Theoretical Computer Science 475 (2013) 47–58

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

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

Factorization of products of hypergraphs: Structure and algorithms☆

Alain Bretto[a,*], Yannick Silvestre[a], Thierry Vallée[b]

[a] Normandie Univ. GREYC CNRS UMR 6072, Campus II, Bd Maréchal Juin BP 5186, 4032 Caen cedex, France

[b] Université Paris Diderot - Paris 7, Laboratoire PPS, Case 7014, 75205 Paris Cedex 13, France

ARTICLE INFO

Article history:
Received 23 May 2011
Received in revised form 23 December 2012
Accepted 26 December 2012
Communicated by G. Ausiello

Keywords:
Hypergraph
Cartesian product
Factorization algorithm

ABSTRACT

On the one hand Cartesian products of graphs have been extensively studied since the 1960s. On the other hand hypergraphs are a well-known and useful generalization of graphs.

In this article, we present an algorithm able to factorize into its prime factors any bounded-rank and bounded-degree hypergraph in $O(nm)$, where n is the number of vertices and m is the number of hyperedges of the hypergraph.

First the algorithm applies a graph factorization algorithm to the 2-section of the hypergraph. Then the 2-section factorization is used to build the factorization of the hypergraph via the factorization of its L2-section. The L2-section is a recently introduced way to interpret a hypergraph as a labeled-graph.

The graph factorization algorithm used in this article is due to Imrich and Peterin and is linear in time and space. Nevertheless any other such algorithm could be extended to a hypergraph factorization algorithm similar to the one presented here.

1. Introduction

In the 1960s Vizing and Sabidussi independently showed [17,19] that, for every finite connected graph, there is a unique (up to isomorphism) decomposition of the graph into prime factors. This theorem was the starting point for research on Cartesian products of graphs. Some of the questions raised during these years are still open, as for Vizing's conjecture.[1]

An important motivation for the study of Cartesian products is that factorization allows us to reduce algorithmic complexity by transferring the search for solutions from the product to the factors. Several classical problems in graph theory were made easier following this approach. For instance, it is well-known that the chromatic number of a Cartesian product is the maximum of the chromatic numbers of its factors [16] and that lower and upper bounds for the independence number of a product can be given using the independence numbers of its factors [19,13]. Several other useful parameters or properties of graphs were also investigated, especially in coloring theory. For instance, several interesting results concerning the antimagicness [21,9,20] as well as the game chromatic number [15] of various classes of Cartesian products were recently published. Thus, all these parameters and properties are easily computable thanks to Cartesian product operations.

Moreover, most of the networks used in the context of parallel and distributed computation are Cartesian products: the hypercube, grid graphs, etc. In this context, the problem of finding a "Cartesian" embedding of an interconnection network

☆ A short version of this paper has been presented to COCOON 2010.

* Corresponding author. Tel.: +33 2315467485.
E-mail addresses: alain.bretto@unicaen.fr (A. Bretto), yannick.silvestre@unicaen.fr (Y. Silvestre), vallee@pps.jussieu.fr (T. Vallée).

1 This conjecture expressed by Vizing in 1968 states that the domination number of the Cartesian product of graphs is greater than the product of the domination numbers of its factors.

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2012.12.050

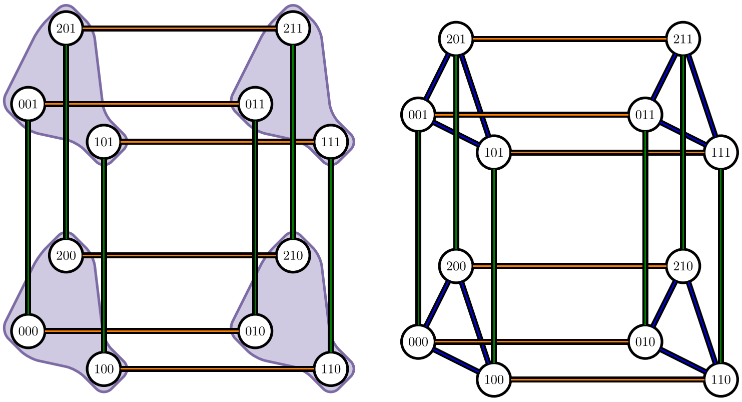

Fig. 1. A graph H and its 2-section $[H]_2$. The 2-section is the Cartesian product $K_3\Box K_2\Box K_2$ while H is the Cartesian product $T\Box K_2\Box K_2$, where T is the hypergraph $T = (\{0, 1, 2\}, \{\{0, 1, 2\}\})$.

into another one is also a topic of interest and thus has gained considerable attention (see for instance [1,10]). Note that Cartesian products are also used in telecommunications [18].

Finally, in 2006, Imrich and Peterin [12] gave an algorithm able to compute the prime factorization of connected graphs in linear time and space, making the use of Cartesian products even more attractive.

Hypergraphs are a well-known generalization of graphs introduced in the 1960s [3]. Since then, they had many applications in several fields of computer science: machine learning, game theory, indexing of databases, SAT problem, data mining and optimization (for a survey see [4]).

Cartesian products of hypergraphs can be defined in the same way as for graphs. The unicity of the prime factorization was first proved for finite hypergraphs in [11] and then generalized to infinite hypergraphs in [14]. As for graphs, it is often possible to facilitate the search for solutions by studying the factors rather than the product. In particular, several hypergraph properties and parameters (see [5,7]), like linearity, conformality, Helly property, transversal and matching numbers, can be easily deduced from the same properties and parameters on the factors. For instance, a hypergraph has the Helly property if its factors have it.

Summary of the results

In this article, we present an algorithm (Algorithm 1) able to factorize any hypergraph into its prime factors. It is, up to our knowledge, the first such algorithm. It is based on the algorithm of Imrich and Peterin introduced in [12], but it is easily adaptable to any algorithm which factorizes Cartesian products of graphs.

One way to interpret the greater generality of hypergraphs over graphs is to say, for fixed parameters, that a hypergraph can store more information than a graph. In [8], this interpretation is made explicit by the introduction of some sort of labeled-graphs where the labels are used to store the additional information.

In the sequel, we use an alternative and equivalent way, introduced first in [7,6], to represent hypergraphs by labeled-graphs. These labeled-graphs are named *labeled 2-sections (L2-sections)* as they interpret a hypergraph H by its *2-section* G endowed with an additional labeling function $\mathcal{L}$, which associates with each edge of G the set of all hyperedges of H containing the vertices of this edge. It is then easy to retrieve the hypergraph H from its L2-section $(G, \mathcal{L})$. It is also straightforward to show that the prime factorization of H can be easily built from the prime factorization of $(G, \mathcal{L})$. Note that this is not true for the 2-section. Indeed, except in the quite narrow case of conformal hypergraphs, the 2-section does not even contain enough information to decide the number of factors in the hypergraph prime factorization. For instance, Figs. 1 and 2 give an example of two hypergraphs which have the same 2-section and have respectively three and two prime factors. It is also not so difficult to see that there exists another hypergraph which has again the same 2-section and which is prime.

