



HAL
open science

On the complexity of the representation of simplicial complexes by trees

Jean-Daniel Boissonnat, Dorian Mazauric

► **To cite this version:**

Jean-Daniel Boissonnat, Dorian Mazauric. On the complexity of the representation of simplicial complexes by trees. [Research Report] RR-8647, Inria Sophia Antipolis; INRIA. 2014. hal-01089846v1

HAL Id: hal-01089846

<https://inria.hal.science/hal-01089846v1>

Submitted on 4 Dec 2014 (v1), last revised 4 Sep 2015 (v2)

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.



On the complexity of the representation of simplicial complexes by trees

Jean-Daniel Boissonnat, Dorian Mazauric

**RESEARCH
REPORT**

N° 8647

December 2014

Project-Team Geometrica



On the complexity of the representation of simplicial complexes by trees*

Jean-Daniel Boissonnat[†], Dorian Mazauric[†]

Project-Team Geometrica

Research Report n° 8647 — December 2014 — 21 pages

Abstract: In this paper, we investigate the problem of the representation of simplicial complexes by trees. We introduce and analyze local and global tree representations. We prove that the global tree representation is more efficient in terms of time complexity for searching a given simplex and we show that the local tree representation is more efficient in terms of size of the structure. The simplicial complexes are modeled by hypergraphs. We then prove that the associated combinatorial optimization problems are very difficult to solve and to approximate even if the set of maximal simplices induces a cubic graph, a planar graph, or a bounded degree hypergraph. However, we prove polynomial time algorithms that compute constant factor approximations and optimal solutions for some classes of instances.

Key-words: simplicial complexes, hypergraphs, tree representations, NP-complete, approximation algorithms.

* This research has been partially supported by the European Research Council under Advanced Grant 339025 GUDHI (Algorithmic Foundations of Geometric Understanding in Higher Dimensions).

[†] Inria Sophia Antipolis - Méditerranée, France

**RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE**

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

De la difficulté de représenter des complexes simpliciaux par des arbres

Résumé : Dans cet article, nous nous intéressons au problème de la représentation des complexes simpliciaux par des arbres. Nous introduisons et analysons une représentation locale et une représentation globale. Nous prouvons que la représentation globale est la plus efficace en termes de complexité en temps pour la recherche d'un simplexe et nous montrons que la représentation locale est la plus efficace en termes de taille de structure. Les complexes simpliciaux sont modélisés par des hypergraphes. Nous prouvons ensuite que les problèmes d'optimisation combinatoire associés sont très difficiles à résoudre voire à approximer même lorsque l'ensemble des simplexes maximaux induit un graphe cubique, un graphe planaire ou hypergraphe de degré borné. Nous proposons cependant des algorithmes polynomiaux exacts et d'approximation pour certaines classes d'instances.

Mots-clés : complexes simpliciaux, hypergraphes, représentations par des arbres, NP-complet, algorithmes d'approximation.

Contents

1	Introduction	4
2	Local and global tree representations of hypergraphs	6
2.1	Need for additional constraints	6
2.2	Definitions and equivalences	7
2.3	Efficient simplex search, remove, and add algorithms	10
2.4	Combinatorial optimization problems	11
2.5	Comparison between the three tree representations	12
3	Computing optimal tree representations is difficult	14
3.1	Tree representations of graphs	14
3.2	Tree representations of bounded degree hypergraphs	16
3.2.1	A polynomial time algorithm for hypergraphs of degree two	16
3.2.2	NP-completeness results for hypergraphs of degree three	18
3.2.3	APX-hardness results and constant factor approximations	20
4	Future works	20

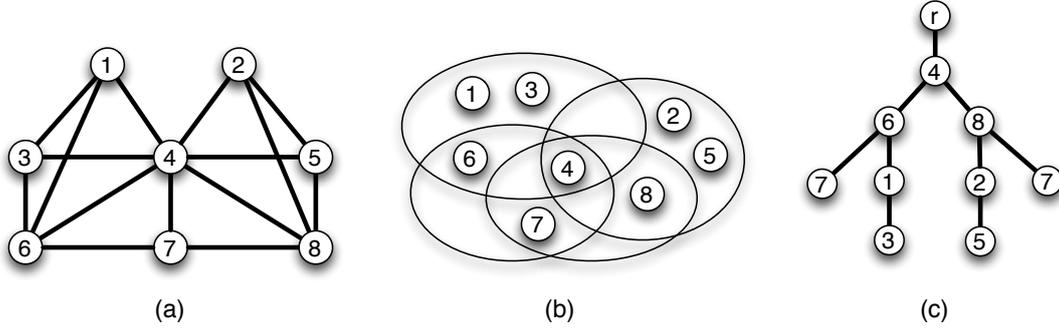


Figure 1: **(a)** Simplicial complex \mathcal{K} composed of eight vertices $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$. A vertex with $i \in \llbracket 1, 8 \rrbracket$ represents v_i . The set of maximal simplices is composed of two tetrahedrons induced by $\{v_1, v_3, v_4, v_6\}$ and $\{v_2, v_4, v_5, v_8\}$, and two triangles induced by $\{v_4, v_6, v_7\}$ and $\{v_4, v_7, v_8\}$. **(b)** Hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$ and $\mathcal{E} = \{\{v_1, v_3, v_4, v_6\}, \{v_2, v_4, v_5, v_8\}, \{v_4, v_6, v_7\}, \{v_4, v_7, v_8\}\}$, that models \mathcal{K} . A node with $i \in \llbracket 1, 8 \rrbracket$ represents v_i . **(c)** Tree representation T of \mathcal{H} . A node u with $i \in \llbracket 1, 8 \rrbracket$ is such that $L_1(u) = v_i$.

1 Introduction

In this article, we consider the problem of representing simplicial complexes by compact structures as trees. In addition to decreasing the size of the representation, searching a simplex and updating such a structure must be done efficiently. In this section, we first explain the importance of representing simplicial complexes by trees. We then introduce the notion of tree representation, and explain why we need to introduce some new constraints. We also describe some related works and open questions about the representation of simplicial complexes. We finally summarize our contributions and give the organization of the paper.

Need for a compact structure. Simplicial complexes are used extensively in combinatorial and computational topology. There are many applications that involve simplicial complexes (e.g. topological data analysis and geometric inference). One of the main problems is that the size of the complexes is very large and increases significantly with the dimension of the structures. Consequently, the use of simplicial complexes is limited in practice. An important problem is to store simplicial complexes by using compact structures. One of the most natural and efficient ways of compacting the size of a simplicial complex is to represent it as a rooted node-labeled tree. Intuitively, every maximal simplex is represented by a path between the root and a leaf.

Modeling maximal simplices by hypergraphs. In this paper, we consider the problem of representing all the maximal simplices of a given simplicial complex \mathcal{K} by a rooted node-labeled tree. To do that, \mathcal{K} is modeled by a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where the set of vertices \mathcal{V} is the set of vertices of \mathcal{K} and the set of hyperedges \mathcal{E} is the set of maximal simplices of \mathcal{K} . Note that $e \not\subseteq e'$ for all $e, e' \in \mathcal{E}$, $e \neq e'$. We deduce that all the results presented in this paper are also valid for representing a hypergraph in which there is no hyperedge that is contained into another hyperedge. We restrict our attention to those hypergraphs we simply call hypergraphs in the sequel.

Notations. Let us define some notations used in the paper. Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph. Let $v \in \mathcal{V}$ be any node of \mathcal{H} . The set $N_{\mathcal{H}}(v)$ represents the set of neighbors of v in \mathcal{H} , that is $N_{\mathcal{H}}(v) = \{v' \in \mathcal{V} \mid v' \neq v, v' \in e, v \in e, e \in \mathcal{E}\}$. We define $N_{\mathcal{H}}[v] = N_{\mathcal{H}}(v) \cup \{v\}$ as the

closed neighborhood of v . We denote by $|e|$ the size of hyperedge $e \in \mathcal{E}$, that is the number of distinct vertices that are contained in e . Let $d_{\mathcal{H}} = \max_{e \in \mathcal{E}} |e|$ be the dimension of \mathcal{H} . Let $\mathcal{E}_v = \{e \setminus \{v\} \mid v \in e, e \in \mathcal{E}\}$ be the set of all hyperedges of \mathcal{H} that contain node v , for which we have removed node v . Let $\bar{\mathcal{E}}_v = \{e \mid v \notin e, e \in \mathcal{E}\}$ be the set of all hyperedges of \mathcal{H} that do not contain node v . Let $\mathcal{E}[v] = \{e \mid v \in e, e \in \mathcal{E}\}$ be the set of all hyperedges of \mathcal{H} that contain node v . Let $\bar{\mathcal{E}}[v] = \bar{\mathcal{E}}_v$. Let $\Delta_{\mathcal{H}} = \max_{v \in \mathcal{V}} |\mathcal{E}[v]|$ be the maximum degree of \mathcal{H} . Let $n = |\mathcal{V}|$. Let $\Sigma(\mathcal{V})$ be the set of all the orderings of \mathcal{V} . Let $\sigma = (\sigma_1, \dots, \sigma_n) \in \Sigma(\mathcal{V})$ be any ordering of \mathcal{V} . For every $e \in \mathcal{E}$, we define $\sigma^e = (\sigma_1^e, \dots, \sigma_{|e|}^e)$ as the ordering induced by the subset of nodes of $e \in \mathcal{E}$ from σ .

Let $T = (V, E)$ be a tree rooted at $r \in V$ and let $u \in V$ be any node of T . The tree $T[u]$ is the subtree of T rooted at u , that is the tree induced by the set of nodes $\{u' \in V \mid u \in V(P_{u',r})\}$, where $P_{u',r}$ is the simple path in T between u' and r . For every $u \in V \setminus \{r\}$, let $p_T(u)$ be the parent of u in T , that is $\{p_T(u), u\} \in E$ and $p_T(u) \notin V(T[u])$.

Tree representation. As mentioned before, the problems studied in this paper concern the representation of maximal simplices (hyperedges) by rooted node-labeled trees. The idea is to factorize the representation of the vertices that appear in several maximal simplices (hyperedges), in order to minimize the space used to store a simplicial complex (hypergraph). Let \mathcal{K} be any simplicial complex and let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be the hypergraph modeling \mathcal{K} . The problem considered in our work is to construct a rooted node-labeled tree that represents \mathcal{H} . This notion, called *tree representation*, is formalized in Definition 1.

Definition 1 (tree representation) *Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph. A tree representation of \mathcal{H} is defined as a node-labeled tree $T = (V, E, L_1)$ rooted at $r \in V$, $L_1 : V \rightarrow \mathcal{V}$, such that:*

1. *for every hyperedge $e \in \mathcal{E}$, there exists a simple path P_e in T between the root r and a leaf such that for all $v \in e$, there exists a node $u \in V(P_e) \setminus \{r\}$ such that $L_1(u) = v$;*
2. *the number of leaves of T is $|\mathcal{E}|$.*

Note that $L_1(r)$ is arbitrarily chosen. Property (1) means that for every hyperedge, there exists a simple path between the root and a leaf that represents this hyperedge. Property (2) states that every simple path between the root and a leaf represents a hyperedge. Thus, there is a bijection between the set of hyperedges and the set of simple paths of the tree.

