On the complexity of the representation of simplicial complexes by trees

Jean-Daniel Boissonnat, Dorian Mazauric

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

Jean-Daniel Boissonnat, Dorian Mazauric
Inria Sophia Antipolis - Méditerranée, France

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 planar graph of maximum degree at most three or a bounded degree hypergraph. However, we prove polynomial time algorithms that compute constant factor approximations and optimal solutions for some classes of instances.

Keywords: simplicial complexes, hypergraphs, tree representations, NP-completeness, APX-completeness, approximation algorithms.

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. Consequenlty, the use of simplicial complexes is limited in pratice. 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

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

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 $N_{\mathcal{H}}(v) = \{v' \in V \mid v' \neq v, v' \in e, v \in e, e \in \mathcal{E}\}$. 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 $\sigma = (\sigma_1, \ldots, \sigma_n) \in \Sigma(\mathcal{V})$ be any ordering of $\mathcal{V}$. For every $e \in \mathcal{E}$, we define $\sigma^e = (\sigma_1^e, \ldots, \sigma_{|e|}^e)$ 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 $r \in V$, $L_1 : V \to \mathcal{V}$, 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,

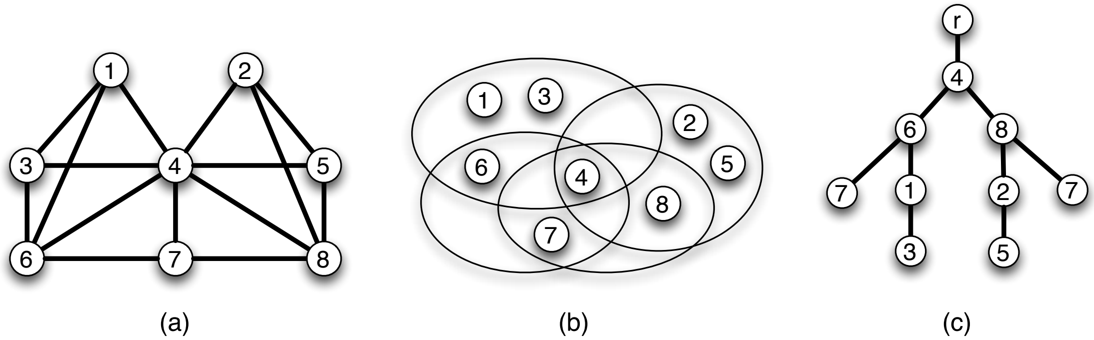


Figure 1: **(a)** Simplicial complex $\mathcal{K}$ composed of eight vertices $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$. A vertex with $i \in [\![1, 8]\!]$ represents v_i. The set of maximal simplices is composed of two tetrahedrons induced by $\{v_1, v_3, v_4, v_6\}$ and $\{v_2, v_4, v_5, v_8\}$,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

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 [BM14] for representing simplicial complexes in a compact way. Recently, in [BCST15], 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 [BCST15]) 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.

We then analyze the complexity of the combinatorial optimization problems, namely the tree representation problems, introduced in this paper. In Section 3, we prove that these problems are equivalent when the maximal simplices are all of size two (class of graphs), are NP-complete even in the class of planar graphs of maximum degree at most three, and admit a linear time

2-approximation algorithm for this class of instances.

In Section 4, we 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 APX-complete even when all the vertices are in at most three maximal simplices (hypergraphs of maximum degree three), 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 $e = \{a_1, a_2, a_3, a_4, b_1, c_1\}$ belongs to the hypergraph $\mathcal{H}$ represented by T. Indeed, the root r of T has two neigbhors $u, u' \in N_T(r)$ 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 $(r, u, \ldots)$ or by a path $(r, u', \ldots)$.

Lemma 1 (simplex search algorithm for tree representation) *Let $\boldsymbol{A}$ 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 $\boldsymbol{A}$ 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 $\mathcal{V} = \{v_1, \ldots, v_{2n}\}$ and $\mathcal{E} = \{e_1, \ldots, e_n\}$ with $e_i = \{v_i, v_{n+1}, \ldots, v_{2n}\}$ for every $i \in [\![1, n]\!]$. We construct a tree representation $T = (V, E, L_1)$ rooted at $r \in V$ of $\mathcal{H}$ as follows. Let $V = \{r\} \cup V_1 \cup \ldots \cup V_n \cup \{u^1, \ldots, u^n\}$ with $V_i = \{u_i^{n+1}, \ldots, u_i^{2n}\}$. 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 $j \in [\![n+1, 2n]\!]$ and for every $i \in [\![1, n]\!]$.

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 $i \in [\![1, n]\!]$, the path $P_i = (r, u_i^{n+1}, \ldots, u_i^{2n})$ 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 $i \in [\![1, n]\!]$. Thus, the time complexity of Algorithm $\mathbf{A}$ is $\Omega(|\mathcal{V}|^2) = \Omega(|V(T)|)$. $\square$

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 $r \in V$, $L_1 : V \to \mathcal{V}$, $L_2 : V \to [\![0, |\mathcal{V}|]\!]$, 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.

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 $r \in V$, $L_1 : V \to \mathcal{V}$, $L_2 : V \to [\![0, |\mathcal{V}|]\!]$, 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 $L_2(u) < L_2(u')$.*

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 $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ 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 $(r, u, \ldots)$ 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 $(r, u', \ldots)$ 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. $\square$

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.

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 $r \in V$, $L_1 : V \to \mathcal{V}$, $L_2 : V \to [\![0, |\mathcal{V}|]\!]$, constructed as follows. Set $T_\sigma = (\{r\}, \emptyset)$. For all $e \in \mathcal{E}$:*

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

Recall that, for every $e \in \mathcal{E}$, $\sigma^e = (\sigma_1^e, \ldots, \sigma_{|e|}^e)$ 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.

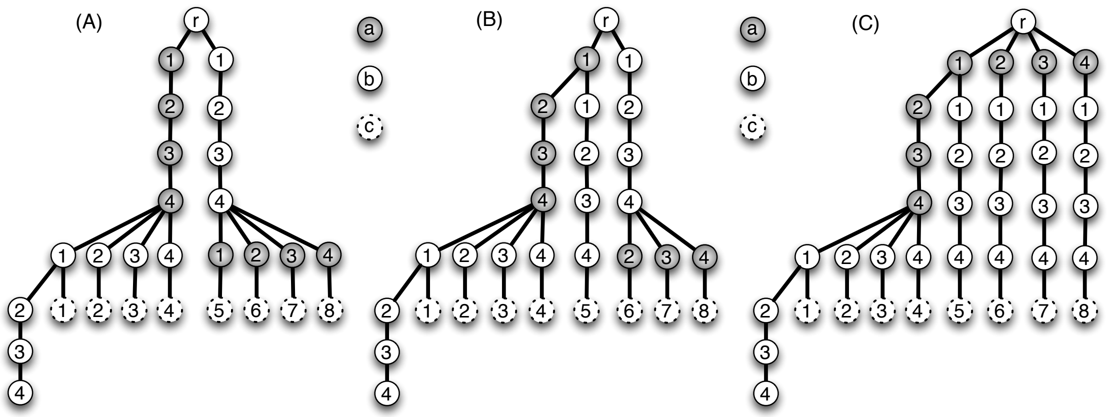


Figure 2: Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{a_1, \ldots, a_4, b_1, \ldots, b_4, c_1, \ldots, c_8\}$ and $\mathcal{E} = \{\{a_1, a_2, a_3, a_4, b_1, c_1\}$, $\{a_1, a_2, a_3, a_4, b_2, c_2\}$, $\{a_1, a_2, a_3, a_4, b_3, c_3\}$, $\{a_1, a_2, a_3, a_4, b_4, c_4\}$, $\{b_1, b_2, b_3, b_4, a_1, c_5\}$, $\{b_1, b_2, b_3, b_4, a_2, c_6\}$, $\{b_1, b_2, b_3, b_4, a_3, c_7\}$, $\{b_1, b_2, b_3, b_4, a_4, c_8\}$, $\{a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4\}\}$. A filled black node u with $i \in [\![1, 4]\!]$ is such that $L_1(u) = a_i$ and $L_2(u) = i$. A full line node u with $i \in [\![1, 4]\!]$ is such that $L_1(u) = b_i$ and $L_2(u) = i + 4$. A dashed node u with $i \in [\![1, 8]\!]$ is such that $L_1(u) = c_i$ and $L_2(u) = i + 8$. **(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 $\mathcal{E}_{a_1} = \{\{a_2, a_3, a_4, b_1, c_1\}$, $\{a_2, a_3, a_4, b_2, c_2\}$, $\{a_2, a_3, a_4, b_3, c_3\}$, $\{a_2, a_3, a_4, b_4, c_4\}$, $\{b_1, b_2, b_3, b_4, c_5\}\}$. 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 $\bar{\mathcal{E}}_{a_1} = \{\{b_1, b_2, b_3, b_4, a_2, c_6\}$, $\{b_1, b_2, b_3, b_4, a_3, c_7\}$, $\{b_1, b_2, b_3, b_4, a_4, c_8\}\}$. 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 $\sigma = (a_1, \ldots, a_4, b_1, \ldots, b_4, c_1, \ldots, c_8)$.