The basic idea behind the design of Algorithm 1 is to use the L2-section $(G, \mathcal{L})$ of H. In particular, the algorithm runs the algorithm of Imrich and Peterin on the 2-section G to obtain its prime factors $G_1, \dots, G_k$. Then, it is not so difficult to see that the prime factors of $(G, \mathcal{L})$ must be of the form $(G_{c_1}, \mathcal{L}_1), \dots, (G_{c_m}, \mathcal{L}_m)$, where $c_1, \dots, c_m$ is a partition of $\bar{k} = \{1, \dots, k\}$ and where each G_{c_j} is the Cartesian product of the G_i's, for $i \in c_j$, and $\mathcal{L}_j$ is some labeling-function on G_{c_j}. So it remains to find the partition $c_1, \dots, c_m$ as well as the $\mathcal{L}_1, \dots, \mathcal{L}_m$ functions. Since by definition a factor contains at least two vertices, G has at most $\log_2 n$ factors, where n is the number of vertices of G, and so the set $\bar{k}$ has at most $2^{\log_2 n} = n$ subsets. Hence, it is feasible to find the partition by trying every possible subset $c \subseteq \bar{k}$. Nevertheless, this way does not give any information about how to define the $\mathcal{L}_j$'s. Indeed, except in the case where H is conformal, there are many possibilities to define $\mathcal{L}_j$ in such a way that $(G_{c_j}, \mathcal{L}_j)$ is an L2-section, that is, there are many possibilities to define a hypergraph corresponding to G_{c_j}.

Hence, in the sequel we define explicitly the right partition $c_1, \dots, c_m$ (cf. Definition 9) in such a way that the $\mathcal{L}_j$'s can be defined, up to isomorphism, as the restrictions of $\mathcal{L}$ to some subgraphs of G, called the *c_j-layers* of G.

In the second section of this article, we show how to build the L2-section of a hypergraph and conversely how to retrieve a hypergraph from its L2-section. Then we introduce Cartesian products of hypergraphs as well as Cartesian products of

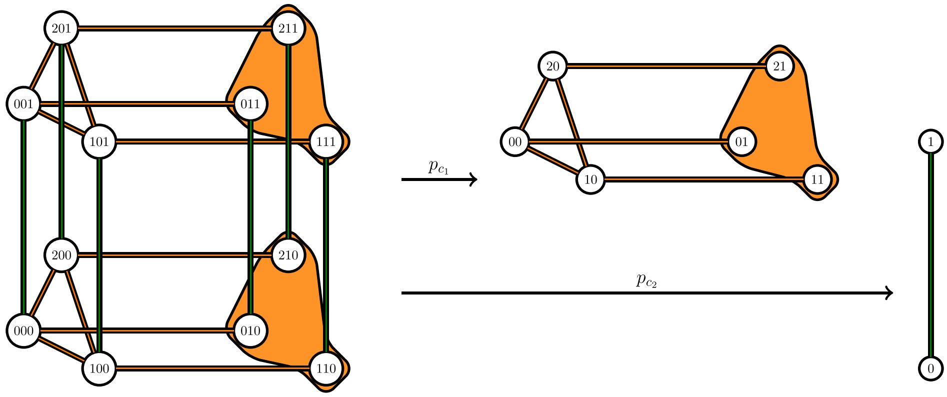

Fig. 2. A hypergraph $\mathscr{H}$ with two factors corresponding to two equivalence classes $c_1 = \{1, 2\}$ and $c_2 = \{3\}$ according to Definition 9.

L2-sections. In the fourth section, we introduce the graph factorization algorithm of Imrich and Peterin and give some properties of Cartesian products. This algorithm is based on a *coloring* of the Cartesian product which exhibits the underlying factorization of the graph. In the fifth section, we introduce an equivalence relation on the set of colors induced by the factorization of the 2-section of the hypergraph. This relation induced a new coloring of the 2-section and is shown to induce the prime factorization of the hypergraph. Finally, in the sixth section, based on these results, we introduce an algorithm able to perform the prime factorization of any bounded-rank and bounded-degree hypergraph in $O(nm)$, where n is the number of vertices and m is the number of hyperedges of the hypergraph.

Preliminaries

In the sequel, the cardinality of a set A is denoted by $|A|$. The set $\mathscr{P}_2(A)$ is the set of pairs $\{x, y\}$ such that $x, y \in A$ and $x \neq y$, $\mathscr{P}(A)$ is the powerset of A and $\mathscr{P}^*(A) = \{a \in \mathscr{P}(A) : a \neq \emptyset\}$. The union of A is the set $\bigcup A = \{x \in a : a \in A\}$. For f a function, we define $\mathrm{Im}(f) = \{y : \exists x, f(x) = y\}$ and for every subset A of its domain $f[A] = \{f(a) : a \in A\}$.

The general terminology concerning graphs and hypergraphs in this article is similar to the one used in [2,3].

A *hypergraph* H on a non-empty set of vertices V is a pair (V, E), where E is a set of non-empty subsets of V, called *hyperedges*. A hypergraph is *simple* if no hyperedge is contained in another. In the sequel, unless explicitly stated, we suppose hypergraphs to be simple and that no hyperedge is a loop, that is, the cardinality of a hyperedge is at least 2. Hypergraphs are considered non-trivial here, that is, to contain at least two vertices and one hyperedge.

A *graph* $G = (V, E)$ is a particular case of (simple) hypergraph where every $e \in E$ is of size 2. Hyperedges of graphs are simply called *edges*. The set of vertices (resp. hyperedges) of the hypergraph H is often written $V(H)$ (resp. $E(H)$).

A path p in H is either a single vertex x, or a sequence of vertices $(x_0, \ldots, x_n)$, where $n \geq 1$, containing no repetition of vertex and such that x_i, x_{i+1}, belong to an hyperedge of H, for every $i \in \{0, \ldots, n-1\}$. The integer $n = \mathit{length}(p)$ is the *length* of p and the vertices x_0 and x_n are said to be *connected* by p in H. In particular, if $p = x$ then $\mathit{length}(p) = 0$ and it is convenient to consider that p connects x to itself.

The hypergraph H is *connected* if every pair $\{x, y\} \subseteq V(H)$ is connected by a path. From now on, unless explicitly stated, we assume that hypergraphs are connected. Note that this implies that every vertex is incident to at least one edge, that is, there is no isolated vertex and $V = \bigcup E$.

Two subgraphs of a graph G are *vertex-disjoint* if they have no common vertex. Two vertex-disjoint subgraphs G_1, G_2 are *adjacent* if there exists an edge of G having one vertex in G_1 and the other in G_2. They are *connected* by a path p if the first vertex of p is in G_1 and its last vertex is in G_2. The distance between G_1 and G_2 in G is the minimal length of a path connecting G_1 and G_2. Moreover, it is convenient to consider that every subgraph G' of G is connected to itself by single-vertex paths of the form $p = x$, where x is a vertex of G', and so that the distance of G' to itself is 0.

The subgraph of $G = (V, E)$ *induced* by a set of vertices $V' \subseteq V$ is the graph (V', E'), where $E' = \{\{u, v\} \in E : \{u, v\} \subseteq V'\}$.

The number of hyperedges of a hypergraph H is denoted by $m(H)$. The *rank* of H is $r(H) = \max\{|e| : e \in E(H)\}$.

Given two simple hypergraphs $H_1 = (V_1, E_1)$, $H_2 = (V_2, E_2)$, a hypergraph *isomorphism* is a bijection $f : V_1 \to V_2$ such that, for all $e \subseteq V_1$, we have $e \in E_1$ if and only if $f[e] \in E_2$.

2. Hypergraphs and labeled 2-sections

We introduce below the notion of *labeled 2-section (L2-section)* of a hypergraph which is a particular case of labeled-graph.