Example. Figure 1(a) represents a simplicial complex \mathcal{K} , Figure 1(b) is the hypergraph \mathcal{H} modeling \mathcal{K} , and Figure 1(c) depicts a tree representation T of \mathcal{H} . Observe that vertex v_4 (v_6, v_8 , respectively) appears in four (two, respectively) hyperedges but there is a unique node labeled by v_4 (v_6, v_8 , respectively) in T .

Need for additional constraints and problems. We prove that the notion of tree representation (Definition 1) is not sufficient for designing efficient algorithms for searching, removing, and adding a given maximal simplex in T , and so to update the structure efficiently. Thus, some other constraints are needed for tree representations. Informally, given a hypergraph, the associated combinatorial problems consist in computing a rooted node-labeled tree T such that: (a) T is a tree representation of the hypergraph; (b) T satisfies some additional constraints in order to admit efficient algorithms (e.g. for searching a simplex); (c) T has a minimum number of nodes with respect to the two first properties.

Related works. The Hasse diagram of a simplicial complex \mathcal{K} is the graph that associate a node to each simplex $\tau_1 \in \mathcal{K}$ and an edge between two nodes if the associated simplices τ_1 and τ_2 satisfy $\tau_1 \subset \tau_2$ and $\dim(\tau_1) = \dim(\tau_2) - 1$. The Hasse diagram does not permit to efficiently (in terms of size) represent a simplicial complex. The notion of simplex tree has been introduced in [2] for representing simplicial complexes in a compact way. Recently, in [1], the problem of

compressing the simplex tree has been investigated. The constraint is that the compact simplex tree must preserve the functionalities of the original structure (e.g. admitting efficient algorithms for searching a simplex). Our article focus on the global tree representation (equivalent to one of the structures introduced in [1]) and on the local tree representation. These two representations satisfy the additional constraints seen before.

Our contributions. In Section 2, we prove that there is no efficient algorithm for the problem of searching a given maximal simplex in a tree representation. We thus introduce local and global tree representations that permit to design efficient algorithms for the problems of searching, removing, and adding a given maximal simplex. Both these two tree representations are efficient depending on the objective parameter: the local tree representation is to be chosen for the size of the structure and the global tree representation is to be chosen for the time complexity for searching a given simplex.

In Section 3, we analyze the complexity of the combinatorial optimization problems, namely the tree representation problems, introduced in this paper. We prove that these problems are equivalent when the maximal simplices are all of size two (class of graphs), are NP-complete even for the class of cubic graphs and for the class of planar graphs of maximum degree at most three, and admit a linear time 2-approximation algorithm for this class of instances. We also show that the tree representation problems are in P when all the vertices are in at most two maximal simplices (hypergraphs with maximum degree at most two), are NP-complete even when all the vertices are in at most three maximal simplices (hypergraphs of maximum degree three), are APX-complete for bounded degree complexes, are not in APX for unbounded maximum degree complexes, and admit polynomial time constant factor approximation algorithms in the class of bounded degree complexes.

2 Local and global tree representations of hypergraphs

We first motivate the introduction of the local and global tree representations. We then analyze the complexity of the problem of searching, removing, and adding a given simplex. We finally define the associated combinatorial optimization problems and analyze quantitatively the difference between the different tree representations.

2.1 Need for additional constraints

As mentioned before, the problems of searching, removing, and adding a given maximal simplex must admit efficient algorithms. We formally prove in Lemma 1 that there is no efficient algorithm for the problem of searching a given maximal simplex in a tree representation (that it when only the properties of Definition 1 are satisfied). To illustrate that point, consider the tree representation $T = (V, E, L_1)$ rooted at $r \in V$ depicted in Figure 2(a). We cannot easily verify if the hyperedge $e = \{a_1, a_2, a_3, a_4, b_1\}$ belongs to the hypergraph \mathcal{H} represented by T . Indeed, the root r of T has two neighbors $u, u' \in N_T(r)$ such that $L_1(u) = a_1 \in e$ and $L_1(u') = b_1 \in e$. Thus, we do not know if hyperedge e is represented by a path (r, u, \dots) or by a path (r, u', \dots) .

Lemma 1 (simplex search algorithm for tree representation) *Let \mathbf{A} be any algorithm for the problem of searching a given maximal simplex. Then, there exists a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ and a tree representation T of \mathcal{H} such that the time complexity of \mathbf{A} is $\Omega(|\mathcal{V}|^2) = \Omega(|V(T)|)$.*

Proof Let $n \geq 1$ be any integer. Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph, where $\mathcal{V} = \{v_1, \dots, v_{2n}\}$ and $\mathcal{E} = \{e_1, \dots, e_n\}$ with $e_i = \{v_i, v_{n+1}, \dots, v_{2n}\}$ for every $i \in \llbracket 1, n \rrbracket$. We construct a tree representation $T = (V, E, L_1)$ rooted at $r \in V$ of \mathcal{H} as follows. Let $V = \{r\} \cup V_1 \cup \dots \cup V_n \cup$

$\{u^1, \dots, u^n\}$ with $V_i = \{u_i^{n+1}, \dots, u_i^{2n}\}$. Let $E = \{\{r, u_i^{n+1}\} \mid 1 \leq i \leq n\} \cup \{\{u_i^j, u_i^{j+1}\} \mid n+1 \leq j \leq 2n-1, 1 \leq i \leq n\} \cup \{\{u_i^{2n}, u^i\} \mid 1 \leq i \leq n\}$. Let $L_1(u_i^j) = v_j$ and $L_1(u^i) = v_i$ for every $j \in \llbracket n+1, 2n \rrbracket$ and for every $i \in \llbracket 1, n \rrbracket$.

Let \mathbf{A} be any algorithm for the problem of deciding if a given simplex belongs to a tree representation. Assume, without loss of generality, that the maximal simplex is $e_n \in \mathcal{E}$. Since, for every $i \in \llbracket 1, n \rrbracket$, the path $P_i = (r, u_i^{n+1}, \dots, u_i^{2n})$ is such that $L_1(u) \in e_n$ for every $u \in V(P_i) \setminus \{r\}$, then, in the worst case, Algorithm \mathbf{A} visits all nodes of every path P_i for every $i \in \llbracket 1, n \rrbracket$. Thus, the time complexity of Algorithm \mathbf{A} is $\Omega(|\mathcal{V}|^2) = \Omega(|V(T)|)$. \blacksquare

Consequently, we propose in the next section some additional requirements and we get two more constrained definitions of tree representation.

2.2 Definitions and equivalences

We introduce in Definition 2 the notion of local tree representation, a recursively constructed tree that represents a given hypergraph.

Definition 2 (local tree representation) *Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph. A local tree representation of \mathcal{H} is a node-labeled tree $T = (V, E, L_1, L_2)$ rooted at $r \in V$, $L_1 : V \rightarrow \mathcal{V}$, $L_2 : V \rightarrow \llbracket 0, |\mathcal{V}| \rrbracket$, such that:*

1. *if $|\mathcal{E}| = 0$, then $T = (\{r\}, \emptyset)$;*
2. *if $|\mathcal{E}| \geq 1$, then there exists a node $u \in N_T(r)$, with $L_1(u) = v \in \mathcal{V}$, such that:*
 - (a) *for every $u' \in N_T(r) \setminus \{u\}$, then $L_2(u) < L_2(u')$;*
 - (b) *the tree $T[u]$ rooted at $r' = u$ is a local tree representation of $(\mathcal{V} \setminus \{v\}, \mathcal{E}_v)$;*
 - (c) *the tree $T \setminus T[u]$ rooted at r is a local tree representation of $(\mathcal{V} \setminus \{v\}, \bar{\mathcal{E}}_v)$.*

Property (1) deals with the case where there are no hyperedges. Property (2.b) states that all the hyperedges containing v are represented in $T[u]$ and Property (2.c) states that all the other hyperedges are represented in $T \setminus T[u]$. Furthermore, as proved in Lemma 4, Property (2.a) allows to search efficiently the path of T that represents a given hyperedge (if it exists). We prove in Lemma 2 an equivalent definition of a local tree representation (see Section ?? for the proof).

Lemma 2 (equivalence for the local tree representation) *Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph. A tree $T = (V, E, L_1, L_2)$ rooted at $r \in V$, $L_1 : V \rightarrow \mathcal{V}$, $L_2 : V \rightarrow \llbracket 0, |\mathcal{V}| \rrbracket$, is a local tree representation of \mathcal{H} if and only if*

1. *T is a tree representation of \mathcal{H} (Definition 1),*
2. *every simple path P_e representing $e \in \mathcal{E}$ is such that for all $u \in V(P_e) \setminus \{r\}$ and for all $u' \in N_{T[p_T(u)]}(p_T(u))$ with $L_1(u') \in e$, then $L_2(u) < L_2(u')$.*

Property (1) states that a local tree representation is a tree representation. Property (2) allows to determine efficiently the path in the tree that corresponds to any given hyperedge. Figure 2(b) depicts a local tree representation for some hypergraph.

Proof \Rightarrow Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be any hypergraph. Let $T = (V, E, L_1, L_2)$ rooted at $r \in V$ be any local tree representation of \mathcal{H} . We prove that T satisfies the properties of Lemma 2. We prove the result by induction on the number of hyperedges. The result is clearly true for every hypergraph

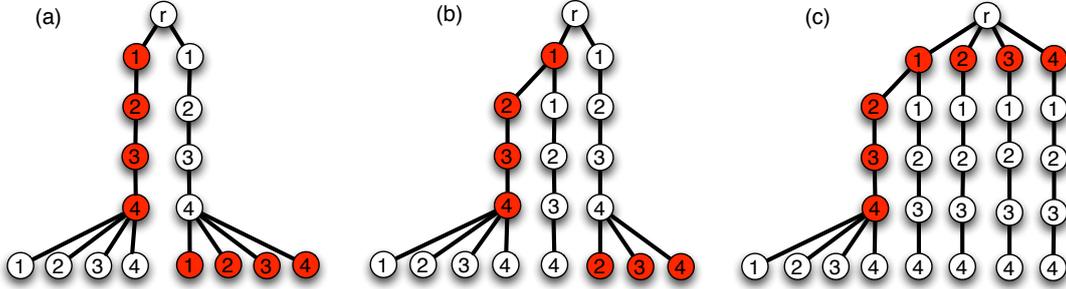


Figure 2: Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4\}$ and $\mathcal{E} = \{\{a_1, a_2, a_3, a_4, b_1\}, \{a_1, a_2, a_3, a_4, b_2\}, \{a_1, a_2, a_3, a_4, b_3\}, \{a_1, a_2, a_3, a_4, b_4\}, \{b_1, b_2, b_3, b_4, a_1\}, \{b_1, b_2, b_3, b_4, a_2\}, \{b_1, b_2, b_3, b_4, a_3\}, \{b_1, b_2, b_3, b_4, a_4\}\}$. A filled red node u with $i \in \llbracket 1, 4 \rrbracket$ is such that $L_1(u) = a_i$ and $L_2(u) = i$. A white node u with $i \in \llbracket 1, 4 \rrbracket$ is such that $L_1(u) = b_i$ and $L_2(u) = i + 4$. **(a)** Tree representation of \mathcal{H} . **(b)** Local tree representation of \mathcal{H} . Let u_1 be such that $L_1(u_1) = a_1$. The tree $T[u_1]$ rooted at u_1 is a local tree representation of $(\mathcal{V} \setminus \{a_1\}, \mathcal{E}_{a_1})$. Note that $\mathcal{E}_{a_1} = \{\{a_2, a_3, a_4, b_1\}, \{a_2, a_3, a_4, b_2\}, \{a_2, a_3, a_4, b_3\}, \{a_2, a_3, a_4, b_4\}, \{b_1, b_2, b_3, b_4\}\}$. The tree $T \setminus T[u_1]$ rooted at r is a local tree representation of $(\mathcal{V} \setminus \{a_1\}, \bar{\mathcal{E}}_{a_1})$. Note that $\bar{\mathcal{E}}_{a_1} = \{\{b_1, b_2, b_3, b_4, a_2\}, \{b_1, b_2, b_3, b_4, a_3\}, \{b_1, b_2, b_3, b_4, a_4\}\}$. All the hyperedges containing a_1 are represented in $T[u_1]$ and all the other hyperedges are represented in $T \setminus T[u_1]$. **(c)** Global tree representation of \mathcal{H} . A corresponding ordering is $\sigma = (a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4)$.