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 $r \in V$, $L_1 : V \to \mathcal{V}$, $L_2 : V \to [\![0, |\mathcal{V}|]\!]$, 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 $P = (r, u_1, \ldots, u_t)$ of T, then $L_2(u_i) < L_2(u_{i+1})$ for all $i \in [\![1, t-1]\!]$.*

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 $\sigma = (\sigma_1, \ldots, \sigma_n)$ 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 $\sigma = (\sigma_1, \ldots, \sigma_n)$ 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 $\sigma = (\sigma_1, \ldots, \sigma_n)$ 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 $(r, u_1^1, \ldots, u_{|e_1|}^1)$ 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 $i \in [\![1, |e_1|]\!]$. 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 $\{e_1, \ldots, e_{m'}\}$. We now represent the hyperedge $e_{m'+1}$. We add in T_σ the path $(u_{t+1}^{m'+1}, \ldots, u_{|e|}^{m'+1})$ 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 $P = (r = u_0, u_1, \ldots, u_t)$ in T_σ with $L_1(u_i) = \sigma_i^{m'+1}$ for all $i \in [\![1, t]\!]$. 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 $i \in [\![t+1, |e_{m'+1}|]\!]$. Thus, T_σ satisfies the properties of Lemma 3 for the hypergraph induced by the set of hyperedges $\{e_1, \ldots, e_{m'+1}\}$. $\square$

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 $(r, u, \ldots)$ 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. $\square$

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 $\sigma = (\sigma_1, \ldots, \sigma_n)$ 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 $(r, u, \ldots)$ 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. $\square$

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 1.

Corollary 1 (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 2.

Corollary 2 (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 $\max^* = 29$ and $|V(T^*)| = 28$ (Figure 2(A)), $\max^*_{local} = 26$ and $|V(T^*_{local})| = f(\mathcal{V}, \mathcal{E}) + 1 = 31$ (Figure 2(B)), and $\max^*_{global} = 18$ and $|V(T^*_{global})| = 39$ (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 6 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 6 *For any $n \geq 1$, there exists a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, with $n = |\mathcal{V}|/4$, such that $|V(T^*_{local})| \leq 8n$ and $|V(T^*_{global})| \geq n^2$.*

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

Note that we do not consider the set of nodes C because every $c \in C$ appears in a unique hyperedge and so we cannot factorize any node of the tree corresponding to a node of C. Thus, without loss of generality, we represent these nodes as leaves of the tree. Furthermore, we do not represent the hyperedge $\{a_1, \ldots, a_n, b_1, \ldots, b_n\}$. Indeed, for any tree representation for $\mathcal{E}$ defined above, adding this hyperedge consists in adding $n-1$ nodes (as path) in this tree representation. Thus, we add $2n-1$ nodes in order to get the tree representation for the orginal hypergraph.

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

Proof. Let us construct a local tree representation T_{local} rooted at $r \in V(T_{local})$ such that $|V(T_{local})| = 5n$. Let $V(T_{local}) = \{r\} \cup \{u_1, \ldots, u_n\} \cup \{v_1, \ldots, v_n\} \cup \{u'_2, \ldots, u'_n\} \cup \{v'_1, \ldots, v'_n\} \cup \{w_1, \ldots, w_n\}$. 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 $i \in [\![1, n]\!]$. Set $L_1(u'_i) = a_i$ and $L_2(u'_i) = i$ for all $i \in [\![2, n-1]\!]$. 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 $i \in [\![1, n]\!]$, we associate the path $(r, u_1, \ldots, u_n, v'_i)$ with the hyperedge $\{a_1, \ldots, a_n, b_i\}$. For all $i \in [\![2, n]\!]$, we associate the path $(r, v_1, \ldots, v_n, u'_i)$ with the hyperedge $\{b_1, \ldots, b_n, a_i\}$. We finally associate the path $(r, u_1, v_1, \ldots, v_n)$ with the hyperedge $\{b_1, \ldots, b_n, a_1\}$. By construction of L_2, T_{local} is an admissible solution for Problem 2. We get that $|V(T^*_{local})| \leq |V(T_{local})| = 5n$. $\square$

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

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

$$\begin{cases} \sigma_i \in A \text{ for all } i \in [\![1 + \sum_{k=1}^{j-1} \alpha_k + \beta_k, 1 + \alpha_j + \sum_{k=1}^{j-1} \alpha_k + \beta_k]\!], \text{ for all } j \in [\![1, n]\!]; \\ \sigma_i \in B \text{ for all } i \in [\![1 + \alpha_j + \sum_{k=1}^{j-1} \alpha_k + \beta_k, 1 + \sum_{k=1}^{j} \alpha_k + \beta_k]\!] \text{ for all } j \in [\![1, n]\!]; \\ \alpha_j \geq 1 \text{ for all } j \in [\![1, t_a]\!]; \alpha_j = 0 \text{ for all } j \in [\![t_a + 1, n]\!]; \\ \beta_j \geq 1 \text{ for all } j \in [\![1, t_b]\!]; \beta_j = 0 \text{ for all } j \in [\![t_b + 1, n]\!]; \\ \sum_{k=0}^{n} \alpha_k = n; \sum_{k=0}^{n} \beta_k = n. \end{cases}$$

For any two subsets of nodes $A' = \{a'_1, \ldots, a'_{|A'|}\} \subseteq A$ and $B' = \{b'_1, \ldots, b'_{|B'|}\} \subseteq B$, consider the hypergraph $(\mathcal{V}_{A',B'}, \mathcal{E}_{A',B'})$, where $\mathcal{V}_{A',B'} = A' \cup B'$ and $\mathcal{E}_{A',B'} = \{\{a'_1, \ldots, a'_{|A'|}, b\} \mid b \in B'\}$ $\cup \{\{b_1, \ldots, b_{|B'|}, a\} \mid a \in A'\}$. 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 $i \in [\![1, \alpha_1]\!]$, and that $\sigma_i \in B$ for all $i \in [\![\alpha_1 + 1, \alpha_1 + \beta_1]\!]$. Let us consider $\mathcal{V}_{A_1,B_1} = A_1$ $\cup B_1$ with $A_1 = \{a_1, \ldots, a_{\alpha_1}\}$ and $B_1 = \{b_1, \ldots, b_{\beta_1}\}$, and $\mathcal{E}_{A_1,B_1} = \{\{a_1, \ldots, a_{\alpha_1}, b\} \mid b \in B_1\}$ $\cup \{\{b_1, \ldots, b_{\beta_1}, a\} \mid a \in A_1\}$. 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 $(a_1, \ldots, a_{\alpha_1}, b_1, \ldots, b_{\beta_1})$, is such that $|V(T_{A_1,B_1})| = \alpha_1\beta_1 + 2\alpha_1$. More precisely, $V(T_{A_1,B_1}) = \{r\} \cup \{u_1, \ldots, u_{\alpha_1}\} \cup \{u'_2, \ldots, u'_{\alpha_1}\} \cup \{v_1, \ldots, v_{\beta_1}\} \cup_{i=1}^{\alpha_1} \{v^i_1, \ldots, v^i_{\beta_1}\}$. Furthermore, $L_1(u_j) = a_j$ for all $j \in [\![1, \alpha_1]\!]$; $L_1(u'_j) = a_j$ for all $j \in [\![2, \alpha_1]\!]$; $L_1(v_j) = b_j$ for all $j \in [\![1, \beta_1]\!]$; $L_1(v^i_j) = b_j$ for all $j \in [\![1, \beta_1]\!]$ and for all $i \in [\![1, \alpha_1]\!]$. 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 $\mathcal{E}'_{A_1,B_1} = \{\{a_1, \ldots, a_n, b\} \mid b \in B_1\} \cup \{\{b_1, \ldots, b_n, a\} \mid a \in A_1\}$, 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 $\{a_{\alpha_1+1}, \ldots, a_n, b_{\beta_1+1}, \ldots, b_n\}$. 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 $\{\{a_1, \ldots, a_n, b\} \mid b \in B \setminus B_1\} \cup \{\{b_1, \ldots, b_n, a\} \mid a \in A \setminus A_1\}$, 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 $\{r, u_1, \ldots, u_{\alpha_1}, u'_2, \ldots, u'_{\alpha_1}\}$, because there is a unique hyperedge that contains the subset of nodes $\{a_1, \ldots, a_{\alpha_1, b_j}\}$ for all $j \in [\![1, n]\!]$, and there is a unique hyperedge that contains the subset of nodes $\{b_1, b_2, a_j\}$ for all $j \in [\![1, n]\!]$.