2.1. Definitions and basic facts

The *2-section* of the hypergraph $H = (V, E)$ is the graph $[H]_2 = (V, \mathbb{E})$ where $\mathbb{E} = \bigcup_{e \in E} \mathscr{P}_2(e)$, that is, two distinct vertices are adjacent in $[H]_2$ if and only if they belong to a same hyperedge of H (see Fig. 1). Note that every hyperedge of H

is a clique of $[H]_2$ and that $\bigcup E = V$ implies $\bigcup \mathbb{E} = V$. Note also that a hypergraph is connected if and only if its 2-section is. Note finally that if f is an isomorphism between $[H]_2$ and a graph G then $H' = (f[V], \{f[e] : e \in E\})$ is also a hypergraph isomorphic to H. This remark will allow us to work up to isomorphism when dealing with isomorphisms between $[H]_2$ and Cartesian products of graphs.

In order to define the labeled 2-section, we introduce first a slightly more general concept of labeled-graph, called *hyperedge-set labeled-graph*. Note that the hypergraph $H = (V, E)$ associated with a hyperedge-set labeled-graph may not be simple, nor connected (so we may have $\bigcup E \subsetneq V$).

Definition 1. A *(hyperedge-set) labeled-graph* is a pair $\Gamma = (G, \mathcal{L})$, where $G = (V, E')$ is a graph and $\mathcal{L}$ is a function from E' to $\mathcal{P}^*(\mathcal{P}^*(V))$.
The *inverse* $[\Gamma]_{L2}^{-1}$ of the labeled-graph $\Gamma = (G, \mathcal{L})$ is the hypergraph $H = (V, \bigcup \mathrm{Im}(\mathcal{L}))$, where V is the set of vertices of G.

Definition 2 (*L2-Section*)**.** For every hypergraph $H = (V, E)$, the L2-*section of* H is the labeled-graph $[H]_{L2} = ([H]_2, \mathcal{L})$, where the function $\mathcal{L}$ is defined, for every edge $\{x, y\}$ of $[H]_2$, by $\mathcal{L}(\{x, y\}) = \{e \in E : x, y \in e\}$.

A labeled-graph Γ is a *labeled 2-section (L2-section)* if there exists a hypergraph H such that $[H]_{L2} = \Gamma$.

The following result is straightforward.

Proposition 1. *For all hypergraph H and L2-section Γ we have $[[H]_{L2}]_{L2}^{-1} = H$ and $[[\Gamma]_{L2}^{-1}]_{L2} = \Gamma$.*

Definition 3. An isomorphism between two labeled-graphs $\Gamma_1 = (G_1, \mathcal{L}_1)$ and $\Gamma_2 = (G_2, \mathcal{L}_2)$, where $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, is a bijection $f : V_1 \to V_2$ such that

1. f is a graph isomorphism from G_1 to G_2.
2. $\mathcal{L}_2(\{f(x), f(y)\}) = \{f[e] : e \in \mathcal{L}_1(\{x, y\})\}$, for all $\{x, y\} \in E_1$.

We write $\Gamma_1 \cong \Gamma_2$ and we say that Γ_1 and Γ_2 are L2-isomorphic if there exists an isomorphism from Γ_1 and Γ_2.

Note that by the first condition of the definition, f is a graph isomorphism from G_1 to G_2 and so $\{x, y\} \in E_1$ if and only if $\{f(x), f(y)\} \in E_2$, for every pair of vertices of V_1. This ensures that $\mathcal{L}_1(\{x, y\})$ is defined if and only if $\mathcal{L}_2(\{f(x), f(y)\})$ is, and so that the second condition of the definition makes sense.

It is also easy to check that if $f : \Gamma_1 \to \Gamma_2$ and $g : \Gamma_2 \to \Gamma_3$ are labeled-graph isomorphisms then $g \circ f : \Gamma_1 \to \Gamma_3$ is also an isomorphism. It is also clear that if Γ_1 and Γ_2 are isomorphic then Γ_1 is an L2-section if and only if Γ_2 is. The following result is straightforward.

Lemma 1. *Two hypergraphs are isomorphic if and only if their L2-sections are.*

We remind the reader that a hypergraph $H' = (V', E')$ is a *partial hypergraph* of $H = (V, E)$ if $E' \subseteq E$ and $V' = \bigcup E'$. We give below a sufficient condition for a labeled-graph Γ_0 to be a sub-section of Γ, that is, to be a labeled-graph such that $[\Gamma_0]_{L2}^{-1}$ is a partial hypergraph of H.

Definition 4 (*Subsection*)**.** Let $\Gamma = (G, \mathcal{L})$ be the L2-section of H, where $G = (V, E')$. A pair $\Gamma_0 = (G_0, \mathcal{L}_0)$ is a *subsection* of Γ if the following conditions are satisfied:

1. $G_0 = (V_0, E'_0)$ is a connected subgraph of G (that is, $V_0 \subseteq V$ and $E'_0 \subseteq E'$).
2. $\mathcal{L}_0$ is the restriction of $\mathcal{L}$ to E'_0.
3. If $e \in \bigcup \mathrm{Im}(\mathcal{L}_0)$ then $\mathcal{P}_2(e) \subseteq E'_0$.

Lemma 2. *Let Γ be the L2-section of the simple connected hypergraph H and $\Gamma_0 = (G_0, \mathcal{L}_0)$ be a subsection of Γ. Then $H_0 = [\Gamma_0]_{L2}^{-1}$ is a simple connected partial hypergraph of H with L2-section Γ_0.*

Proof. Let $H = (V, E)$, $\Gamma = ([H]_2, \mathcal{L})$, where $[H]_2 = (V, \mathbb{E})$, $\Gamma_0 = (G_0, \mathcal{L}_0)$, where $G_0 = (V_0, E'_0)$.

We show first that, under the hypotheses of the lemma, Γ_0 is a labeled-graph. Note that G_0 is a connected graph by the first condition of Definition 4, and so $V_0 = \bigcup E'_0$. It remains to show that $\mathcal{L}_0$ is a function from E'_0 to $\mathcal{P}^*(\mathcal{P}^*(V_0))$. Since, $\mathcal{L}_0$ is the restriction of $\mathcal{L}$ to E'_0, we know already that $\mathcal{L}_0$ is defined from E'_0 to $\mathcal{P}^*(\mathcal{P}^*(V))$. So it is sufficient to show that, for every $e \in \bigcup \mathrm{Im}(\mathcal{L}_0)$, e is a non-empty subset of V_0. The non-emptiness is immediate since $e \in \mathcal{P}^*(V)$. Note now that, since $|e| \geq 2$, $e = \bigcup \mathcal{P}_2(e)$. Hence, since $\mathcal{P}_2(e) \subseteq E'_0$ by the third condition of Definition 4, it comes $e = \bigcup \mathcal{P}_2(e) \subseteq \bigcup E'_0 = V_0$.