$\mathcal{H} = (\mathcal{V}, \mathcal{E})$ such that $|\mathcal{E}| \leq 1$. Assume now that the result is true for every hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ such that $|\mathcal{E}| \leq m$, for any $m \geq 2$. We prove that it is also true for every hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ such that $|\mathcal{E}| \leq m + 1$. Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be any hypergraph such that $|\mathcal{E}| \leq m + 1$ and let $T = (V, E, L_1, L_2)$ rooted at $r \in V$ be any local tree representation of \mathcal{H} . Let $u \in N_T(r)$ such that $L_2(u) < L_2(u')$ for every $u' \in N_T(r) \setminus \{u\}$. Let $v = L_1(u) \in \mathcal{V}$. By assumption, the tree $T[u]$ rooted at $r' = u$ is a local tree representation of $(\mathcal{V} \setminus \{v\}, \mathcal{E}_v)$ and the tree $T \setminus T[u]$ rooted at r is a local tree representation of $(\mathcal{V} \setminus \{v\}, \bar{\mathcal{E}}_v)$. There are two cases.

- If $|\mathcal{E}_v| \leq m$, then, by induction hypothesis, $T[u]$ and $T \setminus T[u]$ satisfy the properties of Lemma 2. Thus, T satisfies the properties of Lemma 2 and the result is true for every hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ such that $|\mathcal{E}| \leq m + 1$.
- If $|\mathcal{E}_v| = m + 1$, then we prove the result for $T[u]$ instead of T , and so for $(\mathcal{V} \setminus \{v\}, \mathcal{E}_v)$ instead of $(\mathcal{V}, \mathcal{E})$. Indeed, the number of nodes of the hyperedge strictly decreases, that is $|\mathcal{V} \setminus \{v\}| = |\mathcal{V}| - 1$. Thus, this procedure ends with a hypergraph $\mathcal{H}' = (\mathcal{V}', \mathcal{E}')$ such that $|\mathcal{E}'| \leq m$ because if there are no nodes in a hypergraph, then there are no hyperedges in it. Finally, T satisfies the properties of Lemma 2 and the result is true for every hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ such that $|\mathcal{E}| \leq m + 1$.

\Leftarrow Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be any hypergraph. Let $T = (V, E, L_1, L_2)$ be any tree rooted at $r \in V$ that satisfies the properties of Lemma 2. We prove that T satisfies the properties of Definition 2. We prove the result by induction on the number of hyperedges. The result is clearly true for every hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ such that $|\mathcal{E}| \leq 1$. Assume now that the result is true for every hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ such that $|\mathcal{E}| \leq m$, for any $m \geq 2$. We prove that it is also true for every hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ such that $|\mathcal{E}| \leq m + 1$. Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be any hypergraph such that $|\mathcal{E}| \leq m + 1$ and let $T = (V, E, L_1, L_2)$ be any tree rooted at $r \in V$ that satisfies the properties

of Lemma 2 for \mathcal{H} . Let $u \in N_T(r)$ such that $L_2(u) < L_2(u')$ for every $u' \in N_T(r) \setminus \{u\}$. Let $v = L_1(u) \in \mathcal{V}$. From Property (2) of Lemma 2, every hyperedge $e \in \mathcal{E}[v]$, that is every hyperedge that contains v , is represented by a path (r, u, \dots) in T . Thus, every hyperedge $e' \in \bar{\mathcal{E}}[v] = \bar{\mathcal{E}}_v$, that is every hyperedge that does not contain v , is represented by a path (r, u', \dots) in T , for some $u' \in N_T(r) \setminus \{u\}$. Consider the tree $T[u]$ rooted at $r' = u$ and the tree $T \setminus T[u]$ rooted at r . There are two cases.

- If $|\mathcal{E}_v| \leq m$, then, by induction hypothesis, $T[u]$ is a local tree representation of $(\mathcal{V} \setminus \{v\}, \mathcal{E}_v)$, and $T \setminus T[u]$ is a local tree representation of $(\mathcal{V} \setminus \{v\}, \bar{\mathcal{E}}_v)$. Thus, T is a local tree representation of \mathcal{H} , and so the result is true for every hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ such that $|\mathcal{E}| \leq m + 1$.
- If $|\mathcal{E}_v| = m + 1$, then we prove the result for $T[u]$ instead of T , and so for $(\mathcal{V} \setminus \{v\}, \mathcal{E}_v)$ instead of $(\mathcal{V}, \mathcal{E})$. Indeed, the number of nodes of the hyperedge strictly decreases, that is $|\mathcal{V} \setminus \{v\}| = |\mathcal{V}| - 1$. Thus, this procedure ends with a hypergraph $\mathcal{H}' = (\mathcal{V}', \mathcal{E}')$ such that $|\mathcal{E}'| \leq m$ because if there are no nodes in a hypergraph, then there are no hyperedges in it. Finally, T is a local tree representation of \mathcal{H} , and so the result is true for every hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ such that $|\mathcal{E}| \leq m + 1$.

In conclusion, the properties of Definition 2 and the properties of Lemma 2 are equivalent. \blacksquare

We now formalize in Definition 3 the notion of global tree representation from an ordering of the set of nodes of a hypergraph.

Definition 3 (global tree representation) *Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph. Let σ be any ordering of \mathcal{V} . A global tree representation of \mathcal{H} is a node-labeled tree $T_\sigma = (V, E, L_1, L_2)$ rooted at $r \in V$, $L_1 : V \rightarrow \mathcal{V}$, $L_2 : V \rightarrow \llbracket 0, |\mathcal{V}| \rrbracket$, constructed as follows. Set $T_\sigma = (\{r\}, \emptyset)$. For all $e \in \mathcal{E}$:*

1. let $P = (u_0 = r, \dots, u_t)$ be the maximal path in T_σ such that $L_1(u_i) = \sigma_i^e$ for all $i \in \llbracket 1, t \rrbracket$;
2. add the path $(u_{t+1}^e, \dots, u_{|e|}^e)$ and the edge $\{u_t, u_{t+1}^e\}$ in T_σ ;
3. for all $i \in \llbracket t + 1, |e| \rrbracket$, set $L_1(u_i^e) = \sigma_i^e$ and $L_2(u_i^e) = k$, where $\sigma_k = L_1(u_t^e)$.

Recall that, for every $e \in \mathcal{E}$, $\sigma^e = (\sigma_1^e, \dots, \sigma_{|e|}^e)$ is the ordering induced by the subset of nodes of $e \in \mathcal{E}$ from σ . We prove in Lemma 3 an equivalent definition of the global tree representation.

Lemma 3 (equivalence for the global tree representation) *Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph. A tree $T = (V, E, L_1, L_2)$ rooted at $r \in V$, $L_1 : V \rightarrow \mathcal{V}$, $L_2 : V \rightarrow \llbracket 0, |\mathcal{V}| \rrbracket$, is a global tree representation of \mathcal{H} if and only if*

1. T is a local tree representation of \mathcal{H} ;
2. for all $u, u' \in V$, then $L_1(u) = L_1(u')$ if and only if $L_2(u) = L_2(u')$;
3. for every simple path $P = (r, u_1, \dots, u_t)$ of T , then $L_2(u_i) < L_2(u_{i+1})$ for all $i \in \llbracket 1, t - 1 \rrbracket$.

Property (1) states that a global tree representation is a local tree representation. Property (2) ensures that if two different nodes of the tree represent a same node of the hypergraph, then these two nodes have the same label for L_2 . Property (3) means that every path in the tree is strictly increasing for the labeling function L_2 . Figure 2(c) depicts a global tree representation for some hypergraph.

Proof Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be any hypergraph. Let $n = |\mathcal{V}|$.

\Leftarrow Let $T = (V, E, L_1, L_2)$ rooted at $r \in V$ be any tree that satisfies the properties of Lemma 3. We prove that T is a global tree representation of \mathcal{H} . In other words, we prove that there exists an ordering $\sigma = (\sigma_1, \dots, \sigma_n)$ of \mathcal{V} such that $T_\sigma = T$ is constructed from σ (Definition 3). From Property (2) of Lemma 3, for all $u, u' \in V$ such that $L_1(u) \neq L_1(u')$, then $L_2(u) \neq L_2(u')$. Let $\sigma = (\sigma_1, \dots, \sigma_n)$ be an ordering of \mathcal{V} such that for all $u, u' \in V$ such that $L_1(u) = \sigma_i$ and $L_1(u') = \sigma_j$ with $1 \leq i < j \leq n$, then $L_2(u) < L_2(u')$. It follows that, from the ordering σ , we get $T_\sigma = T$.

\Rightarrow Let $T = (V, E, L_1, L_2)$ be any global tree representation of \mathcal{H} rooted at $r \in V$. Let $\sigma = (\sigma_1, \dots, \sigma_n)$ be an ordering of \mathcal{V} such that $T_\sigma = T$ is constructed from σ (Definition 3). We prove that T_σ satisfies the properties of Lemma 3. We prove the result by induction on the number m' of hyperedges added in T_σ . The result is true for $m' = 0$ because $T_\sigma = (\{r\}, \emptyset)$ satisfies the properties of Lemma 3 for the hypergraph (\emptyset, \emptyset) . The result is true for $m' = 1$ because T_σ is the path $(r, u_1^1, \dots, u_{|e_1|}^1)$ with $L_1(u_i^1) = \sigma_i^1$ and $L_2(u_i^1) = k$, where $\sigma_k = L(u_i^j)$ for all $i \in \llbracket 1, |e_1| \rrbracket$. Indeed, the path represents the hyperedge e_1 because every node of e_1 is represented and the path is increasing for L_2 .

Suppose it is true for m' added hyperedges, $1 \leq m' < m$. We prove it is also true for $m' + 1$. The tree T_σ satisfies the properties of Lemma 3 for the hypergraph induced by the set of hyperedges $\{e_1, \dots, e_{m'}\}$. We now represent the hyperedge $e_{m'+1}$. We add in T_σ the path $(u_{t+1}^{m'+1}, \dots, u_{|e|}^{m'+1})$ and the edge $\{u_t, u_{t+1}^{m'+1}\}$. Recall that $t \geq 0$ is the largest integer such that there exists a simple path $P = (r = u_0, u_1, \dots, u_t)$ in T_σ with $L_1(u_i) = \sigma_i^{m'+1}$ for all $i \in \llbracket 1, t \rrbracket$. We set $L_1(u_i^{m'+1}) = \sigma_i^{m'+1}$ and we set $L_2(u_i^{m'+1}) = k$, where $\sigma_k = L(u_i^{m'+1})$, for all $i \in \llbracket t+1, |e_{m'+1}| \rrbracket$. Thus, T_σ satisfies the properties of Lemma 3 for the hypergraph induced by the set of hyperedges $\{e_1, \dots, e_{m'+1}\}$. \blacksquare

2.3 Efficient simplex search, remove, and add algorithms

We prove in Lemma 4 (Lemma 5, respectively) that a local (global, respectively) tree representation admits an efficient algorithm for searching, removing, and adding a simplex.