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 $\mathcal{E}' = \{\{a_{\alpha_1+1}, \ldots, a_n, b\} \mid b \in B \setminus B_1\} \cup$

$\{\{b_{\beta_1+1}, \ldots, b_n, a\} \mid a \in A \setminus A_1\}$. 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 $|V(T^*_{global})| \geq n^2$. □

The two previous claims conclude the proof of Lemma 6. □

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 even for graphs

We consider here the class of graphs (hyperedges of size two). We prove in Theorem 1 that the decision variants of the tree representation problem, the local tree representation problem, and the global tree representation problem are NP-complete. Our NP-completeness result holds when the graph has no triangle. On the positive side, we show that there exists a linear time 2-approximation algorithm for these problems. We first prove in Lemma 7 that the three tree representation problems are equivalent in graphs.

Lemma 7 *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 $N_T(r) = \{u_1, \ldots, u_k\}$, 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 $i \in [\![1,k]\!]$ and for every $j \in [\![1,i]\!]$. 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 $N_{T_{global}}(r') = \{u'_1, \ldots, u'_k\}$. Then, for $i = 1, \ldots, k$, 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 $j \in [\![i+1, |N_{T_{global}}(r')|]\!]$, we remove e' from E' and we remove u' fron V'; and we add u'' in V' and we add $\{u'_i, u''\}$ in E' with $L_1(u'') = L_1(u'_j)$. 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 $L_1(X) = \{v_{k+1}, \ldots, v_{|\mathcal{V}|}\}$. Then, for $i = k+1, \ldots, v_{|\mathcal{V}|}$, 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 $|V(T^*_{global})| = |V(T^*_{local})| = |V(T^*)|$. $\square$

In the reduction of the proof of Theorem 1, 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 [GJS74] and for the class of planar graphs of degree at most three [GJ77]. The set $X = \{v_1, v_6, v_8\} \subseteq \mathcal{V}$ of filled black 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 $|V(T^*_{global})| = 1 + |X| + |\mathcal{E}| = 15$, and so $x^*_{global} = 8$.

We now prove that the decision variant of the vertex cover problem is NP-complete even for graphs without triangle. Actually, we prove a stronger result in Lemma 8. Recall that $N_{\mathcal{G}}[v]$ is the close neighborhood of any node $v \in V(G)$ in G, that is $N_{\mathcal{G}}[v] = N_{\mathcal{G}}(v) \cup \{v\}$.

Lemma 8 *The decision variant of the vertex cover problem is NP-complete even if the graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *is planar,* $|N_{\mathcal{G}}(v)| \leq 3$*, and* $|N_{\mathcal{G}}(v) \cap N_{\mathcal{G}}(v')| \leq 1$ *for all* $v, v' \in \mathcal{V}$*,* $v \neq v'$.

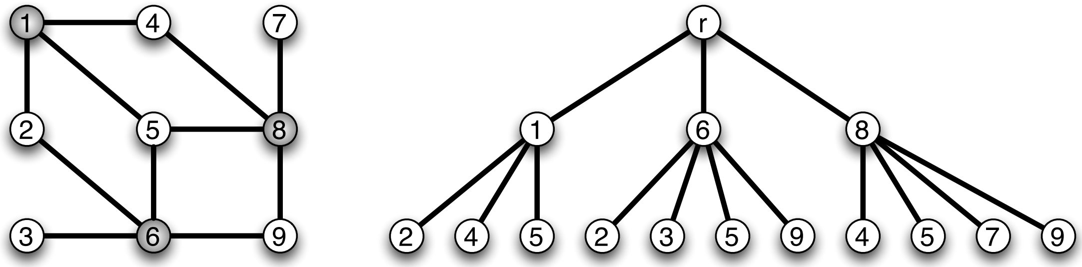
Proof. In [GJ77], the vertex cover problem has been proved NP-complete in the class of planar graphs of degree at most three. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be any planar graph of degree at most three. Let $n = |\mathcal{V}|$ and let $m = |\mathcal{E}|$. Let $k \geq 1$ be any integer. We now construct a graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ from $\mathcal{G}$ such that $|N_{\mathcal{G}'}(u)| \leq 3$ and $|N_{\mathcal{G}'}(u) \cap N_{\mathcal{G}'}(v)| \leq 1$ for all $u, v \in \mathcal{V}'$, $u \neq v$. Let $\mathcal{V} = \{v_1, \ldots, v_n\}$ and let $\mathcal{E} = \{e_1, \ldots, e_m\}$. Set $\mathcal{V}' = \mathcal{V} \cup \{u_1, u'_1, \ldots, u_m, u'_m\}$. Set $\mathcal{E}' = \{\{u_i, u'_i\} \mid 1 \leq i \leq m\} \cup W$, where W is constructed as follows. If $e_t = \{v_i, v_j\} \in \mathcal{E}$, $1 \leq i < j \leq n$, $1 \leq t \leq m$, then $\{v_i, u_t\} \in W$ and $\{v_j, u'_t\} \in W$. To summarize, the graph $\mathcal{G}'$ is constructed from $\mathcal{G}$ by replacing every edge of $\mathcal{G}$ by a path composed of four nodes.

We now prove that there exists a vertex cover X of $\mathcal{G}$ of size $|X| \leq k$ if, and only if, there exists a vertex cover of $\mathcal{G}'$ of size $|X'| \leq k + m$.

$\Rightarrow$ Assume that there exists a vertex cover X of $\mathcal{G}$ of size $|X| \leq k$. We prove that there exists a vertex cover of $\mathcal{G}'$ of size $|X'| \leq k + m$. Without loss of generality, assume that $|X| = k$ and that $X = \{v_1, \ldots, v_k\}$. We construct X' as follows. For every edge $e_t = \{v_i, v_j\} \in \mathcal{E}$, $1 \leq i < j \leq n$, $1 \leq t \leq m$, then $u'_t \in X'$. Note that $v_i \in X$ because $i < j$ and $X \cap \{v_i, v_j\} \neq \emptyset$. Finally we add X into X'. We get that $X' = |X| + m = k + m$. The set X' is a vertex cover of $\mathcal{G}'$ because X is a vertex cover of $\mathcal{G}$ and for every path (v_i, u_t, u'_t, v_j) of $\mathcal{G}'$ that replaces the edge $e_t = \{v_i, v_j\} \in \mathcal{E}$, $1 \leq i < j \leq n$, $1 \leq t \leq m$, then $v_i, u'_t \in X'$.

$\Leftarrow$ Assume that there exists a vertex cover of $\mathcal{G}'$ of size $|X'| \leq k + m$. We prove that there exists a vertex cover X of $\mathcal{G}$ of size $|X| \leq k$. By construction of $\mathcal{G}'$, $\{u_t, u'_t\} \cap X' \neq \emptyset$ for all $t \in [\![1, m]\!]$. Set $X = X' \cap \mathcal{V}$. By the previous remark, $|X| \leq k$. If X is not a vertex cover of $\mathcal{G}$, then it means that there exists an edge $e_t = \{v_i, v_j\} \in \mathcal{E}$, $1 \leq i < j \leq n$, $1 \leq t \leq m$, such that $\{v_i, v_j\} \cap X = \emptyset$. But since X' is a vertex cover of $\mathcal{G}'$, then it means that $u_t, u'_t \in X'$, and so that $|X' \setminus \mathcal{V}| \geq m + 1$ and $|X| < k$. We add v_i in X. If X is a vertex cover of $\mathcal{G}$, then it is done. Otherwise, we apply the same construction while X is not a vertex cover of $\mathcal{G}$.