We show now that the inverse of Γ_0 is a simple connected partial hypergraph of H. So let $H_0 = (V_0, E_0)$, where $E_0 = \bigcup \mathrm{Im}(\mathcal{L}_0)$. Note first that, since Γ_0 is a labeled-graph, H_0 is a hypergraph and $\bigcup E_0 \subseteq V_0$. Since $\mathcal{L}_0$ is the restriction of $\mathcal{L}$ to E'_0, every hyperedge of H_0 comes from H, and so clearly H_0 is simple by simplicity of H. We show now that H_0 is connected. Let u, v be two vertices of V_0. They are connected by a path in G_0. Now, for each edge $\{z, w\}$ of this path, $\mathcal{L}(\{z, w\})$ contains at least one hyperedge which contains z, w by construction of Γ from H. Since $\mathcal{L}_0$ is the restriction of $\mathcal{L}$ to E'_0 and $E_0 = \bigcup \mathrm{Im}(\mathcal{L}_0)$, this hyperedge is in H_0. It is then easy to build a path in H_0 from u to v using these hyperedges. Hence H_0 is connected and so $\cup E_0 = V_0$, and since clearly $E_0 \subseteq E$, H_0 is a partial hypergraph of H. $\square$

3. Cartesian products of hypergraphs and L2-sections

We remind the reader that $\bar{k} = \{1, \ldots, k\}$, for every positive integer k.

Definition 5. Let $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$ be hypergraphs. The *Cartesian product* of H_1 and H_2 is the hypergraph $H_1 \Box H_2$ with set of vertices $V_1 \times V_2$ and set of edges:

$$E_1 \Box E_2 = \underbrace{\{\{x\} \times e : x \in V_1 \text{ and } e \in E_2\}}_{A_1} \cup \underbrace{\{e \times \{u\} : e \in E_1 \text{ and } u \in V_2\}}_{A_2}.$$

Note that up to the isomorphism the Cartesian product is commutative and associative. That will allow us to denote simply by $u = x_1, \ldots, x_k$ the vertices of $V_1 \times \cdots \times V_k$. In particular, every permutation $\pi : \bar{k} \mapsto \bar{k}$ induces an isomorphism f_π between $H_1 \Box \cdots \Box H_k$ and $H_{\pi(1)} \Box \cdots \Box H_{\pi(k)}$ defined by $f_\pi(x_1, \ldots, x_k) = x_{\pi(1)}, \ldots, x_{\pi(k)}$.

For every $i \in \bar{k}$, the i^{th} projection $p_i : V_1 \times \cdots \times V_k \to V_i$ is the function which associates with every k-uple u its i^{th}-coordinate. To simplify notations $p_i(u)$ will be denoted by u_i as soon as there will be no ambiguity. We denote by $u[i := y]$ the vertex of $V_1 \times \cdots \times V_k$ having the same coordinates than u except that u_i is replaced by $y \in V_i$, that is, if $u = x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_k$ then $u[i := y] = x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_k$.

It is easy to check that $\varepsilon \subseteq V_1 \times \cdots \times V_k$ is a hyperedge of $H_1 \Box \cdots \Box H_k$ if and only if there exist a unique $i \in \bar{k}$, $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k \in V_1 \times \cdots \times V_{i-1} \times V_{i+1} \times \cdots \times V_k$ and $e \in E_i$ such that $\varepsilon = \{x_1\} \times \cdots \times \{x_{i-1}\} \times e \times \{x_{i+1}\} \times \cdots \times \{x_k\}$. Such a hyperedge ε is then called an i-hyperedge and clearly $\varepsilon = \{u[i := y] : y \in e\}$, for every $u \in \varepsilon$.

Note that a Cartesian product of hypergraphs is a graph (resp. connected) if and only if all its factors are. Moreover, if H_i is isomorphic to H'_i, for every $i \in \bar{k}$, then $H_1 \Box \cdots \Box H_k$ is also isomorphic to $H'_1 \Box \cdots \Box H'_k$.

Fig. 1 illustrates the notion of Cartesian product of graphs and hypergraphs. We give now two results from [5].

Lemma 3. *We have $A_1 \cap A_2 = \emptyset$. Moreover, $|e \cap e'| \leq 1$ for any $e \in A_1$ and any $e' \in A_2$.*

Proposition 2. *If H_1 and H_2 are hypergraphs then the 2-section of their Cartesian product is the Cartesian product of their 2-sections.*

The Cartesian product is now extended to (hyperedge-set) labeled-graphs.

Definition 6. Let $\Gamma_1 = (G_1, \mathcal{L}_1)$ and $\Gamma_2 = (G_2, \mathcal{L}_2)$ be two labeled-graphs, where $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Their Cartesian product $\Gamma_1 \Box \Gamma_2$ is the labeled-graph $(G_1 \Box G_2, \mathcal{L}_1 \Box \mathcal{L}_2)$, where $\mathcal{L}_1 \Box \mathcal{L}_2$ is defined respectively on every hyperedge of A_1 and every hyperedge of A_2 (cf. Definition 5) by

- $\mathcal{L}_1 \Box \mathcal{L}_2(\{x\} \times \{u, v\}) = \{\{x\} \times e : e \in \mathcal{L}_2(\{u, v\}\}$.
- $\mathcal{L}_1 \Box \mathcal{L}_2(\{x, y\} \times \{u\}) = \{e \times \{u\} : e \in \mathcal{L}_1(\{x, y\}\}$.

It is easy to check that $\mathcal{L}_1 \Box \mathcal{L}_2$ is a function from $E_1 \Box E_2$ to $\mathcal{P}^*(\mathcal{P}^*(V_1 \times V_2))$ and so that $\Gamma_1 \Box \Gamma_2$ is indeed a labeled-graph (using in particular the fact that $\mathcal{L}_i : E_i \to \mathcal{P}^*(\mathcal{P}^*(V_i)), i \in \{1, 2\}$).

It is also straightforward to check that, up to isomorphism, the Cartesian product on labeled-graphs is commutative and associative. That will allow us to overlook parentheses in the sequel. It is also straightforward to show that if Γ_i is isomorphic to Γ'_i, for every $i \in \bar{k}$, then $\Gamma_1 \Box \cdots \Box \Gamma_k$ and $\Gamma'_1 \Box \cdots \Box \Gamma'_k$ are.

Lemma 4. *For all hypergraphs H_1, H_2 and L2-sections Γ_1, Γ_2, we have*

1. $[H_1]_{L2} \Box [H_2]_{L2} = [H_1 \Box H_2]_{L2}$.
2. $[\Gamma_1 \Box \Gamma_2]_{L2}^{-1} = [\Gamma_1]_{L2}^{-1} \Box [\Gamma_2]_{L2}^{-1}$.

Proof. Note that second point of the lemma is an easy application of the first one and of Proposition 1. In order to show the first point, let $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$ be two hypergraphs and $[H_1]_{L2} = ([H_1]_2, \mathcal{L}_1)$ and $[H_2]_{L2} = ([H_2]_2, \mathcal{L}_2)$ be their L2-sections, where $[H_1]_2 = (V_1, \mathbb{E}_1)$ and $[H_2]_2 = (V_2, \mathbb{E}_2)$. Note that by Proposition 2, we have already $[H_1 \Box H_2]_2 = [H_1]_2 \Box [H_2]_2$. It remains to show that the labeling function of $[H_1 \Box H_2]_{L2}$ and the function $\mathcal{L}_1 \Box \mathcal{L}_2$ given in Definition 6 are equal. Let us denote by $\mathcal{L}$ the first one. Let $\{(x, u), (y, v)\} \in \mathbb{E}_1 \Box \mathbb{E}_2$, we have to show $\mathcal{L}(\{(x, u), (y, v)\} = \mathcal{L}_1 \Box \mathcal{L}_2(\{(x, u), (y, v)\})$. By definition of the L2-section, there exists a hyperedge ε_0 of $E_1 \Box E_2$ such that $(x, u), (y, v) \in \varepsilon_0$ and, clearly, either $\varepsilon_0 = \{x\} \times e_0$, where $x \in V_1$ and $e_0 \in E_2$, or $\varepsilon_0 = e_0 \times \{u\}$, where $u \in V_2$ and $e_0 \in E_1$. We show the result for the first case, the second case is similar. Since $x = y$ and $u \neq v$, clearly every hyperedge ε of $E_1 \Box E_2$ containing both (x, u) and (y, v) is of the form $\{x\} \times e$ where $e \in E_2$. It comes by Definitions 2 and 6:

$$\begin{aligned}
\mathcal{L}(\{(x, u), (x, v)\}) &= \{\varepsilon : (x, u), (x, v) \in \varepsilon \in E_1 \Box E_2\} \\
&= \{\{x\} \times e : (x, u), (x, v) \in \{x\} \times e \in E_1 \Box E_2\} \\
&= \{\{x\} \times e : u, v \in e \in E_2\} \\
&= \{\{x\} \times e : e \in \mathcal{L}_2(\{u, v\})\} \\
&= \mathcal{L}_1 \Box \mathcal{L}_2(\{(x, u), (x, v)\}). \quad \Box
\end{aligned}$$

4. Colorings and factorization of graphs

In [12], Imrich and Peterin designed an algorithm able to factorize any finite connected graph into its prime factors.

We remind the reader that hypergraphs and graphs are supposed to be non-trivial (that is, non-reduced to a single vertex and having at least one edge).In particular, if $G = G_1\square\cdots\square G_k$ is a Cartesian product of graphs, its factors $G_1, \cdots, G_k$ are all supposed non-trivial.

4.1. The algorithm of Imrich and Peterin

A *prime graph* is a graph which cannot be factorized as a Cartesian product of non-trivial graphs. A factorization is *prime* if each factor is *prime*.

The algorithm of Imrich and Peterin is based on the fact that if $G = G_1\square\cdots\square G_{\underline{k}}$ then this factorization induces a *coloring* of the edges of G. Indeed, if $G = G_1\square\cdots\square G_k$ then, for all $u \in V_1 \times \cdots \times V_k$ and $i \in \bar{k}$, there is a subgraph G_i^u of G such that the i^{th} projection p_i induces an isomorphism between G_i^u and G_i. Indeed, as already noticed, $\{u, v\} \subseteq V(G)$ is a edge of G if and only if there exists a unique $i \in \bar{k}$ such that $\{u_i, v_i\}$ is an edge of G_i, and $u_j = v_j$, for every $j \in \bar{k}, j \neq i$. An edge of that form is said to be an *i-edge* or to have the color i. Since, every edge of G is an i-edge for a unique i, clearly $\bar{k}$ induces a coloring of the edges of G. The graph G_i^u is then defined as the connected subgraph of G induced by the set of vertices V_i^u, where a vertex belongs to V_i^u if and only if it is connected to u by an i-path, that is, a path containing only i-edges. The set of edges of G_i^u is denoted below by $\mathbb{E}_i^u$.

Subgraphs of the form G_i^u are said to be the i-layers of G and it is clear that every edge of G is contained in exactly one layer. Note that if v is a vertex of G_i^u then clearly $G_i^v = G_i^u$. This implies that the vertex used to denote G_i^u can be freely chosen among the vertices of V_i^u. It is also clear that two distinct i-layers are vertex-disjoint. Finally, since each G_i is connected, it is easy to check that every vertex u is adjacent to at least one i-edge, for each $i \in \bar{k}$, and so G_i^u is a non-trivial graph.

Note that $\{u, v\}$ is an i-edge of G if and only if $u = v[i := u_i]$ (or equivalently $v = u[i := v_i]$) and $\{u_i, v_i\}$ is an edge of G_i. More generally, u and v are connected by an i-path of length n in G_i^u if and only if $u = v[i := u_i]$ (or equivalently $v = u[i := v_i]$) and u_i and v_i are connected by a path of length n is G_i. It is also easy to check that u and v are connected by a path in G if and only if, for every $i \in \bar{k}$, u_i and v_i are connected by a path in G_i.

Let now $\{u, v\}$ and $\{u, w\}$ be respectively an i-edge and a j-edge of G, where $i \neq j$. We show that these edges are contained into an induced square of G. Indeed, by hypothesis we have $v = u[i := v_i]$ and $\{v_i, u_i\} \in E_i$, as well as, $w = u[j := w_j]$ and $\{w_j, u_j\} \in E_j$. Moreover, since $i \neq j$, we have $w_i = u_i$ and $v_j = u_j$, and so $\{v_i, w_i\} \in E_i$ and $\{w_j, v_j\} \in E_j$. Hence, since $u' = u[i := v_i][j := w_j]$ is a vertex of G such that $u'_i = v_i$ and $u'_j = w_j$, it comes $\{u'_i, w_i\} \in E_i$ and $\{u'_j, v_j\} \in E_j$. Since moreover w and u' differ only of their i^{th}-coordinate, and v and u' differ only on their j^{th}-coordinate, $\{w, u'\}$ and $\{v, u'\}$ are respectively an i-edge and a j-edge of G. It is also easy to see that there is no edge between u and u' and no edge between v and w. Hence, the subgraph of G induced by $V' = \{u, v, w, u'\}$ is a square. Note that opposite edges in this square have the same color. Moreover, it is easy to show that this induced square is the unique one containing both $\{u, v\}$ and $\{u, w\}$. This fact is expressed in the following result from [12].

Lemma 5 (*Square Lemma*). *Let $G = G_1\square\cdots\square G_k$ be a cartesian product. If two edges of G are adjacent edges with different colors then they lay in a unique induced square (with opposite edges in the square having the same color).*

A straightforward consequence of the Square Lemma is that every clique of G is necessarily contained in the same layer. From the Square Lemma we easily get the following result, also given in [12].

Lemma 6. *Let $G = G_1\square\cdots\square G_k$. Then every clique of G is contained in a single layer. Moreover, if two cliques share an edge then they both are contained in the same layer.*

Since a hyperedge of a hypergraph H induces a clique in its 2-section, the definition of the labeling function in the L2-section gives easily.

Corollary 1. *If $\Gamma = (G, \mathcal{L})$ is the L2-section of H and $G = G_1\square\cdots\square G_k$ then e is a clique of G_i^u for all $i \in \bar{k}$ and i-edge $\{u, v\}$ such that $e \in \mathcal{L}(\{u, v\})$.*

4.2. I-paths, I-layers, I-projection and edge-induced isomorphisms

We fix now a graph $G = G_1\square\cdots\square G_k$.

We generalize first the notion of i-layer to the notion of *I-layer*, where $I \subseteq \bar{k}$. Indeed, we let $\{u, v\}$ be an I-edge of G if and only if $\{u, v\}$ is an i-edge for some $i \in I$. Then, for every $u \in V$, the *I-layer* $G_I^u = (V_I^u, \mathbb{E}_I^u)$ is the connected subgraph of G induced by the set of vertices V_I^u, where a vertex of G belongs to V_I^u if and only if it is connected to u by an I-path, that is, a path containing only I-edges.

For $I = \{i_1, \ldots, i_n\} \subseteq \bar{k}$, p_I is the I-projection on G, that is, the mapping which associates with every sequence $u \in V$ the sequence $u_{i_1}, \ldots, u_{i_n} \in V_{i_1} \times \cdots \times V_{i_n}$. To simplify notations, $p_I(u)$ is often simply written u_I and $v[I := u_I]$ is the k-tuple obtained from v by replacing v_i by u_i, for every $i \in I$.