Lemma 4 (simplex search/remove/add algorithm for local tree representation) *Let \mathcal{H} be any hypergraph and let T be a local tree representation of \mathcal{H} . There exists a $O(d_{\mathcal{H}}^2 \log_2(\Delta_T))$ -time complexity algorithm for the problem of searching/removing/adding a given maximal simplex.*

Proof Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be any hypergraph. Let $T = (V, E, L_1, L_2)$ rooted at $r \in V$ be a local tree representation of \mathcal{H} . We first prove the result for the problem of searching a simplex. Without loss of generality, assume that the given maximal simplex belongs to \mathcal{H} . Let $e \in \mathcal{E}$ be any hyperedge of \mathcal{H} . We prove that the problem of computing the node $u \in N_T(r)$ such that the path that represents e in T is (r, u, \dots) can be solved in $O(d_{\mathcal{H}} \log_2(\Delta_T))$ -time. For every node $v \in e$, we compute the node $u \in N_T(r)$ (if it exists) such that $L_1(u) = v$. This can be done in $O(\log_2(\Delta_T))$ -time. Note that we suppose that the set of nodes $N_T(r)$ is ordered. Since $|e| \leq d_{\mathcal{H}}$, then we get a $O(d_{\mathcal{H}} \log_2(\Delta_T))$ -time complexity. This computation is done for every node of the path of T that represents e . Thus, we get a $O(d_{\mathcal{H}}^2 \log_2(\Delta_T))$ -time complexity algorithm for the problem of searching the path (if it exists) representing a given maximal simplex. Finally, the problem of removing a simplex consists in first searching the path corresponding to this simplex and then removing a subpath of this path, and the problem of adding a simplex consists in searching a path representing a subset of nodes of this simplex. \blacksquare

Lemma 5 (simplex search/remove/add algorithm for global tree representation) *Let \mathcal{H} be any hypergraph and let T be a global tree representation of \mathcal{H} . There exists a $O(d_{\mathcal{H}} \log_2(\Delta_T))$ -time complexity algorithm for the problem of searching/removing/adding a given maximal simplex.*

Proof Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be any hypergraph. Let $T = (V, E, L_1, L_2)$ rooted at $r \in V$ be a global tree representation of \mathcal{H} . Let $\sigma = (\sigma_1, \dots, \sigma_n)$ be an ordering of \mathcal{V} corresponding to T . Without loss of generality, assume that the given maximal simplex belongs to the hypergraph. Thus, let $e \in \mathcal{E}$ be any hyperedge of \mathcal{H} . We prove that the problem of computing the node $u \in N_T(r)$ such that the path that represents e in T is (r, u, \dots) can be solved in $O(\log_2(\Delta_T))$ -time. Indeed, it is sufficient to search the node $v \in e$ such that $\sigma_i = v$ and for every $v' \in e \setminus \{v\}$, then $\sigma_j = v'$ is such that $j > i$. That computation can be done in $O(\log_2(\Delta_T))$ -time. Note that we suppose that the set of nodes $N_T(r)$ is ordered. We repeat this computation for every node of the path of T that represents e , and so we get a $O(d_{\mathcal{H}} \log_2(\Delta_T))$ -time algorithm. Finally, the problem of removing a simplex consists in first searching the path corresponding to this simplex and then removing a subpath of this path, and the problem of adding a simplex consists in searching a path representing a subset of nodes of this simplex. ■

Note that $\log_2(\Delta_T) = O(\log_2(|\mathcal{V}|))$. The time complexity for searching, removing, and adding a given simplex is better for the global tree representation. However, we prove in Section 2.5 that the size of an optimal global tree representation is always greater than the size of an optimal local tree representation. We present in the next section the different combinatorial optimization problems studied in this article.

2.4 Combinatorial optimization problems

In this paper, we aim at computing tree representations of a given hypergraph with the smallest number of nodes. Intuitively, we aim at determining the maximum number of nodes that can be factorized in the different tree representations. The problems investigated in this article and some equivalences are formally described below.

Problem 1 (tree representation problem) *Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be any hypergraph. The tree representation problem consists in computing the maximum max^* such that there exists a tree representation T^* of \mathcal{H} with $|V(T^*)| = -max^* + 1 + \sum_{e \in \mathcal{E}} |e|$.*

The term $\sum_{e \in \mathcal{E}} |e|$ represents the size of \mathcal{E} , and so the size of \mathcal{H} , without optimization. The term 1 is added for the root of the tree.

Problem 2 (local tree representation problem) *Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be any hypergraph. The local tree representation problem consists in computing the maximum max_{local}^* such that there exists a local tree representation T_{local}^* of \mathcal{H} with $|V(T_{local}^*)| = -max_{local}^* + 1 + \sum_{e \in \mathcal{E}} |e|$.*

From Lemma 2, we deduce Corollary 6.

Corollary 6 (equivalence for the local tree representation problem) *Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be any hypergraph. Then, $max_{local}^* = -f(\mathcal{V}, \mathcal{E}) + \sum_{e \in \mathcal{E}} |e|$, that is $|V(T_{local}^*)| = f(\mathcal{V}, \mathcal{E}) + 1$, where the function f is defined as follows: $f(\mathcal{V}, \mathcal{E}) = 0$ if $|\mathcal{E}| = 0$ and $f(\mathcal{V}, \mathcal{E}) = \min_{v \in \mathcal{V}} (f(\mathcal{V} \setminus \{v\}, \mathcal{E}_v) + f(\mathcal{V} \setminus \{v\}, \bar{\mathcal{E}}_v)) + 1$ if $|\mathcal{E}| \geq 1$.*

Problem 3 (global tree representation problem) *Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be any hypergraph. The global tree representation problem consists in computing the maximum max_{global}^* such that there exists a global tree representation T_{global}^* of \mathcal{H} with $|V(T_{global}^*)| = -max_{global}^* + 1 + \sum_{e \in \mathcal{E}} |e|$.*

From Lemma 3, we deduce Corollary 7.

Corollary 7 (equivalence for the global tree representation problem) *Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be any hypergraph. Then, $\max_{global}^* = -\min_{\sigma \in \Sigma(\mathcal{V})} |V(T_\sigma)| + 1 + \sum_{e \in \mathcal{E}} |e|$, that is $|V(T_{global}^*)| = \min_{\sigma \in \Sigma(\mathcal{V})} |V(T_\sigma)|$.*

As an illustration, we have $\max^* = 24$ and $|V(T^*)| = 17$ (Figure 2(a)), $\max_{local}^* = 21$ and $|V(T_{local}^*)| = f(\mathcal{V}, \mathcal{E}) + 1 = 20$ (Figure 2(b)), and $\max_{global}^* = 13$ and $|V(T_{global}^*)| = 28$ (Figure 2(c)). Intuitively, \max^* (\max_{local}^* , \max_{global}^* , respectively) represents the maximum number of nodes that can be factorized for the (local, global, respectively) tree representation problem.

2.5 Comparison between the three tree representations

From Lemma 2 and Lemma 3, we first deduce Property 1.

Property 1 *Let \mathcal{H} be any hypergraph. Then, $|V(T_{global}^*)| \geq |V(T_{local}^*)| \geq |V(T^*)|$ and $\max_{global}^* \leq \max_{local}^* \leq \max^*$.*

We now prove in Lemma 8 that there exists an infinite class of hypergraphs \mathcal{C} such that for every hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E}) \in \mathcal{C}$, then $|V(T_{local}^*)| = O(|\mathcal{V}|)$ and $|V(T_{global}^*)| = \Omega(|\mathcal{V}|^2)$.

Lemma 8 *For any $n \geq 1$, there exists a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, with $n = |\mathcal{V}|$, such that $|V(T_{local}^*)| \leq 5n$ and $|V(T_{global}^*)| \geq n^2$, and $\max_{local}^* \geq 2n^2 - 3n + 1$ and $\max_{global}^* \leq (n + 1)^2$.*

Proof Let $n \geq 1$ be any integer. Let $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_n\}$. We now define the hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$. Let $\mathcal{V} = A \cup B$ be the set of nodes. Let $\mathcal{E} = \{\{a_1, \dots, a_n, b\} \mid b \in B\} \cup \{\{b_1, \dots, b_n, a\} \mid a \in A\}$ be the set of $2n$ hyperedges. We prove that $|V(T_{local}^*)| \leq 5n$ and $|V(T_{global}^*)| \geq n^2$.

Claim 1 $|V(T_{local}^*)| \leq 5n$.

Proof (Claim 1). Let us construct a local tree representation T_{local} rooted at $r \in V(T_{local})$ such that $|V(T_{local})| = 5n$. Let $V(T_{local}) = \{r\} \cup \{u_1, \dots, u_n\} \cup \{v_1, \dots, v_n\} \cup \{u'_1, \dots, u'_n\} \cup \{v'_1, \dots, v'_n\} \cup \{w_1, \dots, w_n\}$. Set $L_1(u_i) = a_i$, $L_2(u_i) = i$, $L_1(v_i) = b_i$, $L_2(v_i) = i + n$, $L_1(v'_i) = b_i$, $L_2(v'_i) = i + n$, $L_1(w_i) = b_i$, and $L_2(w_i) = i + n$ for all $i \in \llbracket 1, n \rrbracket$. Set $L_1(u'_i) = a_i$ and $L_2(u'_i) = i$ for all $i \in \llbracket 2, n - 1 \rrbracket$. Finally, let $E(T_{local}) = \{\{r, u_1\}\} \cup \{\{u_i, u_{i+1}\} \mid 1 \leq i \leq n - 1\} \cup \{\{u_n, v'_i\} \mid 1 \leq i \leq n\} \cup \{u_1, w_1\} \cup \{\{w_i, w_{i+1}\} \mid 1 \leq i \leq n - 1\} \cup \{\{r, v_1\}\} \cup \{\{v_i, v_{i+1}\} \mid 1 \leq i \leq n - 1\} \cup \{\{v_n, u'_i\} \mid 2 \leq i \leq n\}$. Figure 2(b) depicts T_{local} for $n = 4$. We now prove that T_{local} is an admissible solution for Problem 2. First, for every $e \in \mathcal{E}$, there exists a path between the root r and a leaf of T_{local} that corresponds to e . For all $i \in \llbracket 1, n \rrbracket$, we associate the path $(r, u_1, \dots, u_n, v'_i)$ with the hyperedge $\{a_1, \dots, a_n, b_i\}$. For all $i \in \llbracket 2, n \rrbracket$, we associate the path $(r, v_1, \dots, v_n, u'_i)$ with the hyperedge $\{b_1, \dots, b_n, a_i\}$. We finally associate the path $(r, u_1, v_1, \dots, v_n)$ with the hyperedge $\{b_1, \dots, b_n, a_1\}$. By construction of L_2 , T_{local} is an admissible solution for Problem 2. We get that $|V(T_{local}^*)| \leq |V(T_{local})| = 5n$. \blacksquare

Claim 2 $|V(T_{global}^*)| \geq n^2$.