Finally, we have proved that there exists a vertex cover X of $\mathcal{G}$ of size $|X| \leq k$ if, and only if, there exists a vertex cover of $\mathcal{G}'$ of size $|X'| \leq k + m$. Note that the construction of $\mathcal{G}'$ can be done in polynomial time in the size of $\mathcal{G}$. Thus, the decision variant of the vertex cover problem is NP-complete even if the graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is a planar graph of degree at most 3 and $|N_{\mathcal{G}'}(u) \cap N_{\mathcal{G}'}(v)| \leq 1$ for all $u, v \in \mathcal{V}'$, $u \neq v$. $\square$



(a) (b)

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 $\mathcal{E} = \{\{v_1, v_2\}, \{v_1, v_4\}, \{v_1, v_5\}, \{v_2, v_6\}, \{v_3, v_6\}, \{v_4, v_8\}, \{v_5, v_6\}, \{v_5, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_8, v_9\}\}$. A node with $i \in [\![1, 9]\!]$ represents $v_i \in \mathcal{V}$. The set $X = \{v_1, v_6, v_8\}$ of filled black nodes represents a minimum vertex cover of $\mathcal{G}$. **(b)** Optimal global tree representation T^*_{global} of $\mathcal{G}$. Observe that $|V(T^*_{global})| = 1 + |X| + |\mathcal{E}| = 15$. A node u with $i \in [\![1, 9]\!]$ 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 $L_1(u) = v_9$.

To illustrate the construction described in the proof of Lemma 8, Figure 4(a) represents a planar graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5\}$ and $\mathcal{E} = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}, \{v_2, v_4\}, \{v_3, v_5\}, \{v_4, v_5\}\}$. The maximum degree of $\mathcal{G}$ is three. A node with the integer $i \in [\![1, 6]\!]$ represents $v_i \in \mathcal{V}$. Note that $|N_{\mathcal{G}}(v_2) \cap N_{\mathcal{G}}(v_4)| = 2$. The set of filled black nodes represents a minimum vertex cover $X = \{v_1, v_3, v_4\}$ of $\mathcal{G}$ of size $|X| = 3$. Figure 4(b) represents the planar graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ constructed from $\mathcal{G}$ (proof of Lemma 8). The maximum degree of $\mathcal{G}'$ is three. Furthermore, we have $|N_{\mathcal{G}'}(v) \cap N_{\mathcal{G}'}(v')| \leq 1$ for all $v, v' \in \mathcal{V}'$, $v \neq v'$. The set of filled black nodes in Figure 4(b) represents a minimum vertex cover X' of $\mathcal{G}'$ of size $|X'| = |X| + |\mathcal{E}| = 9$.

We now prove Theorem 1.

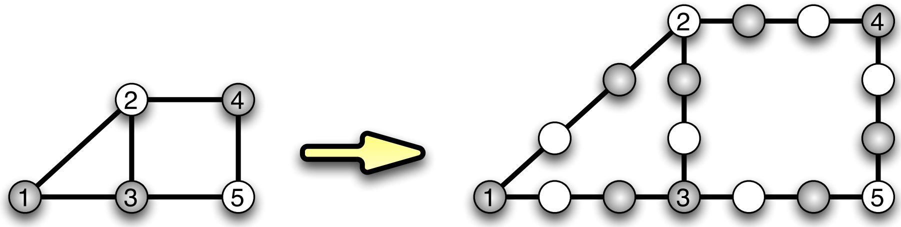
Theorem 1 *The decision variants of the tree representation problems are NP-complete even if the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is planar, $|N_{\mathcal{G}}(v)| \leq 3$, and $|N_{\mathcal{G}}(v) \cap N_{\mathcal{G}}(v')| \leq 1$ for all $v, v' \in \mathcal{V}$, $v \neq v'$.*

Proof. From Lemma 7, 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 $\mathcal{V} = \{v_1, \ldots, v_n\}$. Let $m = |\mathcal{E}|$ and let $\mathcal{E} = \{e_1, \ldots, e_m\}$. 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,



(a) (b)

Figure 4: **(a)** Planar graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of maximum degree three, with $\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5\}$ and $\mathcal{E} = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}, \{v_2, v_4\}, \{v_3, v_5\}, \{v_4, v_5\}\}$. A node with the integer $i \in [\![1, 6]\!]$ represents $v_i \in \mathcal{V}$. Note that $|N_{\mathcal{G}}(v_2) \cap N_{\mathcal{G}}(v_4)| = 2$. The set of filled black nodes represents a minimum vertex cover $X = \{v_1, v_3, v_4\}$ of $\mathcal{G}$ of size $|X| = 3$. **(b)** Planar graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ constructed from $\mathcal{G}$ in the proof of Lemma 8. The maximum degree of $\mathcal{G}'$ is three and $|N_{\mathcal{G}'}(v) \cap N_{\mathcal{G}'}(v')| \leq 1$ for all $v, v' \in \mathcal{V}'$, $v \neq v'$. Every edge of $\mathcal{G}$ is replaced by a path composed of four nodes. The set of filled black nodes represents a minimum vertex cover X' of $\mathcal{G}'$ of size $|X'| = |X| + |\mathcal{E}| = 9$.

assume that $X = \{v_1, \ldots, v_k\}$. 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 $N_T(r) = \{u_1, \ldots, u_k\}$ be the set of neighbors of r such that $L_1(u_i) = v_i$ and $L_2(u_i) = i$ for all $i \in [\![1, k]\!]$. For all $i \in [\![1, k]\!]$, let $V^i = \{v_j \in \mathcal{V} \mid \{v_i, v_j\} \in \mathcal{E}, i < j \leq n\}$. For all $i \in [\![1, k]\!]$, let $N_T(u_i) \setminus \{r\} = \{u^i_j \mid v_j \in V^i, i < j \leq n\}$ 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 $u^i_j \in N_T(u_i) \setminus \{r\}$, we set $L_1(u^i_j) = v_j$ and $L_2(u^i_j) = k + 1$. 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 $X = \{v_1, \ldots, v_k\}$ 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 $P_e = (r, u_i, u^i_j)$ represents the edge e, that is $L_1(u_i) = v_i$ and $L_1(u^i_j) = v_j$. 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 $P_e = (r, u_i, u^i_j)$ with $L_1(u_i) = v_i$ and $L_1(u^i_j) = v_j$. 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 neigbhors 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 $N_T(r) = \{u_1, \ldots, u_k\}$. 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 $i \in [\![1, k]\!]$. Thus, for every $e = \{v, v'\} \in \mathcal{E}$, there exists $i \in [\![1, n]\!]$ 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 $X = \{v_1, \ldots, v_k\}$, 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 if the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is planar, $|N_{\mathcal{G}}(v)| \leq 3$, and $|N_{\mathcal{G}}(v) \cap N_{\mathcal{G}}(v')| \leq 1$ for all $v, v' \in \mathcal{V}$, $v \neq v'$. Lemma 8, then the decision variants of the tree representation problems are NP-complete for this class of graphs. $\square$

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 3).

Corollary 3 *Let $\mathcal{G}$ be any graph. There is a linear time algorithm that computes* $\max$ *such that* $2\max \geq \max^* = \max^*_{local} = \max^*_{global}$.

4 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 APX-complete even if $\Delta_{\mathcal{H}} = 3$, and admit a polynomial time k-approximation algorithm when $\Delta_{\mathcal{H}} \leq k$.

4.1 A polynomial time algorithm for hypergraphs of degree two

In Theorem 2, 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 $\{e_1, e'_1\} \cap \{e_2, e'_2\} \neq \emptyset$. Furthermore, $w'_i = w_{e,e'}$ for all $i = \{e, e'\} \in I$. Figures 5(b) and (c) illustrate the previous constructions for the hypergraph depicted in and Figure 5(a).

In order to prove Theorem 2, we define the notion of *independent set* and the associated optimization problem. Given a graph $G = (V, E)$, a set $X \subseteq V$ is an independent set of G if for every $v \in X$, then $N_G(v) \cap X = \emptyset$. The maximum independent set problem consists in computing the maximum k such that there exists an independent set X of G of size $|X| = k$. The *maximum independent set problem* is well known to be hard to approximate. For instance, it is APX-complete in the class of cubic graphs [AK97]. A cubic graph is a 3-regular graph, that is every node has degree 3.