Note that p_I extends naturally to a function on $\mathcal{P}(V)$ by letting $p_I(e) = \{p_I(x) : x \in e\}$, for every $e \subseteq V$. Similarly, it extends to a function on $\mathcal{P}(\mathcal{P}(V))$ by letting $p_I(L) = \{p_I(e) : e \in L\}$, for every $L \subseteq \mathcal{P}(V)$. We use this remark freely in the sequel.

Note finally that we denote by G_I the graph $G_{i_1}\square\cdots\square G_{i_n}$, where $I = \{i_1, \ldots, i_n\}$ is ordered according to the natural order on $\bar{k}$.

Lemma 7. *Two vertices u, v of G are connected by an I-path, $I \subseteq \bar{k}$, if and only if these vertices have the same coordinates except for some coordinates of I.*

Proof. Indeed, since $\{u, v\}$ is an i-edge of G if and only if $u[i := v_i]$ and $\{u_i, v_i\}$ is an edge of G_i, it is easy to check that if u, v are connected by an I-path then their coordinates are the same except for some coordinates of I.

Now suppose that u, v are equal on their coordinates except for some coordinates of I. We show the result by induction on the number n of such coordinates. If $n = 0$ then $u = v$ and the trivial path u connects u to v. Now suppose that u, v are equal on their coordinates except for $n + 1$ coordinates of I and let $i \in I$ be such that a coordinate. Let now $u' = u[i := v_i]$. Clearly u' have the same coordinates as v except for n coordinates of I. Hence, by induction hypothesis, there exists an I-path q between u' and v in G. Now, since G_i is connected, there exists a path between u_i and v_i in G_i. As noticed previously, this path induces an i-path p in G between u and u'. Since $i \in I$, clearly pq is an I-path connecting u and v. $\square$

By definition of $v[I := u_I]$, we have $v[I := u_I]_j = v_j$, for every $j \notin I$. Moreover, if $u_i = v_i$ for some $i \in I$ then clearly $v[I := u_I] = v[I \setminus \{i\} := u_{I\setminus\{i\}}]$. It comes easily.

Corollary 2. *Two vertices u, v are connected by an I-path if and only if $u = v[I := u_I]$ if and only if $v = u[I := v_I]$.*

Corollary 3. *If v, w are vertices of G_I^u, then $v_j = w_j$, for every $j \notin I$.*

Proof. Indeed, since v, w are in G_I^u which is clearly connected by definition, there exists an I-path between w and v. Hence, by the lemma, their coordinates out of I are the same. $\square$

Corollary 4. *Every edge of G_I^u is an I-edge.*

Proof. Let $\{v, w\}$ be an edge of G_I^u. By definition of G_I^u as the subgraph of G induced by V_I^u, $\{v, w\}$ is an edge of G and $v, w \in V_I^u$. So, in particular, $\{v, w\}$ must be a j-edge for some $j \in \bar{k}$. Hence, $\{v_j, w_j\}$ is an edge of G_j and so $v_j \neq w_j$. If we suppose now that $j \notin I$, since $v, w \in V_I^u$, we get $v_j = w_j$ by the previous corollary. Contradiction. $\square$

Lemma 8. *For all $u \in V$ and non-empty set $I \subseteq \bar{k}$, the restriction p_I^u of p_I to G_I^u is a graph isomorphism between G_I^u and G_I.*

Proof. Let $I = \{i_1, \ldots, i_n\} \subseteq \bar{k}$ and let u be a vertex of $V_1 \times \cdots \times V_k$. In order to simplify notations, p_I^u is simply written p_I. By definition $p_I(v)$ is the sequence $v_{i_1}, \ldots, v_{i_n}$ and so clearly p_I is a function from V_I^u to $V_{i_1} \times \cdots \times V_{i_n}$.

Now if $p_I(v) = p_I(w)$, where $v, w \in V_I^u$, this means that $v_i = w_i$, for every $i \in I$. Since v, w are both in G_I^u, we have also $v_j = w_j$ for every $j \notin I$ by Corollary 3. So $v = w$ and p_I (restricted to G_I^u) is an injection.

Now to see that p_I is a surjection, let $x_1, \ldots, x_n$ be a vertex of $V_{i_1} \times \cdots \times V_{i_n}$. Since G_I is a connected graph, there is a path p from the vertex $u_{i_1}, \ldots, u_{i_n}$ to $x_1, \ldots, x_n$. We show now by induction on the length of p that there exists $v \in V_I^u$ such that $p_I(v) = x_1, \ldots, x_n$ and such that v is connected to u by a path of same length as p in G_I^u. If the length is 0, that is, if $u_{i_1}, \ldots, u_{i_n} = x_1, \ldots, x_n$ then u is such a vertex and path (of length 0).Now suppose p of length $m + 1$ between $u_{i_1}, \ldots, u_{i_n}$ and $x_1, \ldots, x_n$. The path can be decomposed into a path q of length m and a last edge incident to $x_1, \ldots, x_n$. By induction hypothesis, the last vertex of q is of the form $w_{i_1}, \ldots, w_{i_n}$, for some $w \in V_I^u$ connected to u by a path q' of the same length as q. The last edge links $w_{i_1}, \ldots, w_{i_n}$ and $x_1, \ldots, x_n$ and, since G_I is a Cartesian product, this edge must be an i_j-edge for some $j \in \bar{n}$. Hence, we have $\{w_{i_j}, x_j\} \in E_{i_j}$ and $w_{i_l} = x_l$, for every $l \in \bar{n} \setminus \{j\}$. It is then easy to check that $w[i_j := x_j]$ and q' extended with the i_j-edge $\{w, w[i_j := x_j]\}$ are respectively the vertex and the path we are looking for.

Finally, by Corollary 4, every edge of G_i^u is an i-edge for some $i \in I$. It is then easy to check that p_I associates an i-edge of G_I with this edge, proving that p_I is a graph morphism. $\square$

Lemma 9. *Let G_I^u and G_I^v be disjoint I-layers of G, where $I \subseteq \bar{k}$ is non-empty, and $\{u, v\}$ be a j-edge, for $j \notin I$. The function f_I^{uv} defined by $f_I^{uv}(w) = w[j := v_j]$, for every vertex w of G_I^u, is a graph isomorphism between G_I^u and G_I^v such that $f_c^{uv}(u) = v$. It is moreover edge-color preserving and $\{w, w[j := v_j]\}$ is a j-edge.*

Proof. First, let us show the injectivity of f_I^{uv}. Indeed, let w, z be vertices of G_I^u. If $w[j := v_j] = z[j := v_j]$ then, for every $i \in \bar{k} \setminus \{j\}$, $w_i = z_i$. Moreover, since $j \notin I$, we have $w_j = z_j$ by Corollary 3, and so $w = z$.

Second, by Lemma 8, both G_I^u and G_I^v are isomorphic to G_I, and so have the same cardinality. That shows that f_I^{uv} is bijective, since the graphs are supposed finite in this article.