Proof (Claim 2). We show that any global tree representation T_{global} is such that $|V(T_{global})| \geq n^2$. Let us consider any ordering $(\sigma_1, \dots, \sigma_{2n})$ of \mathcal{V} . Without loss of generality, we assume that $\sigma_1 \in A$. Let $\alpha_1, \beta_1, \dots, \alpha_n, \beta_n$ and t_a, t_b be such that:

$$\left\{ \begin{array}{l} \sigma_i \in A \text{ for all } i \in \llbracket 1 + \sum_{k=1}^{j-1} \alpha_k + \beta_k, 1 + \alpha_j + \sum_{k=1}^{j-1} \alpha_k + \beta_k \rrbracket, \text{ for all } j \in \llbracket 1, n \rrbracket; \\ \sigma_i \in B \text{ for all } i \in \llbracket 1 + \alpha_j + \sum_{k=1}^{j-1} \alpha_k + \beta_k, 1 + \sum_{k=1}^j \alpha_k + \beta_k \rrbracket \text{ for all } j \in \llbracket 1, n \rrbracket; \\ \alpha_j \geq 1 \text{ for all } j \in \llbracket 1, t_a \rrbracket; \alpha_j = 0 \text{ for all } j \in \llbracket t_a + 1, n \rrbracket; \\ \beta_j \geq 1 \text{ for all } j \in \llbracket 1, t_b \rrbracket; \beta_j = 0 \text{ for all } j \in \llbracket t_b + 1, n \rrbracket; \\ \sum_{k=0}^n \alpha_k = n; \sum_{k=0}^n \beta_k = n. \end{array} \right.$$

For any two subsets of nodes $A' = \{a'_1, \dots, a'_{|A'|}\} \subseteq A$ and $B' = \{b'_1, \dots, b'_{|B'|}\} \subseteq B$, consider the hypergraph $(\mathcal{V}_{A',B'}, \mathcal{E}_{A',B'})$, where $\mathcal{V}_{A',B'} = A' \cup B'$ and $\mathcal{E}_{A',B'} = \{\{a'_1, \dots, a'_{|A'|}, b\} \mid b \in B'\} \cup \{\{b_1, \dots, b_{|B'|}, a\} \mid a \in A'\}$. We prove by induction that a global tree representation $T_{A',B'}$ of $(\mathcal{V}_{A',B'}, \mathcal{E}_{A',B'})$ is such that $|V(T_{A',B'})| \geq |A'| |B'|$ for all $A' \subseteq A$, $B' \subseteq B$, and so we prove that $|V(T_{global})| \geq |A| |B| = n^2$. It is true for $|A'| = |B'| = 1$ because $|V(T_{A',B'})| = 3$. Suppose it is true for all A' and B' such that $|A'| \leq n-1$ and $|B'| \leq n-1$. Recall that we assume that $\sigma_i \in A$ for all $i \in \llbracket 1, \alpha_1 \rrbracket$, and that $\sigma_i \in B$ for all $i \in \llbracket \alpha_1 + 1, \alpha_1 + \beta_1 \rrbracket$. Let us consider $\mathcal{V}_{A_1, B_1} = A_1 \cup B_1$ with $A_1 = \{a_1, \dots, a_{\alpha_1}\}$ and $B_1 = \{b_1, \dots, b_{\beta_1}\}$, and $\mathcal{E}_{A_1, B_1} = \{\{a_1, \dots, a_{\alpha_1}, b\} \mid b \in B_1\} \cup \{\{b_1, \dots, b_{\beta_1}, a\} \mid a \in A_1\}$. The global tree representation T_{A_1, B_1} of $(\mathcal{V}_{A_1, B_1}, \mathcal{E}_{A_1, B_1})$, with the ordering $(a_1, \dots, a_{\alpha_1}, b_1, \dots, b_{\beta_1})$, is such that $|V(T_{A_1, B_1})| = \alpha_1 \beta_1 + 2\alpha_1$. More precisely, $V(T_{A_1, B_1}) = \{r\} \cup \{u_1, \dots, u_{\alpha_1}\} \cup \{u'_2, \dots, u'_{\alpha_1}\} \cup \{v_1, \dots, v_{\beta_1}\} \cup \cup_{i=1}^{\alpha_1} \{v^i_1, \dots, v^i_{\beta_1}\}$. Furthermore, $L_1(u_j) = a_j$ for all $j \in \llbracket 1, \alpha_1 \rrbracket$; $L_1(u'_j) = a_j$ for all $j \in \llbracket 2, \alpha_1 \rrbracket$; $L_1(v_j) = b_j$ for all $j \in \llbracket 1, \beta_1 \rrbracket$; $L_1(v^i_j) = b_j$ for all $j \in \llbracket 1, \beta_1 \rrbracket$ and for all $i \in \llbracket 1, \alpha_1 \rrbracket$. We set $E(T_{A_1, B_1}) = \{r, u_1\} \cup \{\{r, u'_j\} \mid 2 \leq j \leq \alpha_1\} \cup \{\{u_j, u_{j+1}\} \mid 1 \leq j \leq \alpha_1 - 1\} \cup \{\{u_{\alpha_1}, v_j\} \mid 1 \leq j \leq \beta_1\} \cup \{u_1, v^1_1\} \cup \{\{u'_i, v^i_1\} \mid 2 \leq i \leq \alpha_1\} \cup \{\{v^i_j, v^i_{j+1}\} \mid 1 \leq i \leq \alpha_1, 1 \leq j \leq \beta_1 - 1\}$. We get $|V(T_{A_1, B_1})| = \alpha_1 \beta_1 + 2\alpha_1 + \beta_1$. Figure 2(b) depicts T_{A_1, B_1} for $\alpha_1 = 4$ and $\beta_1 = 4$.

From T_{A_1, B_1} , the minimum number of nodes we have to add in order to represent the set of hyperedges $\mathcal{E}'_{A_1, B_1} = \{\{a_1, \dots, a_n, b\} \mid b \in B_1\} \cup \{\{b_1, \dots, b_n, a\} \mid a \in A_1\}$, is $\alpha_1(n - \beta_1) + \beta_1(n - \alpha_1)$. Indeed, we obtain this number of additional nodes for any ordering for the set of nodes $\{a_{\alpha_1+1}, \dots, a_n, b_{\beta_1+1}, \dots, b_n\}$. The resulting tree T'_{A_1, B_1} is then such that $|V(T'_{A_1, B_1})| = n(\alpha_1 + \beta_1) - \alpha_1 \beta_1 + 2\alpha_1 + \beta_1$.

In order to represent the set of hyperedges $\{\{a_1, \dots, a_n, b\} \mid b \in B \setminus B_1\} \cup \{\{b_1, \dots, b_n, a\} \mid a \in A \setminus A_1\}$, we now add new nodes from T'_{A_1, B_1} . By construction, we can only use $2\alpha_1$ nodes of T'_{A_1, B_1} , namely $\{r, u_1, \dots, u_{\alpha_1}, u'_2, \dots, u'_{\alpha_1}\}$, because there is a unique hyperedge that contains the subset of nodes $\{a_1, \dots, a_{\alpha_1}, b_j\}$ for all $j \in \llbracket 1, n \rrbracket$, and there is a unique hyperedge that contains the subset of nodes $\{b_1, b_2, a_j\}$ for all $j \in \llbracket 1, n \rrbracket$.

Thus, the total number of nodes $|V(T_{global})|$ is at least $|V(T'_{A_1, B_1})| - 2\alpha_1$ plus the minimum number of nodes $|V'|$ to represent the set of hyperedges $\mathcal{E}' = \{\{a_{\alpha_1+1}, \dots, a_n, b\} \mid b \in B \setminus B_1\} \cup \{\{b_{\beta_1+1}, \dots, b_n, a\} \mid a \in A \setminus A_1\}$. By induction hypothesis, $|V'| \geq (n - \alpha_1)(n - \beta_1)$. We finally get $|V(T_{global})| \geq |V(T'_{A_1, B_1})| - 2\alpha_1 + |V'| \geq n(\alpha_1 + \beta_1) - \alpha_1 \beta_1 + \beta_1 + (n - \alpha_1)(n - \beta_1)$. Thus, we get $|V(T_{global})| \geq n^2 = |A| |B|$, and so $|V(T_{global}^*)| \geq n^2$. ■

Claim 1 and Claim 2 conclude the proof of Lemma 8. ■

To summarize, both local and global tree representations are efficient depending on the objective parameter (size of the optimal representation or time complexity for searching/removing/adding a given simplex).

3 Computing optimal tree representations is difficult

We prove that the tree representation problems are difficult even in the class of graphs (Section 3.1) and in the class of bounded degree hypergraphs (Section 3.2). However, we also prove polynomial time (approximation and exact) algorithms for some classes of instances.

3.1 Tree representations of graphs

We consider here the class of graphs (hyperedges of size two). We prove in Theorem 10 that the decision variants of the tree representation problem, the local tree representation problem, and the global tree representation problem are NP-complete. On the positive side, we show that there exists a linear time 2-approximation algorithm for these problems. We first prove in Lemma 9 that the three tree representation problems are equivalent in graphs.

Lemma 9 *Let \mathcal{G} be any graph. Then, $|V(T_{global}^*)| = |V(T_{local}^*)| = |V(T^*)|$ and $max_{global}^* = max_{local}^* = max^*$.*

Proof Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be any graph. The number of leaves of T_{global}^* , T_{local}^* , and T^* is the number of edges $|\mathcal{E}|$ of \mathcal{G} . The height of T_{global}^* , T_{local}^* , and T^* is 2 because $|e| = 2$ for every $e \in \mathcal{E}$. Let $T = (V, E, L_1)$ be any tree representation rooted at $r \in V$ of \mathcal{G} . We prove that there exist a global tree representation T_{global} of \mathcal{G} and a local tree representation T_{local} of \mathcal{G} such that $|V(T_{global})| = |V(T_{local})| \leq |V(T)|$.

Let us first construct $T_{global} = (V', E', L'_1, L'_2)$ of \mathcal{G} such that $|V'| \leq |V|$. Let $N_T(r) = \{u_1, \dots, u_k\}$, where $k = |N_T(r)|$ is the degree of r in T . Without loss of generality, assume that $L_1(u_i) \neq L_1(u_j)$ for every $i \in \llbracket 1, k \rrbracket$ and for every $j \in \llbracket 1, i \rrbracket$. Indeed, otherwise, we consider in our construction the tree obtained by merging any two nodes $u \in N_T(r)$ and $u' \in N_T(r)$ such that $L_1(u) = L_1(u')$. The construction of T_{global} rooted at $r' \in V'$ from T is sequential. Initially, set $T_{global} = T$, that is $V' = V$, $E' = E$, and $L'_1 = L_1$. Let $L'_2(u') = 0$ for every $u' \in V'$. Observe that $|V'| = |V|$. Let $N_{T_{global}}(r') = \{u'_1, \dots, u'_k\}$. Then, for $i = 1, \dots, k$, we apply the following procedure. Let $F = \{\{u'_j, u'\} \in E' \mid L_1(u') = L_1(u'_i), i < j \leq |N_{T_{global}}(r')|\}$. For every $e' = \{u'_j, u'\} \in F$, for some $j \in \llbracket i + 1, |N_{T_{global}}(r')|\rrbracket$, we remove e' from E' and we remove u' from V' ; and we add u'' in V' and we add $\{u'_i, u''\}$ in E' with $L_1(u'') = L_1(u'_i)$. Let $L_2(u) = i$ for every $u \in V'$ such that $L_1(u) = L_1(u'_i)$. Finally, consider the set of nodes $X = \{u' \in V' \setminus \{r'\} \mid L_2(u') = 0\}$ for which L_2 has not been defined. Let $L_1(X) = \{L_1(u') \mid u' \in X\} \subset \mathcal{V}$. Without loss of generality, let $L_1(X) = \{v_{k+1}, \dots, v_{|\mathcal{V}|}\}$. Then, for $i = k + 1, \dots, v_{|\mathcal{V}|}$, set $L_2(u) = i$ for every $u \in V'$ such that $L_1(u) = v_i$. By construction, the tree T_{global} is a global tree representation of \mathcal{G} and is such that $|V'| = |V|$.