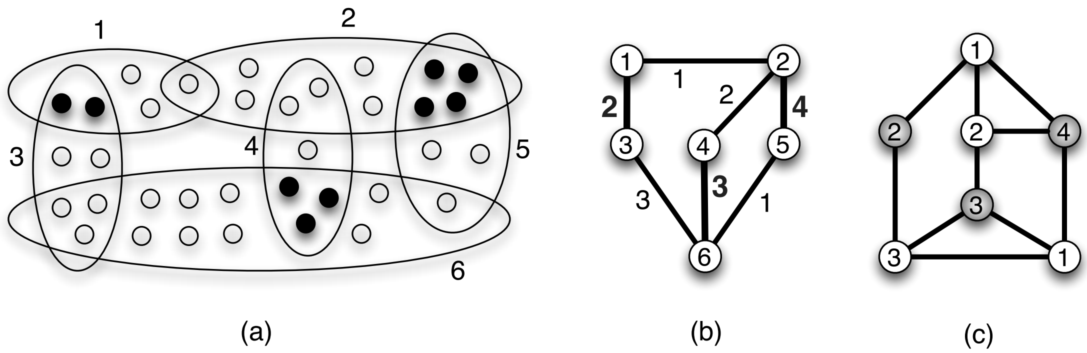


Figure 5: **(a)** Hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ such that $|\mathcal{E}[v]| \leq 2$ for all $v \in \mathcal{V}$. Let $\mathcal{E} = \{e_1, e_2, e_3, e_4, e_5, e_6\}$. An edge with $j \in [\![1, 6]\!]$ represents $e_j \in \mathcal{E}$. The set of filled nodes represents the maximal 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 $j \in [\![1, 6]\!]$ 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 black nodes represents a maximum weighted independent set of L of total weight 9. From Theorem 2, $\max^*_{local} = \max^*_{global} = 9$.

Theorem 2 *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 [Pas08], and so the global tree representation problem is in P. $\square$

4.2 APX-hardness result for hypergraphs of degree three

We prove in Theorem 4 that if the maximum degree of the hypergraph is three, then the global and local tree representation problems are APX-complete. In other words, there is a constant $k > 1$ such that there is no polynomial time k-approximation algorithm for the global and local tree representation problems, unless P = NP. In our reduction, we use a new problem, called *induced induced-star decomposition problem* (*IISD problem*). We formalize this decomposition and the corresponding optimization problem in Definition 4 and Definition 5.

Let $G = (V, E)$ be a graph. Let $k \geq 0$. The graph S induced by the set of nodes $\{v, v_0, \ldots, v_k\} \subseteq V$ is an *induced star* of center v if and only if $\{v, v_i\} \in E$ and $\{v_i, v_j\} \notin E$ for all i, j, $0 \leq i < j \leq k$. Let S, S' be two induced stars of G. The distance between S et S' is

$$d_G(S, S') = \min_{v \in V(S), v' \in V(S')} d_G(v, v'),$$

where $d_G(v, v')$ is the usual distance between two nodes, that is the number of edges of a shortest path between v and v' in G.

Definition 4 (Induced induced-star decomposition) *Given a graph $G = (V, E)$, an induced induced-star decomposition (IISD) of G is a set $\mathcal{S}$ of induced stars such that $d_G(S, S') \geq 2$ for every $S, S' \in \mathcal{S}$, $S \neq S'$.*

Definition 5 (Induced induced-star decomposition problem) *Given a graph $G = (V, E)$, the induced induced-star decomposition problem (IISD problem) consists in computing an IISD $\mathcal{S}$ of G such that $\sum_{S \in \mathcal{S}} 1 + |V(S)| \geq k$.*

Note that $\sum_{S \in \mathcal{S}} 1 + |V(S)| = \sum_{S \in \mathcal{S}} 2 + |E(S)|$. Observe also that if $|V(S)| = 1$ for all $S \in \mathcal{S}$, then $\mathcal{S}$ is an independent set of G, and if $|V(S)| = 2$ for all $S \in \mathcal{S}$, then $\mathcal{S}$ is an induced matching of G. Recall that an independent set of G is a set of nodes that do not share any neighbor. Given a graph $G = (V, E)$, an induced matching of G is a set of edges $F \subseteq E$ such that $|\{e' \in F \mid e' \in N_G(e)\}| \leq 1$ for every edge $e \in E$, where $N_G(e)$ is the set of edges adjacent to e. Informally, an induced matching of G is a matching such that the graph induced by the set of nodes corresponding to this matching, is a matching of G.

We describe in Figure 6(a) a cubic graph G (3-regular graph) for which there exists an IISD $\mathcal{S}$ of G such that $\sum_{S \in \mathcal{S}} 1 + |V(S)| > 2|X|$ for every independent set X of G and such that $\sum_{S \in \mathcal{S}} 1 + |V(S)| > 3|F|$ for every induced matching F of G. In other words, any optimal solution $\mathcal{S}$ for the IISD problem for G is such that $\mathcal{S}$ contains at least one induced star composed of one node and at least one induced star composed of two nodes. Indeed, a maximum independent set of G has size at most $|X| = 7$ (Figure 6(b)), a maximum induced matching of G has size at most $|F| = 5$ (Figure 6(c)), and an optimal solution $\mathcal{S}$ for the IISD problem is such that $|\{S, |V(S)| = 1, S \in \mathcal{S}\}| = 2$ and $|\{S, |V(S)| = 2, S \in \mathcal{S}\} = 4$ (Figure 6(d)). Thus, $\sum_{S \in \mathcal{S}} 1 + |V(S)| = 16 > 2|X| = 14$ and $\sum_{S \in \mathcal{S}} 1 + |V(S)| = 16 > 3|F| = 15$.

We prove in Theorem 3 that the IISD problem is APX-complete in the class of cubic graphs. To do that, we use the fact that the maximum independent set problem is APX-complete in the class of cubic graphs [AK97]. We first show in Lemma 9 that the IISD problem is APX-hard in the class of cubic graphs. We then prove in Lemma 10 that the IISD problem is in APX in that class of graphs.

Lemma 9 *The IISD problem is APX-hard in the class of cubic graphs.*

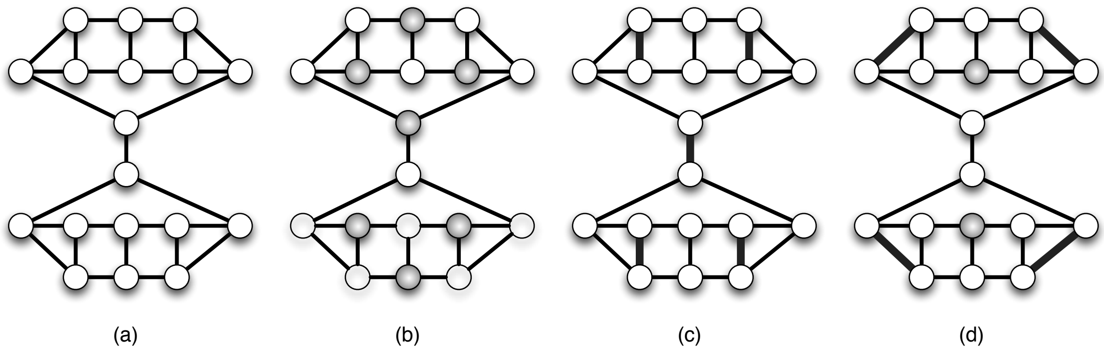


Figure 6: **(a)** Cubic graph $G = (V, E)$. **(b)** Maximum independent set X of G such that $2|X| = 14$. **(c)** Maximum induced matching F of G such that $3|F| = 15$. **(d)** Optimal solution $\mathcal{S}^*$ for the IISD problem for G such that $\sum_{S \in \mathcal{S}^*} 1 + |V(S)| = 16$.