Finally, since there is a j-edge between u and v, we have $v = u[j := u_j] = f_i^{uv}(u)$. Moreover, since an edge $\{x, y\}$ of the Cartesian product G is an i-edge if and only if x and y are equal for every coordinate different from i and $\{x_i, y_i\}$ is an edge of G_i, and since $j \notin I$, it is straightforward to check that $\{w, z\}$ is an i-edge of G_I^u if and only if $\{w[j := v_j], z[j := v_j]\}$ is an i-edge of G_I^v. Hence, f_I^{uv} is an edge-color preserving graph isomorphism. It is also straightforward to check that $\{w, w[j := v_j]\}$ is a j-edge using the Square Lemma. $\square$

Corollary 5. *Under the hypotheses of the lemma, $f_I^{uv} = f_I^{u'v'}$ for all vertex u' in G_I^u and v' in G_I^v such that $\{u', v'\}$ is a j-edge.*

Proof. Indeed, since v' is in G_I^v and $j \notin I$, we have $v'_j = v_j$ (Corollary 3). Since moreover $u' \in G_I^u$, we have $G_I^u = G_I^{u'}$, and so the result. $\square$

5. Prime factorization of hypergraphs via their L2-sections

In this section we fix an hypergraph $H = (V, E)$, its 2-section $G = (V, \mathbb{E})$ and its L2-section $\Gamma = (G, \mathcal{L})$. We suppose moreover that $G = G_1 \Box \cdots \Box G_k$ is colored with the set $\bar{k}$ as shown previously.

We show how the prime factorization of H can be deduced from the prime factorization of G using Γ. This is done by introducing an equivalence relation on $\bar{k}$ (cf. Definition 9) from which are defined the prime factors of Γ.

Let m be an integer and let, for every $i \in \bar{m}$, f_i be a graph isomorphism from a graph G_i to a graph G'_i. It is straightforward to show that the function $f_1 \times \cdots \times f_m$ defined, for every vertex $x_1, \ldots, x_m \in V_1 \times \cdots \times V_m$, by $f_1 \times \cdots \times f_m(x_1, \ldots, x_m) = (f_1(x_1), \ldots, f_m(x_m))$, is a graph isomorphism from $G_1 \Box \cdots \Box G_m$ to $G'_1 \Box \cdots \Box G'_m$.

Definition 7. Let $c_1, \ldots, c_m$ be a partition of $\bar{k}$ and u be a vertex of G. We defined the function $h_u : V_{c_1}^u \times \cdots \times V_{c_m}^u \mapsto V_1 \times \cdots \times V_k$ on every $\bar{v} \in V_{c_1}^u \times \cdots \times V_{c_m}^u$ by $h_u(\bar{v})$ is the sequence which i^{th} coordinate is equal to $p_i(\bar{v}_j)$, where $i \in c_j$, for every $i \in \bar{k}$ and $j \in \bar{m}$.

Lemma 10. *For all partition $c_1, \ldots, c_m$ of $\bar{k}$ and vertex u of G, h_u is a graph isomorphism from $G_{c_1}^u \Box \cdots \Box G_{c_m}^u$ to $G_1 \Box \cdots \Box G_k$.*

Proof. It is easy to check that h_u in Definition 7 is well defined as a function from $V_{c_1}^u \times \cdots \times V_{c_m}^u$ to $V_1 \times \cdots \times V_k$. Note that by the remark above and Lemma 8, the function $p_{c_1} \times \cdots \times p_{c_m}$ is a graph isomorphism from $G_{c_1}^u \Box \cdots \Box G_{c_m}^u$ to $G_{c_1} \Box \cdots \Box G_{c_m}$. Now let $j_1, j_2, \ldots, j_k$ be the elements of $\bar{k}$ ordered in such a way that the indices of c_1 appear first in the natural order, then the indices of c_2 appear second in the natural order, and so on until c_m. Since $c_1, \ldots, c_m$ is a partition of $\bar{k}$, the function $\pi : \bar{k} \mapsto \bar{k}$ defined by $\pi(i) = j_i$ is a permutation. It is then straightforward to check that the function h_u is equal to the function $f_\pi^{-1} \circ (p_{c_1} \times \cdots \times p_{c_m})$, where f_π is the isomorphism induced by π. Hence h_u is a graph isomorphism as a composition of two graph isomorphisms. $\square$

Definition 8. For every $c \subseteq \bar{k}$ and $u \in V$, we let Γ_c^u be the graph G_c^u endowed with the restriction of $\mathcal{L}$ to $\mathbb{E}_c^u$. The labeled-graph Γ_c^u is called the *c-Cartesian join* of u and we let $H_c^u = [\Gamma_c^u]_{L2}^{-1}$. If $c = \{i\}$ then Γ_c^u (resp. H_c^u) is simply written Γ_i^u (resp. H_i^u).

In order to simplify notations we use $\mathcal{L}$ to denote the restriction of $\mathcal{L}$ to $\mathbb{E}_c^u$, that is, we write $\Gamma_c^u = (G_c^u, \mathcal{L})$.

Lemma 11. *For all $u \in V$ and non-empty $c \subseteq \bar{k}$, H_c^u is a partial hypergraph of H with L2-section Γ_c^u.*

Proof. By Lemma 2, it is sufficient to show that Γ_c^u is a subsection of Γ. We prove first that G_c^u is the subgraph of G induced by $\mathbb{E}_c^u$, that is, we have to prove $V_c^u = \bigcup \mathbb{E}_c^u$. Since G_c^u is by definition the subgraph induced by V_c^u, we have $\mathbb{E}_c^u = \{\{v, w\} \in \mathbb{E} : v, w \in V_c^u\}$ and so $\bigcup \mathbb{E}_c^u \subseteq V_c^u$. Now, if $v \in V_c^u$, since clearly G_c^u is connected, there exists w such that $\{v, w\} \in \mathbb{E}_c^u$. Hence, we have also $V_c^u \subseteq \bigcup \mathbb{E}_c^u$.

The fact that G_c^u is endowed with the restriction of $\mathcal{L}$ to $\mathbb{E}_c^u$ is given by definition, so it remains to show the third condition of Definition 4. Let $e \in \mathcal{L}(\{v_0, w_0\})$, where $\{v_0, w_0\} \in \mathbb{E}_c^u$, and let $\{v, w\}$ in $\mathcal{P}_2(e)$. By hypothesis $\{v_0, w_0\}$ is a c-edge and so an i-edge for some $i \in c$. Moreover, by definition of $\mathcal{L}$, we have $e \in \mathcal{L}(\{v_0, w_0\})$ if and only if $v_0, w_0 \in e \in E$. Note now that, by Corollary 1, e is a clique of $G_i^{v_0}$. Hence, since $v, w \in e$, $\{v, w\}$ and $\{v_0, w_0\}$ are edges of the same i-layer $G_i^{v_0}$. Finally, since $v_0 \in V_c^u$, we get $\{v, w\} \in \mathbb{E}_c^u$, proving the result. $\square$

We define now an equivalence relation $\mathcal{R}^*$ on the set of colors $\bar{k}$ which uses the graph isomorphisms of the type f_c^{uv} given by Lemma 9. These graphs isomorphisms are called *edge-induced* isomorphisms and they are denoted by f_i^{uv} when $c = \{i\}$.

Definition 9. We let $\mathcal{R}$ be the binary relation on $\bar{k}$ defined for every ordered pair of distinct i, j by: $i\mathcal{R}j$ if and only if there exist distinct i-layers G_i^u and G_i^v adjacent by a j-edge $\{u, v\}$ such that the edge-induced graph isomorphism f_i^{uv} is not an L2-isomorphism between Γ_i^u and Γ_i^v.