From Lemma 3, a global tree representation is a local tree representation. Thus, $T_{local} = T_{global}$ is a local tree representation such that $|V(T_{local})| \leq |V(T)|$.

Finally, from Property 1, we have $|V(T_{global}^*)| \geq |V(T_{local}^*)| \geq |V(T^*)|$. Thus, we get that $|V(T_{global}^*)| = |V(T_{local}^*)| = |V(T^*)|$. \blacksquare

In the reduction of the proof of Theorem 10, we use the *vertex cover problem*. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be any graph. A set $X \subseteq \mathcal{V}$ is a *vertex cover* of \mathcal{G} if and only if for all $\{u, v\} \in \mathcal{E}$, then $\{u, v\} \cap X \neq \emptyset$. The vertex cover problem consists in computing the minimum k such that there exists a vertex cover X of \mathcal{G} of size $|X| = k$. The decision variant of the vertex cover problem is NP-complete even for the class of cubic graphs [5] and for the class of planar graphs of degree at most three [3]. The set $X = \{v_1, v_6, v_8\} \subseteq \mathcal{V}$ of filled red nodes is an optimal solution for the vertex cover problem for the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ depicted in Figure 3(a). Figure 3(b) depicts an optimal global tree representation T_{global}^* of \mathcal{G} with $|V(T_{global}^*)| = 1 + |X| + |\mathcal{E}| = 16$, and so $x_{global}^* = 8$.

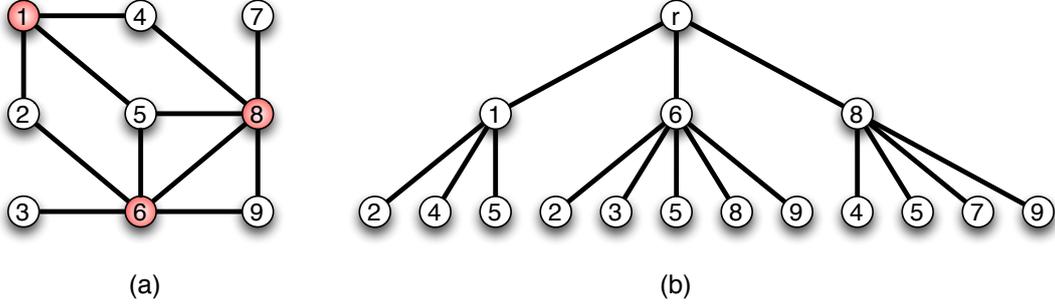


Figure 3: (a) Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$ and $\mathcal{E} = \{\{v_1, v_2\}, \{v_1, v_4\}, \{v_1, v_5\}, \{v_2, v_6\}, \{v_3, v_6\}, \{v_4, v_8\}, \{v_5, v_6\}, \{v_5, v_8\}, \{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_8, v_9\}\}$. A node with $i \in \llbracket 1, 9 \rrbracket$ represents $v_i \in \mathcal{V}$. The set $X = \{v_1, v_6, v_8\}$ of filled red nodes represents a minimum vertex cover of \mathcal{G} . (b) Optimal global tree representation T_{global}^* of \mathcal{G} . Observe that $|V(T_{global}^*)| = 1 + |X| + |\mathcal{E}| = 16$. A node u with $i \in \llbracket 1, 9 \rrbracket$ is such that $L_1(u) = v_i$. Furthermore, $L_2(u) = 1$ if $L_1(u) = v_1$, $L_2(u) = 2$ if $L_1(u) = v_6$, $L_2(u) = 3$ if $L_1(u) = v_8$, $L_2(u) = 4$ if $L_1(u) = v_2$, $L_2(u) = 5$ if $L_1(u) = v_3$, $L_2(u) = 6$ if $L_1(u) = v_4$, $L_2(u) = 7$ if $L_1(u) = v_5$, $L_2(u) = 8$ if $L_1(u) = v_7$, and $L_2(u) = 9$ if $L_1(u) = v_9$.

Theorem 10 *The decision variants of the tree representation problems are NP-complete even for the class of cubic graphs and for the class of planar graphs of degree at most three.*

Proof From Lemma 9, it is sufficient to prove the NP-completeness of the decision variant of the local tree representation problem. First, since the problem of deciding if a tree is a local tree representation can be clearly solved in polynomial time, then the decision variant of the local tree representation problem is in NP.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be any graph and let $k \geq 1$ be any integer. Let $n = |\mathcal{V}|$ and let $\mathcal{V} = \{v_1, \dots, v_n\}$. Let $m = |\mathcal{E}|$ and let $\mathcal{E} = \{e_1, \dots, e_m\}$. Since we consider the class of graphs, then any local tree representation T of \mathcal{G} has m leaves and has height 2.

We prove that there exists a local tree representation T of \mathcal{G} such that $|V(T)| \leq m + k + 1$ if and only if there exists a vertex cover $X \subseteq \mathcal{V}$ of \mathcal{G} of size $|X| \leq k$.

\Leftarrow Suppose there exists a vertex cover $X \subseteq \mathcal{V}$ of \mathcal{G} of size $|X| \leq k$. We prove that there exists a local tree representation T of \mathcal{G} such that $|V(T)| \leq m + k + 1$. Without loss of generality, assume that $X = \{v_1, \dots, v_k\}$. Since X is a minimum vertex cover of \mathcal{G} , then for all $v \in X$, there exists $e = \{v, v'\} \in \mathcal{E}$, $v' \in \mathcal{V}$, such that $v' \notin X$. Indeed, otherwise, we would have a vertex cover $X' = X \setminus \{v\}$ of size $|X'| \leq k - 1$, a contradiction because X is a minimum vertex cover of \mathcal{G} . We now construct a local tree representation $T = (V, E, L_1, L_2)$ rooted at $r \in V$ of \mathcal{G} . Let $N_T(r) = \{u_1, \dots, u_k\}$ be the set of neighbors of r such that $L_1(u_i) = v_i$ and $L_2(u_i) = i$ for all $i \in \llbracket 1, k \rrbracket$. For all $i \in \llbracket 1, k \rrbracket$, let $V^i = \{v_j \in \mathcal{V} \mid \{v_i, v_j\} \in \mathcal{E}, i < j \leq n\}$. For all $i \in \llbracket 1, k \rrbracket$, let $N_T(u_i) \setminus \{r\} = \{u_j^i \mid v_j \in V^i, i < j \leq n\}$ be the set of neighbors of u_i (but r) in T . For all i, j , $1 \leq i \leq k$, $i < j \leq n$, such that $u_j^i \in N_T(u_i) \setminus \{r\}$, we set $L_1(u_j^i) = v_j$ and $L_2(u_j^i) = k + 1$. Thus, the number of nodes is $|V(T)| = m + k + 1$. We finally prove that T satisfies the properties of Definition 2 (local tree representation). We first show that T is a tree representation. Since $X = \{v_1, \dots, v_k\}$ is a vertex cover of \mathcal{G} , then every edge $e = \{v_i, v_j\} \in \mathcal{E}$ is such that $1 \leq i < j \leq n$ and $i \leq k$. Thus, by construction of T , the path $P_e = (r, u_i, u_j^i)$ represents the edge e , that is $L_1(u_i) = v_i$ and $L_1(u_j^i) = v_j$. By construction, the number of leaves of T is m . Thus, T is a tree representation of \mathcal{G} . We finally prove that T satisfies the second property of Lemma 2. Consider

any edge $e = \{v_i, v_j\} \in \mathcal{E}$, $1 \leq i < j \leq n$, $i \leq k$, that is represented by the path $P_e = (r, u_i, u_j^i)$ with $L_1(u_i) = v_i$ and $L_1(u_j^i) = v_j$. Then, there is no node $u \in N_T(r)$ such that $L_1(u) = v_i$, and for every node $u \in N_T(r)$ such that $L_1(u) = v_j$, we necessarily have $L_2(u) > i = L_2(u_i)$. Thus, T is a local tree representation of \mathcal{G} .

\Rightarrow Suppose that there exists a local tree representation T rooted at $r \in V(T)$ of \mathcal{G} such that $|V(T)| \leq m + k + 1$. We prove that there exists a vertex cover $X \subseteq \mathcal{V}$ of \mathcal{G} of size $|X| \leq k$. As previously mentioned, the number of leaves of T is m . Thus, the number of neighbors of r in T is $|N_T(r)| \leq k$.

We first prove, by contradiction, that $L_1(u) \neq L_1(u')$ for all $u, u' \in N_T(r)$ with $u \neq u'$. Suppose that there exist two nodes $u, u' \in N_T(r)$ such that $L_1(u) = L_1(u')$. Since $|e| = 2$ for all $e \in \mathcal{E}$, then u and u' are not leaves of T . Thus, there exist two nodes $u_1, u'_1 \in V(T)$ such that $u_1 \in N_T(u) \setminus \{r\}$ and $u'_1 \in N_T(u') \setminus \{r\}$. Then, for all possible values for $L_2(u)$ and for $L_2(u')$, the second property of Definition 2 is not satisfied for the path (r, u, u_1) that represents $\{L_1(u), L_1(u_1)\} \in \mathcal{E}$ or for the path (r, u', u'_1) that represents $\{L_1(u'), L_1(u'_1)\} \in \mathcal{E}$. Indeed, we cannot have $L_2(u) < L_2(u')$ and $L_2(u') < L_2(u)$. We get a contradiction because T is a local tree representation of \mathcal{G} . Thus, $L_1(u) \neq L_1(u')$ for all $u, u' \in N_T(r)$ with $u \neq u'$.

Let $N_T(r) = \{u_1, \dots, u_k\}$. Recall that $|N_T(r)| \leq k$ because the number of leaves of T is m and $|V(T)| \leq m + k + 1$. Without loss of generality, set $L_1(u_i) = v_i$ for all $i \in \llbracket 1, k \rrbracket$. Thus, for every $e = \{v, v'\} \in \mathcal{E}$, there exists $i \in \llbracket 1, n \rrbracket$ such that $v_i \in \{v, v'\}$ and such that there exists $u \in N_T(u_i) \setminus \{r\}$ with $L_1(u) \in \{v, v'\}$, $v_i \neq L_1(u)$. We deduce that every edge $e \in \mathcal{E}$ is covered by $X = \{v_1, \dots, v_k\}$, and so that there exists a vertex cover of \mathcal{G} of size $|X| = k$.