Proof. We prove the result by contradiction. Let us assume that for every $k > 1$, there is a polynomial time k-approximation algorithm for the IISD problem in the class of cubic graphs. Thus, there is a polynomial time algorithm that computes an IISD $\mathcal{S}$ of any cubic graph $G = (V, E)$ such that $k(\sum_{S \in \mathcal{S}} 1 + |V(S)|) \geq \sum_{S \in \mathcal{S}^*} 1 + |V(S)|$, where $\mathcal{S}^*$ is an optimal solution for the IISD problem for G. We have $\sum_{S \in \mathcal{S}^*} 1 + |V(S)| \geq 2|X^*_{MIS}|$ because $\mathcal{S}^*$ is optimal, where X^*_{MIS} is a maximum independent set of G. For every $i \in \{1, 2, 3, 4\}$, let us consider the independent set $X_i \subset V$ of G induced by the centers of all induced stars of $\mathcal{S}$ composed of exactly i node(s). Note that $|X_1 \cup X_2 \cup X_3 \cup X_4| = |\mathcal{S}|$ because G has maximum degree three and because $\mathcal{S}$ is an IISD of G. We get that $k(2|X_1| + 3|X_2| + 4|X_3| + 5|X_4|) \geq 2|X^*_{MIS}|$ because $\sum_{S \in \mathcal{S}} 1 + |V(S)| = 2|X_1| + 3|X_2| + 4|X_3| + 5|X_4|$. Let $X^{app} = X_1 \cup X_2 \cup X_3 \cup X_4$. Note that X^{app} is an independent set of G. We obtain that $5k/2|X^{app}| \geq |X^*_{MIS}|$. Thus, we get a polynomial time $(5k/2)$-approximation for the maximum independent set problem in the class of cubic graphs for every $k > 1$. A contradiction because the maximum independent set problem is APX-hard in the class of cubic graphs [AK97]. Therefore, the IISD problem is APX-hard in the class of cubic graphs. □

Lemma 10 *The IISD problem is in APX in the class of cubic graphs.*

Proof. Let $c > 1$ be a constant such that there is a polynomial time c-approximation algorithm for the maximum independent set problem for the class of cubic graphs. Such a constant c exists because the maximum independent set problem is in APX in the class of cubic graphs [AK97]. In other words, there is a polynomial time algorithm that computes a maximum independent set X^{app}_{MIS} of any cubic graph $G = (V, E)$ such that $c|X^{app}_{MIS}| \geq |X^*_{MIS}|$, where X^*_{MIS} is an optimal solution for the maximum independent set problem for G. Since G has maximum degree three, then every induced star of G is composed of at most four nodes. Thus, $\sum_{S \in \mathcal{S}^*} 1 + |V(S)| \leq 5|X^*_{MIS}|$, where $\mathcal{S}^*$ is an optimal solution for the IISD problem for G. We deduce that for any constant $c' \geq 5c$, then $2c'|X^{app}_{MIS}| \geq 5|X^*_{MIS}| \geq \sum_{S \in \mathcal{S}^*} 1 + |V(S)|$. Since X^{app}_{MIS} is an IISD of G, we conclude that the previous polynomial time c-approximation algorithm for the maximum independent set problem gives a $5c$-approximation algorithm for the IISD problem. □

Lemma 9 and Lemma 10 prove Theorem 3.

Theorem 3 *The IISD problem is APX-complete in the class of cubic graphs.*

We prove in Lemma 11 that any cubic graph can be labeled in a specific way. The properties of this labeling will be useful to prove the main result of this section (Theorem 4).

Lemma 11 *Given a cubic graph $G = (V, E)$, there exist two labeling functions L_V and L_E, $L_V : V \to \mathbb{N}^3$ and $L_E : E \to \mathbb{N}$ such that:*

1. *for every $v \in V$ such that $L_V(v) = \{a_1, a_2, a_3\}$, then $a_i \neq a_j$ for all $1 \leq i < j \leq 3$,*
2. *for every $u, v \in V$ such that $\{u, v\} \notin E$, then $|L_V(u) \cap L_V(v)| = 0$,*
3. *for every $\{u, v\} \in E$, then $|L_V(u) \cap L_V(v)| = 1$ and $L_E(\{u, v\}) = L_V(u) \cap L_V(v)$,*
4. *for every $e, e' \in E$, then $L_E(e) \neq L_E(e')$.*

Proof. Our proof is constructive. Let $n = |V|$ and $V = \{v_1, \ldots, v_n\}$. Set $L_V(v_1) = \{1, 2, 3\}$. Let e_1, e_2, e_3 be the three adjacent edges of v_1. For every $j \in \{1, 2, 3\}$, set $L_E(e_j) = j$. Suppose that the properties are satisfied for the set of nodes $\{v_1, \ldots, v_i\}$ with $1 \leq i \leq n - 1$, and the set of edges that have at least one incident node in $\{v_1, \ldots, v_i\}$. Note that for every edge $e \in E$ that has exactly one incident node in $\{v_1, \ldots, v_i\}$, the constraint (3) can be written as $L_E(e) \in L_V(v)$, where $v \in \{v_1, \ldots, v_i\}$. The result is true for $i = 1$. We prove that we can assign labels to node v_{i+1} and to some of its adjacent edges such that the properties are satisfied for the set of nodes $\{v_1, \ldots, v_{i+1}\}$ and the set of edges that have at least one incident node in $\{v_1, \ldots, v_{i+1}\}$. Let e_1, e_2, e_3 be the three adjacent edges of v_{i+1}. For every $j \in \{1, 2, 3\}$, let t_j be such that: $t_j = L_E(e_j)$ if $L_E(e_j)$ has been defined before, t_j be the j-th smallest integer that does not belong to any node labels if $L_E(e_j)$ has not been defined before. We set $L_V(v_{i+1}) = \{t_1, t_2, t_3\}$, and $L_E(e_j) = t_j$ for every $j \in \{1, 2, 3\}$. By construction and by induction hypothesis, the properties are satisfied. Thus, we get the labeling properties for G. $\square$

From the proof of Lemma 11, we get Corollary 4.

Corollary 4 *Given a cubic graph $G = (V, E)$, we can compute in polynomial time two labeling functions L_V and L_E, $L_V : V \to \mathbb{N}^3$ and $L_E : E \to \mathbb{N}$, that satisfy the properties of Lemma 11.*

In the following, a *labeled cubic graph* is a cubic graph $G = (V, E)$ and labeling functions L_V and L_E that satisfy the properties of Lemma 11. Figure 7 describes a labeled cubic graph.

We now define the notion of *3-intersection graph* of a hypergraph of maximum degree three (Definition 6) and we then prove in Lemma 12 that any cubic graph is the 3-intersection graph of a hypergraph of maximum degree three.

Definition 6 (3-intersection graph) *Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph of maximum degree three. The 3-intersection graph $G = (V, E)$ of $\mathcal{H}$ is defined as follows:*

1. *for every set of hyperedges $\{e_1, e_2, e_3\} \subseteq \mathcal{E}$ such that $e_i \neq e_j$ for $1 \leq i < j \leq 3$ and such that $e_1 \cap e_2 \cap e_3 \neq \emptyset$, then there is a node $u \in V$ that corresponds to $\{e_1, e_2, e_3\}$,*

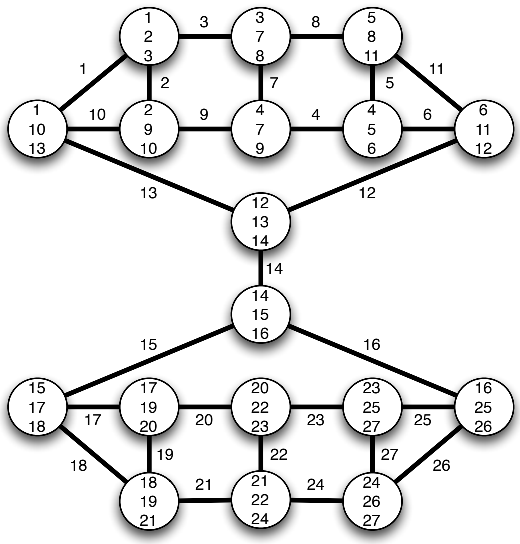


Figure 7: Labeled cubic graph and 3-intersection graph of the hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ composed of $|\mathcal{E}| = 27$ hyperedges, $\mathcal{E} = \{e_1, \ldots, e_{27}\}$. Every integer $i \in [\![1, 27]\!]$ represents e_i. Every set of three integers $\{i, j, k\}$ that is represented in a node of the graph, $1 \leq i < j < k \leq 27$, represents the intersection of the three hyperedges e_i, e_j, e_k. There is a unique node $v \in e_i \cap e_j \cap e_k$, and for every hyperedge $e \in \mathcal{E}$, there is at least one node that belongs only to e.

