler, R. (2020). The hypertrac project: Recent progress and future research directions on hypergraph decompositions. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research* (pp. 3–21). Springer International Publishing. https://doi.org/10.1007/978-3-030-58942-4_1

Gottlob, G., Lanzinger, M., Okulmus, C., & Pichler, R. (2024). Fast parallel hypertree decompositions in logarithmic recursion depth. *ACM Transactions on Database Systems*, *49*(1), 1-43. https://doi.org/10.1145/3638758

Gottlob, G., Lanzinger, M., Pichler, R., & Razgon, I. (2021). Complexity analysis of generalized and fractional hypertree decompositions. *Journal of the ACM (JACM)*, 68(5), 1–50. https://doi.org/10.1145/3457374

Gottlob, G., Leone, N., & Scarcello, F. (1999). Hypertree decompositions and tractable queries. In *Proceedings of the Eighteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (pp. 21–32). ACM. https://doi.org/10.1145/303976.303979

Gottlob, G., Leone, N., & Scarcello, F. (2000). A comparison of structural CSP decomposition methods. *Artificial Intelligence*, 124(2), 243–282. https://doi.org/10.1016/S0004-3702(00)00078-3

Grohe, M. (2007). The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM (JACM)*, 54(1), 1–24. https://doi.org/10.1145/1206035.1206036

Grohe, M., & Marx, D. (2014). Constraint solving via fractional edge covers. *ACM Transactions on Algorithms (TALG)*, 11(1), 1–20. https://doi.org/10.1145/2636918

Guillén Galván, C., Lemuz López, R., & Ayaquica Martínez, I. O. (2013). Model counting in the 2mu-3MON syntactic class. *Computación y Sistemas*, 17(4), 501–513.

Szeider, S. (2003). On fixed-parameter tractable parameterizations of SAT. In *International Conference on Theory and Applications of Satisfiability Testing* (pp. 188–202). Berlin, Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-24605-3_15

Tsang, E. (2014). *Foundations of constraint satisfaction: The classic text*. BoD–Books on Demand.