Since the decision variant of the vertex cover problem is NP-complete even for cubic graphs [5] and for planar graphs of degree at most 3 [3], then the decision variants of the tree representation problems are NP-complete for these two classes of graphs. \blacksquare

Despite this NP-hardness result, a maximal matching of \mathcal{G} , that can be greedily obtained in linear time, gives a 2-approximation algorithm for the vertex cover problem, and so for the tree representation problems (Corollary 11).

Corollary 11 *Let \mathcal{G} be any graph. There is a linear time algorithm that computes \max such that $2 \max \geq \max^* = \max_{local}^* = \max_{global}^*$.*

3.2 Tree representations of bounded degree hypergraphs

In this section, we study the relation between the complexity of the tree representation problems and the maximum degree $\Delta_{\mathcal{H}} = \max_{v \in \mathcal{V}} |\mathcal{E}[v]|$ of the hypergraphs. We prove that the tree representation problems are in P when $\Delta_{\mathcal{H}} \leq 2$, are NP-complete even if $\Delta_{\mathcal{H}} = 3$, and are APX-hard for bounded degree hypergraphs. For the latter class of instances, we also show that there exist polynomial time constant factor approximation algorithms.

3.2.1 A polynomial time algorithm for hypergraphs of degree two

In Theorem 12, we prove a polynomial time algorithm for the global and local tree representation problems for hypergraphs of maximum degree at most two. To do that, we first define the notion of *intersection edge-weighted graph* of a hypergraph and the notion of *node-weighted line graph*. Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph. The intersection edge-weighted graph $G = (\mathcal{E}, I, w)$ of \mathcal{H} is such that for all nodes $e, e' \in \mathcal{E}$, there is an edge $\{e, e'\} \in I$ if and only if $e \cap e' \neq \emptyset$. Furthermore, $w_{e, e'} = |e \cap e'|$ for all $\{e, e'\} \in I$. The node-weighted line graph $L = (I, E', w')$ of G is such that for all nodes $i_1 = \{e_1, e'_1\}, i_2 = \{e_2, e'_2\} \in I$, there is an edge $\{i_1, i_2\} \in E'$ if and only if

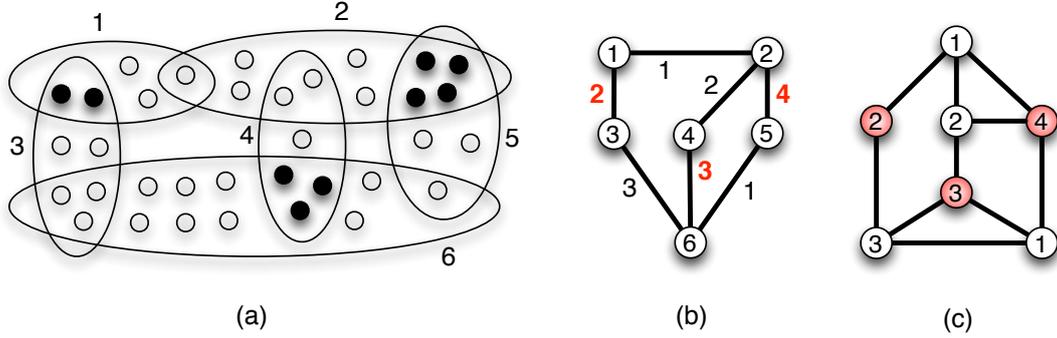


Figure 4: (a) Hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ such that $|\mathcal{E}[v]| \leq 2$ for all $v \in \mathcal{V}$. Let $\mathcal{E} = \{e_1, e_2, e_3, e_4, e_5, e_6\}$. An edge with $j \in \llbracket 1, 6 \rrbracket$ represents $e_j \in \mathcal{E}$. The set of filled nodes represents the maximal cardinality subset of nodes of \mathcal{V} that can be factorized in the tree representation. (b) Intersection edge-weighted graph $G = (\mathcal{E}, I, w)$ constructed from \mathcal{H} . A node with $j \in \llbracket 1, 6 \rrbracket$ represents $e_j \in \mathcal{E}$. For every $i \in I$, the weight w_i is represented by the integer $j \in \mathbb{N}$. (c) Node-weighted line graph $L = (I, E', w')$ of G . For every $i \in I$, the weight w_i is represented by the integer $j \in \mathbb{N}$. The set of filled red nodes represents a maximum weighted independent set of L of total weight 9. From Theorem 12, $\max_{local}^* = \max_{global}^* = 9$.

$\{e_1, e'_1\} \cap \{e_2, e'_2\} \neq \emptyset$. Furthermore, $w'_i = w_{e, e'}$ for all $i = \{e, e'\} \in I$. Figure 4 illustrates the previous constructions and Theorem 12.

Theorem 12 *The global and local tree representation problems are in P for hypergraphs with maximum degree at most two.*

Proof Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph such that $|\mathcal{E}[v]| \leq 2$ for all $v \in \mathcal{V}$. Thus, $|\{u \in V(T_{global}^*) \mid L_1(u) = v\}| \leq 2$ for all $v \in \mathcal{V}$. Then, the global tree representation problem is equivalent to the problem of maximizing the number of nodes u such that $|\{u \in V(T_{global}^*) \mid L_1(u) = v\}| = 1$. Observe that $|\{u \in V(T_{global}^*) \mid L_1(u) = v\}| = 1$ for all $v \in \mathcal{V}$ such that $|\mathcal{E}[v]| = 1$. Thus, in the following, we only consider nodes $v \in \mathcal{V}$ such that $|\mathcal{E}[v]| = 2$.

For any two nodes $v, v' \in \mathcal{V}$ such that $\mathcal{E}[v] \cap \mathcal{E}[v'] \neq \emptyset$ and $\mathcal{E}[v] \neq \mathcal{E}[v']$ (that is such that there exist a hyperedge containing both v and v' , a hyperedge containing v but not containing v' , and a hyperedge containing v' but not containing v), then we have $|\{u \in V(T_{global}^*) \setminus \{r\} \mid L_1(u) = v\}| + |\{u' \in V(T_{global}^*) \setminus \{r\} \mid L_1(u') = v'\}| \geq 3$. Thus, by the first remark, we cannot have $|\{u \in V(T_{global}^*) \setminus \{r\} \mid L_1(u) = v\}| = 1$ and $|\{u' \in V(T_{global}^*) \setminus \{r\} \mid L_1(u') = v'\}| = 1$. Equivalently, this means that two nodes $v, v' \in \mathcal{V}$, v is in the intersection $i_1 = \{e_1, e'_1\} \in I$ and v' is in the intersection $i_2 = \{e_2, e'_2\} \in I$, $\{i_1, i_2\} \in E'$, are such that $|\{u \in V(T_{global}^*) \setminus \{r\} \mid L_1(u) = v\}| + |\{u' \in V(T_{global}^*) \setminus \{r\} \mid L_1(u') = v'\}| \geq 3$. Thus, since the global tree representation problem is equivalent to maximize the number of nodes u such that $|\{u \in V(T_{global}^*) \mid L_1(u) = v\}| = 1$, then the global tree representation problem is equivalent to the maximum weighted independent set problem for L . Finally, since L is a node-weighted line graph of a graph, then the maximum weighted independent set problem for L is in P, and so the global tree representation problem is in P. \blacksquare

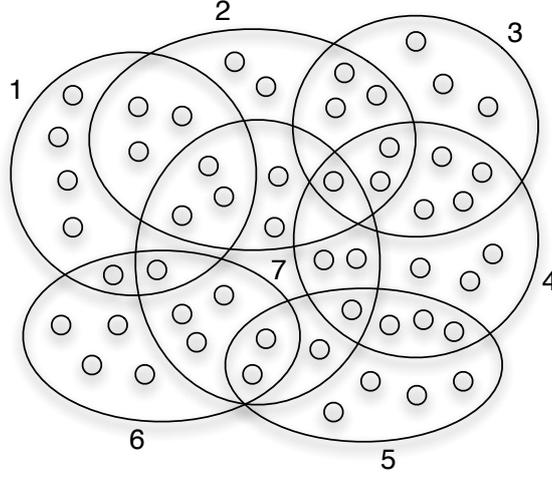


Figure 5: Hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{E} = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$. A hyperedge with $i \in \llbracket 1, 7 \rrbracket$ represents $e_i \in \mathcal{E}$. The auxiliary instance (U, S) of the weighted set packing problem from \mathcal{H} is such that $U = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$ and $S = \{\{e_1, e_2\}, \{e_1, e_6\}, \{e_1, e_7\}, \{e_2, e_3\}, \{e_2, e_4\}, \{e_2, e_7\}, \{e_3, e_4\}, \{e_3, e_7\}, \{e_4, e_5\}, \{e_4, e_7\}, \{e_5, e_6\}, \{e_5, e_7\}, \{e_6, e_7\}, \{e_1, e_2, e_7\}, \{e_1, e_6, e_7\}, \{e_2, e_3, e_7\}, \{e_2, e_3, e_4\}, \{e_3, e_4, e_7\}, \{e_2, e_4, e_7\}, \{e_4, e_5, e_7\}, \{e_5, e_6, e_7\}, \{e_2, e_3, e_4, e_7\}\}$. Furthermore, $w_{\{e_1, e_2\}} = 6$, $w_{\{e_1, e_6\}} = 2$, $w_{\{e_1, e_7\}} = 4$, $w_{\{e_2, e_3\}} = 6$, $w_{\{e_2, e_4\}} = 3$, $w_{\{e_2, e_7\}} = 6$, $w_{\{e_3, e_4\}} = 7$, $w_{\{e_3, e_7\}} = 1$, $w_{\{e_4, e_5\}} = 4$, $w_{\{e_4, e_7\}} = 4$, $w_{\{e_5, e_6\}} = 2$, $w_{\{e_5, e_7\}} = 4$, $w_{\{e_6, e_7\}} = 6$, $w_{\{e_1, e_2, e_7\}} = 9$, $w_{\{e_1, e_6, e_7\}} = 7$, $w_{\{e_2, e_3, e_7\}} = 7$, $w_{\{e_2, e_3, e_4\}} = 10$, $w_{\{e_3, e_4, e_7\}} = 8$, $w_{\{e_2, e_4, e_7\}} = 7$, $w_{\{e_4, e_5, e_7\}} = 5$, $w_{\{e_5, e_6, e_7\}} = 8$, and $w_{\{e_2, e_3, e_4, e_7\}} = 11$. For instance, $\{e_2, e_3, e_4\} \in S$ and $w_{\{e_2, e_3, e_4\}} = 10$ because $|e_2 \cap e_3 \cap e_4| = 3$ and $\max(w_{\{e_2, e_3\}}, w_{\{e_2, e_4\}}, w_{\{e_3, e_4\}}) - 3 = 4$. The packing $C^* = \{\{e_1, e_2\}, \{e_3, e_4\}, \{e_5, e_6, e_7\}\}$ is such that $\sum_{c \in C^*} w_c = 21$ and is optimal for the weighted set packing problem for (U, S) . From Theorem 13, we get that $\max_{local}^* = \max_{global}^* = 21$.