2. *for every node $u \in V$ that corresponds to the set of hyperedges $\{e_1, e_2, e_3\} \subseteq \mathcal{E}$ and for every node $u' \in V$ that corresponds to the set of hyperedges $\{e'_1, e'_2, e'_3\} \subseteq \mathcal{E}$, then there is an edge $\{u, u'\} \in E$ if and only if $|\{e_1, e_2, e_3\} \cap \{e'_1, e'_2, e'_3\}| \geq 1$.*

Lemma 12 *Every labeled cubic graph $G = (V, E)$ is the 3-intersection graph of a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$.*

Proof. Let $G = (V, E)$ be any labeled cubic graph. We construct a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ as follows. For every node $u \in V$ with $L_V(u) = \{i, j, k\}$, $i, j, k \geq 1$, there are three corresponding hyperedges $e_i, e_j, e_k \in \mathcal{E}$ that share at least one node $v \in \mathcal{V}$, that is $v \in e_i \cap e_j \cap e_k$. Furthermore, for every $i, j, k \geq 1$, if there is no node $u \in V$ such that $L_V(u) = \{i, j, k\}$, then there is no node $v \in \mathcal{V}$ that belongs to $e_i \cap e_j \cap e_k$. From the properties of the labeling of G, then there is an edge $\{u, u'\} \in E$ if and only if there is a hyperedge $e \in \mathcal{E}$ that corresponds to both node u and node u'. Then, for every hyperedge $e \in \mathcal{E}$, we add at least one vertex that only belongs to e. Finally, G satisfies the properties of Definition 6, and so G is the 3-intersection graph of $\mathcal{H}$. $\square$

Figure 7 describes a labeled cubic graph that is a 3-intersection graph of the hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ composed of $|\mathcal{E}| = 27$ hyperedges. Let $\mathcal{E} = \{e_1, \ldots, e_{27}\}$. Every integer $i \in [\![1, 27]\!]$ represents e_i. Every set of three integers $\{i, j, k\}$ that is represented in a node of the graph, $1 \leq i < j < k \leq 27$, represents the intersection of the three hyperedges e_i, e_j, e_k. There is a unique node $v \in e_i \cap e_j \cap e_k$. Furthermore, for every hyperedge $e \in \mathcal{E}$, there is at least one node that belongs only to e.

We are now able to prove Theorem 4.

Theorem 4 *The global and local tree representation problems are APX-complete even if the maximum degree of the hypergraph is three.*

Proof. Let $G = (V, E)$ be any labeled cubic graph. Let $n = |V|$ and $V = \{u_1, \ldots, u_n\}$. From Lemma 12, there is a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ such that G is the 3-intersection graph of $\mathcal{H}$. Let $N = |\mathcal{V}|$ and let $\mathcal{V} = \{v_1, \ldots, v_N\}$. Let $\mathcal{E} = \{e_1, e_2, \ldots\}$. A label i of G corresponds to hyperedge $e_i \in \mathcal{E}$. We assume that for all $e_1, e_2, e_3 \in \mathcal{E}$, if $e_1 \cap e_2 \cap e_3 \neq \emptyset$, then $|e_1 \cap e_2 \cap e_3| = 1$. We assume that the set of nodes of $\mathcal{H}$ that belong to a unique hyperedge is $\{v_{n+1}, \ldots, v_N\}$. Let $k \geq 1$. We prove that $\max^*_{global} \geq k$ for $\mathcal{H}$ if and only if there exists an IISD $\mathcal{S}$ of G such that $\sum_{S \in \mathcal{S}} 1 + |V(S)| \geq k$. We then show that $\max^*_{local} = \max^*_{global}$.

$\Leftarrow$ Suppose there exists an IISD $\mathcal{S}$ of G such that $\sum_{S \in \mathcal{S}} 1 + |V(S)| \geq k$. We prove that $\max^*_{global} \geq k$. By assumption, there is a unique node $v \in \mathcal{V}$ that corresponds to u_i, for every i, $1 \leq i \leq n$, that is only v belongs to $e_j \cap e_k \cap e_l$, where $L_V(u_i) = \{e_j, e_k, e_l\}$. For every i, $1 \leq i \leq n$, such a node is denoted $h(u_i)$. Without loss of generality, let $\{u_1, \ldots, u_{|\mathcal{S}|}\}$ be the centers of the stars of $\mathcal{S}$. Let $\sigma = (h(u_1), \ldots, h(u_{|\mathcal{S}|}), \ldots, h(u_n), v_{n+1}, \ldots, v_N)$ be an ordering for the global tree representation T of $\mathcal{H}$. By definition of a global tree representation and by construction of σ, there is a unique node in T that represents $h(u_i)$ for every i, $1 \leq i \leq |\mathcal{S}|$. Note that $h(u_i)$ belongs to three different hyperedges of $\mathcal{H}$. Furthermore, there are two nodes in T that represent $h(u_j)$ for every j, $|\mathcal{S}| + 1 \leq j \leq \sum_{S \in \mathcal{S}} |V(S)|$. Note that $h(u_j)$ belongs to three different hyperedges of $\mathcal{H}$. We obtain that $\max^*_{global} \geq 2|\mathcal{S}| + (\sum_{S \in \mathcal{S}} |V(S)| - |\mathcal{S}|) = \sum_{S \in \mathcal{S}} 1 + |V(S)| \geq k$.

$\Rightarrow$ Suppose that $\max^*_{global} \geq k$. We prove that there exists an IISD $\mathcal{S}$ of G such that $\sum_{S \in \mathcal{S}} 1 + |V(S)| \geq k$. Without loss of generality, let $\sigma = (v_1, \ldots, v_N)$ be an ordering for the global tree representation T of $\mathcal{H}$ such that $\max^*_{global} \geq k$. Recall that the set of nodes of $\mathcal{H}$ that belong to a unique hyperedge is $\{v_{n+1}, \ldots, v_N\}$. Let $k \geq 1$. Every node v_i, $1 \leq i \leq n$, appears in exactly three different hyperedges. Thus, there are either one, two, or three nodes in T that represent v_i, $1 \leq i \leq n$. Let us define the function g as follows: $g(v_i) = 2$ if there is one node in T that represents v_i, $g(v_i) = 1$ if there are two nodes in T that represent v_i, and $g(v_i) = 0$ if there are three nodes in T that represent v_i. By assumption, $\sum_{i=1}^{n} g(v_i) \geq k$. Let $u(v_i)$ be the node of G such that the three corresponding hyperedges of the labeling for $u(v_i)$ contain node v_i. We first prove that $\{u(v_i), g(v_i) \geq 1, i = 1 \ldots n\}$ forms an IISD $\mathcal{S}$ of G. Without loss of generality, assume that $g(v_i) \geq g(v_j)$ for all i, j, $1 \leq i < j \leq n$. We prove the result by induction. Let us assmume that $\{u(v_i), g(v_i) \geq 1, i = 1 \ldots t\}$ forms an IISD of G for any t, $1 \leq t \leq n - 1$. It is clearly true for $t = 1$. We prove that it is also true for $t + 1$, that is $\{u(v_i), g(v_i) \geq 1, i = 1 \ldots t + 1\}$ forms an IISD of G. Consider the node $u(v_{t+1})$.

- If $g(v_{t+1}) = 2$, then $d_G(u(v_{t+1}), u(v_j)) \geq 2$ for all j, $1 \leq j \leq t$. Indeed, otherwise, we would have $g(v_{t+1}) \leq 1$ if $d_G(u(v_{t+1}), u(v_j)) = 1$ for some j, $1 \leq j \leq t$. Recall that in that case $u(v_{t+1})$ and $u(v_j)$ would have a common integer label and so a common hyperedge. Thus, $\{u(v_i), g(v_i) \geq 1, i = 1 \ldots t + 1\}$ forms an IISD of G.

- If $g(v_{t+1}) = 1$, then $u(v_{t+1})$ has a unique neighbor $u(v_j)$ such that $g(v_j) \geq 1$ for some j, $1 \leq j \leq t$. More precisely, $g(v_j) = 2$. Indeed, otherwise, we would have $g(v_{t+1}) = 2$. Furthermore, every node $u(v_{j'}) \notin N_G(u(v_j))$ such that $g(v_{j'}) \geq 1$ is such that $d_G(u(v_{t+1}), u(v_{j'})) \geq 2$. Indeed, otherwise, we would have $g(v_{t+1}) = 0$ because $g(v_j) = 2$ and $g(v_{j'}) \geq 1$. In that case, the node v_j would be represented three times in T: one time when representing the hyperedge that $u(v_t)$ and $u(v_j)$ share; one time when representing the hyperedge that $u(v_{j'})$ and $u(v_j)$

share; and one other time when representing the other (and different) hyperedge that contains $u(v_j)$. Thus, $\{u(v_i), g(v_i) \geq 1, i = 1 \ldots t+1\}$ forms an IISD of G.

- If $g(v_{t+1}) = 0$, then it is done, that is $\{u(v_i), g(v_i) \geq 1, i = 1 \ldots t+1\}$ forms an IISD of G.

We now prove that $\{u(v_i), g(v_i) \geq 1, i = 1 \ldots n\}$ forms an IISD $\mathcal{S}$ of G such that $\sum_{S \in \mathcal{S}} 1 + |V(S)| \geq k$. By the previous induction proof, we observe that there is a unique node $u \in S$ such that $g(u) = 2$ for every $S \in \mathcal{S}$. In other words, $|\{u, u \in S, g(u) = 2\}| = 1$ for every $S \in \mathcal{S}$. If $g(v_i) = 2$, we set that $u(v_i)$ is the center of its star for all i, $1 \leq i \leq n$. Otherwise, $u(v_i)$ is a leaf of a star (if $g(v_i) = 1$) or it is not in a star (if $g(v_i) = 0$). Thus, we get that $\sum_{S \in \mathcal{S}} 1 + |V(S)| \geq k$. $\square$

4.3 Constant factor approximations

In this section, we prove 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. To do that, we transform the instances of global and local tree representation problems into instances of the *weighted set packing problem*. 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 [GJ90, Hoc83]. We formalize in Definition 7 the auxiliary instance of the weighted set packing problem constructed from a hypergraph.

Definition 7 (instance of weighted set packing problem from a hypergraph) *Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph with $\mathcal{E} = \{e_1, \ldots, e_m\}$. The instance (U, S) of the weighted set packing problem from $\mathcal{H}$ is defined as follows. Set $U = \mathcal{E}$. Let $e_{s_1}, \ldots, e_{s_t} \in \mathcal{E}$ be any t hyperedges, with $2 \leq t \leq m$. The subset $\{e_{s_1}, \ldots, e_{s_t}\} \in S$ if and only if $e_{s_1} \cap \ldots \cap e_{s_t} \neq \emptyset$. Furthermore, $w_s = |e_{s_1} \cap e_{s_2}|$ if $t = 2$ and $w_s = (t-1)|e_{s_1} \cap \ldots \cap e_{s_t}| + (t-2)(\max_{e \in s}(w_{\{e_{s_1}, \ldots, e_{s_t}\} \setminus \{e\}}) - |e_{s_1} \cap \ldots \cap e_{s_t}|)$ 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 8 depicts an example of auxiliary instance and illustrates Theorem 5.

Theorem 5 *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.*

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 $C = \{c_1, \ldots, c_q\}$ 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 $c_i = \{e_i^1, \ldots, e_i^{t_i}\}$. Without loss of generality, assume that $w_{e_i^1, \ldots, e_i^j} = (j-1)|e_i^1 \cap \ldots \cap e_i^j| + (j-2)(w_{e_i^1 \cap \ldots \cap e_i^{j-1}} - |e_i^1 \cap \ldots \cap e_i^j|)$ 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 $Y_i^j = (e_i^1 \cap \ldots \cap e_i^j) \setminus (e_i^{j+1} \cup \ldots \cup e_i^{t_i})$.

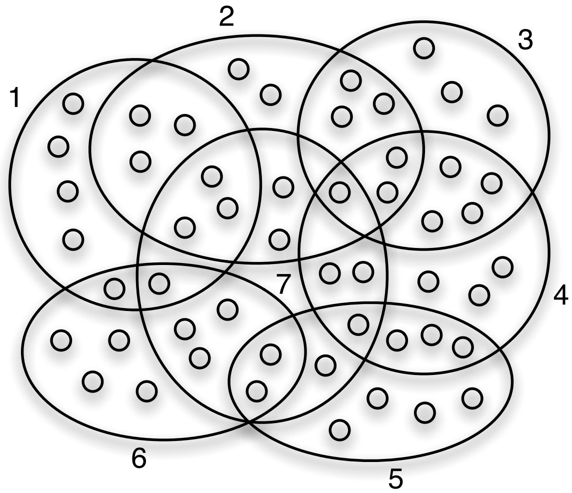


Figure 8: Hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{E} = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$. A hyperedge with $i \in [\![1, 7]\!]$ represents $e_i \in \mathcal{E}$. The auxiliary instance (U, S) of the weighted set packing problem from $\mathcal{H}$ is such that $U = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$ 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 5, we get that $\max^*_{local} = \max^*_{global} = 21$.

Let us construct $T_{local} = (V, E, L_1, L_2)$ rooted at $r \in V$ as follows. For all $i = 1, \ldots, q$, for all $j = t_i, \ldots, 1$, we add $|Y_i^j|$ nodes in T, namely $u_i^{j,x}$, where $x = 1, \ldots, |Y_i^j|$. For all $i = 1, \ldots, q$, for all $j = t_i, \ldots, 1$, for all $x = 1, \ldots, |Y_i^j|$, for all $x' = x + 1, \ldots, |Y_i^j|$, 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 $i = 1, \ldots, q$, for all $j = t_i, \ldots, 1$, for all $x = 1, \ldots, |Y_i^j| - 1$, we add an edge $\{u_i^{j,x}, u_i^{j,x+1}\} \in E(T_{local})$. For all $i = 1, \ldots, q$, we add an edge $\{r, u_i^{t_i,1}\} \in E(T_{local})$. For all $i = 1, \ldots, q$, for all $j = t_i, \ldots, 2$, 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 $i = 1, \ldots, q$, for all $j = t_i, \ldots, 1$, let $\{u_i'^{j,1}, \ldots, u_i'^{j,t_i}\} = (\mathcal{V} \cap e_i^j) \setminus Y$. For all $i = 1, \ldots, q$, for all $j = t_i, \ldots, 1$, 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 $i = 1, \ldots, q$, for all $j = t_i, \ldots, 1$, we add in T the path $(u_{i,j}, u_i'^{j,1}, \ldots, u_i'^{j,t_i})$; 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 $P_e = (r, u_1, \ldots, u_{|e|})$ 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 $C = \{c_1, \ldots, c_q\}$ 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 $N_{T_{local}}(r) = \{u_1, \ldots, u_q\}$, where q is the degree of the root r. We construct the packing $C = \{c_1, \ldots, c_q\}$ of S as follows. For all i, $1 \leq i \leq q$, let $c_i = \{e_i^1, \ldots, e_i^{t_i}\}$ such that $L_1(u_i) \in e_i^j$ and $L_1(u_i) \notin e$ 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 neigbhors of r, that is the q subtrees $T^1_{local}, \ldots, T^q_{local}$ rooted at $u_1, \ldots, u_q$, respectively. By induction hypothesis and by the previous remark, there exists a packing C_i such that $\sum_{c \in C_i} w_c \geq x^i_{local}$ for all i, $1 \leq i \leq q$. Thus, the packing C is such that $\sum_{c \in C} w_c \geq k = \sum_{i=1}^{q} x^i_{local}$.

The trees previously described satisfy the properties of Definition 2 and the properties of Definition 3. Thus, $\max^*_{local} = \max^*_{global}$.

Finally, the $\frac{k}{2}$-approximation algorithm for the weighted set packing problem proved in [GJ90], gives a $\frac{k}{2}$-approximation algorithm for our problems. $\square$

5 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

References

[AK97] Paola Alimonti and Viggo Kann. Hardness of approximating problems on cubic graphs. In Giancarlo Bongiovanni, DanielPierre Bovet, and Giuseppe Di Battista, editors, *Algorithms and Complexity*, volume 1203 of *Lecture Notes in Computer Science*, pages 288–298. Springer Berlin Heidelberg, 1997.

[BCST15] Jean-Daniel Boissonnat, Karthik C. S., and Sébastien Tavenas. Building Efficient and Compact Data Structures for Simplicial Complexes. In *International Symposium on Computational Geometry 2015*, Eindhoven, Netherlands, June 2015.

[BM14] J.-D. Boissonnat and C. Maria. The simplex tree: An efficient data structure for general simplicial complexes. *Algorithmica*, 70(3):406–427, 2014.

[GJ77] 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.

[GJ90] 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.

[GJS74] 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.

[Hoc83] D. S. Hochbaum. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Applied Mathematics*, 6(3):243 – 254, 1983.

[Pas08] Vangelis Th. Paschos. *Combinatorial Optimization and Theoretical Computer Science - Interfaces and Perspectives: 30th Anniversary of the LAMSADE.* Wiley, 2008.