3.2.2 NP-completeness results for hypergraphs of degree three

We prove in Theorem 13 that if the maximum degree of the hypergraph is three, then the global and local tree representation problems are equivalent to the problem of computing a maximum weighted set packing of an auxiliary instance, and so are NP-complete. Let us first define the *weighted set packing problem*. Let U be any set of elements and let S be any set of weighted subsets of U . A subset $C \subseteq S$ is a *packing* of S if and only if for all $c, c' \in C$, $c \neq c'$, then $c \cap c' = \emptyset$. The weighted set packing problem consists in computing the maximum k such that there exists a packing C of S with $\sum_{c \in C} w_c = k$, where w_c is the weight of $c \in C$. The decision variant of the weighted set packing problem is a well known NP-complete problem [4, 6]. We formalize in Definition 4 the auxiliary instance of the weighted set packing problem constructed from a hypergraph.

Definition 4 (instance of weighted set packing problem from a hypergraph) Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph with $\mathcal{E} = \{e_1, \dots, e_m\}$. The instance (U, S) of the weighted set packing problem from \mathcal{H} is defined as follows. Set $U = \mathcal{E}$. Let $e_{s_1}, \dots, e_{s_t} \in \mathcal{E}$ be any t hyperedges, with $2 \leq t \leq m$. The subset $\{e_{s_1}, \dots, e_{s_t}\} \in S$ if and only if $e_{s_1} \cap \dots \cap e_{s_t} \neq \emptyset$. Furthermore, $w_s = |e_{s_1} \cap e_{s_2}|$ if $t = 2$ and $w_s = (t-1)|e_{s_1} \cap \dots \cap e_{s_t}| + (t-2)(\max_{e \in s} (w_{\{e_{s_1}, \dots, e_{s_t}\} \setminus \{e\}}) - |e_{s_1} \cap \dots \cap e_{s_t}|)$ if $t \geq 3$.

The set $U = \mathcal{E}$ of the elements represents all the hyperedges. The set S represents all the different non-empty intersections of all the hyperedges of \mathcal{E} . Note that $w_s \in \mathbb{N}$ for all $s \in S$. Figure 5 depicts an example of auxiliary instance and illustrates Theorem 13.

Theorem 13 *The decision variants of the global and local tree representation problems are NP-complete even for hypergraphs with maximum degree three.*

Proof Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be any hypergraph and let (U, S) be the instance of the weighted set packing problem constructed from \mathcal{H} . Let $k \geq 1$ be any integer. Recall that w_s represents the weight of any subset $s \in S$. We prove that there exists a packing C of S such that $\sum_{c \in C} w_c \geq k$ if and only if $\max_{local}^* \geq k$. Then, we will prove that $\max_{local}^* = \max_{global}^*$.

\Rightarrow Suppose that there exists a packing $C = \{c_1, \dots, c_q\}$ of S such that $\sum_{c \in C} w_c \geq k$, where $q \geq 1$ is an integer such that, without loss of generality, for every $e \in \mathcal{E}$, there exists an i , $1 \leq i \leq q$, such that $e \in c_i$. We prove that $\max_{local}^* \geq k$. By definition of C , we have $c_i \cap c_j = \emptyset$ for all i, j , $1 \leq i < j \leq q$. For all i , $1 \leq i \leq q$, let $t_i = |c_i|$ and $c_i = \{e_i^1, \dots, e_i^{t_i}\}$. Without loss of generality, assume that $w_{e_i^1, \dots, e_i^j} = (j-1)|e_i^1 \cap \dots \cap e_i^j| + (j-2)(w_{e_i^1 \cap \dots \cap e_i^{j-1}} - |e_i^1 \cap \dots \cap e_i^j|)$ for all i, j , $1 \leq i \leq q$, $1 \leq j \leq t_i$. For all i, j , $1 \leq i \leq q$, $1 \leq j \leq t_i$, let $Y_i^j = (e_i^1 \cap \dots \cap e_i^j) \setminus (e_i^{j+1} \cup \dots \cup e_i^{t_i})$.

Let us construct $T_{local} = (V, E, L_1, L_2)$ rooted at $r \in V$ as follows. For all $i = 1, \dots, q$, for all $j = t_i, \dots, 1$, we add $|Y_i^j|$ nodes in T , namely $u_i^{j,x}$, where $x = 1, \dots, |Y_i^j|$. For all $i = 1, \dots, q$, for all $j = t_i, \dots, 1$, for all $x = 1, \dots, |Y_i^j|$, for all $x' = x+1, \dots, |Y_i^j|$, we choose $L_1(u_i^{j,x}) \in Y_i^j$ such that $L_1(u_i^{j,x}) \neq L_1(u_i^{j,x'})$.

For all $i = 1, \dots, q$, for all $j = t_i, \dots, 1$, for all $x = 1, \dots, |Y_i^j| - 1$, we add an edge $\{u_i^{j,x}, u_i^{j,x+1}\} \in E(T_{local})$. For all $i = 1, \dots, q$, we add an edge $\{r, u_i^{t_i,1}\} \in E(T_{local})$. For all $i = 1, \dots, q$, for all $j = t_i, \dots, 2$, we add an edge $\{u_i^{j,|Y_i^j|}, u_i^{j-1,1}\} \in E(T_{local})$.

Let $Y = \cup_{i=1}^q \cup_{j=1}^{t_i} Y_i^j$. For all $i = 1, \dots, q$, for all $j = t_i, \dots, 1$, let $\{u_i^{j,1}, \dots, u_i^{j,t_i}\} = (V \cap e_i^j) \setminus Y$. For all $i = 1, \dots, q$, for all $j = t_i, \dots, 1$, let $P_{i,j}$ be the longest path in T between r and $u_{i,j}$ such that for all $u \in V(P_{i,j})$, then $L_1(u) \in e_i^j$. For all $i = 1, \dots, q$, for all $j = t_i, \dots, 1$, we add in T the path $(u_{i,j}, u_i^{j,1}, \dots, u_i^{j,t_i})$; we obtain the path $P'_{i,j}$ between r and u_i^{j,t_i} . We define $L_1(u) \in e_i^j$ such that $L_1(u) \neq L_1(u')$ for all $u, u' \in V(P'_{i,j}) \setminus \{r\}$, $u \neq u'$.

We now easily define L_2 such that every simple path $P_e = (r, u_1, \dots, u_{|e|})$ representing the hyperedge $e \in \mathcal{E}$ is such that $L_2(u_i) < L_2(u_j)$ for all i, j , $1 \leq i < j \leq |e|$.

The tree T_{local} satisfies the properties of Definition 2, and the number of nodes of T_{local} is $|V(T_{local})| \leq (\sum_{e \in \mathcal{E}} |e|) - k$, and so $\max_{local}^* \geq k$.

\Leftarrow Suppose that $\max_{local}^* \geq k$, that is there exists T_{local} such that $|V(T_{local})| \leq (\sum_{e \in \mathcal{E}} |e|) - k$. We prove that there exists a packing $C = \{c_1, \dots, c_q\}$ of S such that $\sum_{c \in C} w_c \geq k$, where $q \geq 1$ is an integer. We prove the result by induction on the number of hyperedges. It is true when the hypergraph contains one or two hyperedges. Suppose it is true when there are at most $|\mathcal{E}| - 1$ hyperedges. We prove that it is also true with $|\mathcal{E}|$ hyperedges.

Let $N_{T_{local}}(r) = \{u_1, \dots, u_q\}$, where q is the degree of the root r . We construct the packing $C = \{c_1, \dots, c_q\}$ of S as follows. For all i , $1 \leq i \leq q$, let $c_i = \{e_i^1, \dots, e_i^{t_i}\}$ such that $L_1(u_i) \in e_i^j$ and $L_1(u_i) \notin e_i^j$ for all j , $1 \leq j \leq t_i$ and for all $e \in \mathcal{E} \setminus c_i$. Since T_{local} satisfies the properties of Definition 2, then $c_i \cap c_j = \emptyset$ for all i, j , $1 \leq i < j \leq q$. Consider the q subproblems induced by the neighbors of r , that is the q subtrees $T_{local}^1, \dots, T_{local}^q$ rooted at u_1, \dots, u_q , respectively. By induction hypothesis and by the previous remark, there exists a packing C_i such that $\sum_{c \in C_i} w_c \geq x_{local}^i$ for all i , $1 \leq i \leq q$. Thus, the packing C is such that $\sum_{c \in C} w_c \geq k = \sum_{i=1}^q x_{local}^i$.

The trees previously described satisfy the properties of Definition 2 and the properties of Definition 3. Thus, $\max_{local}^* = \max_{global}^*$. To conclude, since the decision variant of the weighted

set packing problem is NP-complete, then the decision variants of the global and local tree representation problems are NP-complete. ■

3.2.3 APX-hardness results and constant factor approximations

From the proof of Theorem 13, we deduce in Corollary 14 the APX-hardness of the global and local tree representation problems for the class of bounded degree hypergraphs. Indeed, this negative result comes from the inapproximability ratio of the weighted set packing problem [4] because, from the proof of Theorem 13, we have $\max_{global}^* = \max_{local}^* = \sum_{c \in C^*} w_c$, where C^* is an optimal solution for the auxiliary instance of the weighted set packing problem.

Corollary 14 *The global and local tree representation problems cannot be approximated within $\frac{k}{\log(k)}$ -ratio in polynomial time for the class of hypergraphs of maximum degree k , where $k \geq 3$ is a constant integer, unless $P=NP$.*

Despite this negative result, we deduce from the proof of Theorem 13 that the global and local tree representation problems admit a polynomial time $\frac{k}{2}$ -approximation algorithm for hypergraphs of maximum degree k , where $k \geq 3$ is a constant integer (Corollary 15). Indeed, the $\frac{k}{2}$ -approximation algorithm for the weighted set packing problem proved in [4], gives a $\frac{k}{2}$ -approximation algorithm for our problems. Recall that the auxiliary instance can be constructed in polynomial time.

Corollary 15 *The global and local tree representation problems admit a polynomial time $\frac{k}{2}$ -approximation algorithm for the class of hypergraphs of maximum degree k , where $k \geq 3$ is a constant integer.*

Finally, we deduce in Corollary 16 that the tree representation problems cannot be approximated within any constant factor in polynomial time for the class of unbounded maximum degree hypergraphs, unless $P=NP$. Indeed, as seen before, these problems are equivalent to the weighted set packing problem that is not in APX if the maximum size of subset is not bounded. Furthermore, our result may be stronger because the construction of the auxiliary instance seems impossible in polynomial time.

Corollary 16 *The global and local tree representation problems are not in APX for the class of unbounded maximum degree hypergraphs.*

4 Future works

As future works, we plan to implement the algorithms described in this article and to design new ones (e.g. branch and bound algorithm that guarantees any approximation ratio). We are also studying the problem of representing maximal simplices by directed (acyclic) graphs. We think that these new representations may reduce significantly the size of the representation of simplicial complexes.

References

- [1] J.-D. Boissonnat, C. S. Karthik, and S. Tavenas. Building efficient and compact, data structures for simplicial complexes. Submitted to SoCG 2015.

-
- [2] J.-D. Boissonnat and C. Maria. The simplex tree: An efficient data structure for general simplicial complexes. *Algorithmica*, 70(3):406–427, 2014.
 - [3] M. R. Garey and D. S. Johnson. The rectilinear Steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, 1977.
 - [4] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.
 - [5] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete problems. In *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*, STOC '74, pages 47–63, New York, NY, USA, 1974. ACM.
 - [6] D. S. Hochbaum. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Applied Mathematics*, 6(3):243 – 254, 1983.



**RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE**

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399