We define now $\mathcal{R}^*$ as the reflexive, symmetric and transitive closure of $\mathcal{R}$, and we let $\bar{k}/\mathcal{R}^*$ be the quotient of $\bar{k}$ by $\mathcal{R}^*$.

Proposition 3. *For all $u, v \in V$ and $c \in \bar{k}/\mathcal{R}^*$, the function g_c^{uv} which associates $w[\bar{k} \setminus c := v_{\bar{k}\setminus c}]$ with every vertex w of G_c^u is an L2-isomorphism between Γ_c^u and Γ_c^v.*

Proof. As a preliminary remark, note that $v_{\bar{k}\setminus c} = v'_{\bar{k}\setminus c}$ for all pair of vertices v, v' belonging to the same c-layer (Corollary 3). Hence, in particular, $w[\bar{k} \setminus c := v_{\bar{k}\setminus c}] = w[\bar{k} \setminus c := v'_{\bar{k}\setminus c}]$, for all $v' \in V_c^v$ and $w \in V$. Hence, since $G_c^u = G_c^{u'}$ for every vertex u' of G_c^u, the definition of g_c^{uv} does not depend on u and v, that is, $g_c^{u'v'} = g_c^{uv}$, for all u' in G_c^u and v' in G_c^v.

The proof is now by induction on the distance d between G_c^u and G_c^v. If $d = 0$ then $G_c^u = G_c^v$ (by definition of the distance between two subgraphs). Hence, since $v_{\bar{k}\setminus c} = w_{\bar{k}\setminus c}$ for every $w \in V_c^u = V_c^v$ (Corollary 3), the function g_c^{uv} is the identity function on V_c^u, and so trivially an L2-isomorphism.

Suppose now that the distance between G_c^u and G_c^v is $n+1$ and let p be a path of minimal length $n+1$ between a vertex of G_c^u and a vertex of G_c^v. By the preliminary remark, we can suppose that p connects u and v and so that $p = u, \ldots, z, v$, where $u, \ldots, z$ is a path of length n and $\{z, v\}$ is a j-edge, for some $j \in \bar{k}$. Note that, by minimality of p, we have $j \notin c$ and so $G_c^z \neq G_c^v$, otherwise we would have $z, v \in V_c^v$ and so $\{z, v\}$ would be an edge of G_c^v, contradicting Corollary 4. By induction hypothesis the function g_c^{uz} is an L2-isomorphism.

Now let w be a vertex of G_c^z. Since $\{z, v\}$ is a j-edge, we have $z_i = v_i$, for every $i \neq j$. Hence, by the preliminary remark, for every $i \in (\bar{k} \setminus c) \setminus \{j\}$, we have $w_i = z_i = v_i$, and so $w[\bar{k} \setminus c := v_{\bar{k}\setminus c}] = w[j := v_j]$. This shows that the function g_c^{zv} is equal to the function f_c^{zv} introduced in Lemma 9. Hence, g_c^{zv} is a graph isomorphism between G_c^z and G_c^v by the same lemma. Note that the lemma also gives that $\{w, w[j := v_j]\}$ is a j-edge, for every $w \in V_c^z$. Now, since $j \notin c$ and c is a class modulo $\mathcal{R}^*$, we cannot have $i\mathcal{R}j$ for any $i \in c$. Hence, in particular, $f_i^{ww[j:=v_j]}$ is a L2-isomorphism, for all $w \in V_c^z$ and $i \in c$. Each such $f_i^{ww[j:=v_j]}$ is clearly the restriction of f_c^{zv} to G_i^w, and so Corollaries 1 and 4 imply easily that $f_c^{zv} = g_c^{zv}$ is an L2-isomorphism. Hence, for every vertex w of G_c^u, it comes: $g_c^{zv} \circ g_c^{uz}(w) = w[\bar{k} \setminus c := z_{\bar{k}\setminus c}][\bar{k} \setminus c := v_{\bar{k}\setminus c}] = w[\bar{k} \setminus c := v_{\bar{k}\setminus c}] = g_c^{uv}(w)$. Hence, as a composition of L2-isomorphisms, g_c^{uv} is a L2-isomorphism. $\square$

Theorem 1. *Let $\Gamma = (G, \mathcal{L})$ be the L2-section of a hypergraph H. If $G = G_1 \square \cdots \square G_k$ then $H \cong \prod_{c\in\bar{k}/\mathcal{R}^*} H_c^u$, for every vertex $u \in V$.*

Proof. Let $u \in V_1 \times \cdots \times V_k$. Note that by Lemma 11, Γ_c^u is an L2-section and $H_c^u = [\Gamma_c^u]_{L2}^{-1}$ is a partial hypergraph of H, for each $c \in \bar{k}/\mathcal{R}^*$. We must show $H \cong \prod_{c\in\bar{k}/\mathcal{R}^*} H_c^u$. Since hypergraphs are isomorphic if and only if their L2-sections are (Lemma 1), and since the Cartesian product commutes with the L2-section operation (Lemma 4), we must equivalently show that $\Gamma \cong \prod_{c\in\bar{k}/\mathcal{R}^*} \Gamma_c^u$.

Let now $c_1, \ldots, c_m$ be an enumeration of $\bar{k}/\mathcal{R}^*$. By Lemma 10, the function h_u introduced by Definition 7 is a graph isomorphism from $G_{c_1}^u \square \cdots \square G_{c_m}^u$ to $G_1 \square \cdots \square G_k$.

We show that h_u is also an L2-isomorphism. Indeed, let $\{\bar{v}, \bar{w}\}$ be an edge of $G_{c_1}^u \square \cdots \square G_{c_m}^u$. We must show $\mathcal{L}(\{h_u(\bar{v}), h_u(\bar{w})\}) = \{h_u(\epsilon) : \epsilon \in \mathcal{L}\square \cdots \square\mathcal{L}(\{\bar{v}, \bar{w}\})\}$. Now, $\{\bar{v}, \bar{w}\}$ is a c_j-edge for some $j \in \bar{m}$. So we have $\bar{v} = \bar{w}[j := \bar{v}_j]$ and $\{\bar{v}_j, \bar{w}_j\}$ is an edge of $G_{c_j}^u$. Let $v = h_u(\bar{v})$ and $g_{c_j}^{uv}$ be the function given in Proposition 3. In order to simplify notations, we let $g = g_{c_j}^{uv}$. Hence, by definition we have $g(w) = w[\bar{k} \setminus c_j := v_{\bar{k}\setminus c_j}]$, for every $w \in V_{c_j}^u$. We show that, for every such w, we have $h_u(\bar{v}[j := w]) = g(w)$. Indeed, let $i \in \bar{k}$. If $i \in c_j$ then $w[\bar{k} \setminus c_j := v_{\bar{k}\setminus c_j}]_i = w_i$ and clearly $h_u(\bar{v}[j := w])_i = w_i$ by definition of h_u. Now if $i \notin c_j$, we have $w[\bar{k} \setminus c_j := v_{\bar{k}\setminus c_j}]_i = v_i$ which is also equal to $h_u(\bar{v}[j := w])_i$ since clearly $h_u(\bar{v}[j := w])_i = h_u(\bar{v})_i = v_i$. Hence, $h_u(\bar{v}[j := w])$ and $g(w)$ are equal on all their coordinates, and so equal.

Now, note that g is an L2-isomorphism from $\Gamma_{c_j}^u$ to $\Gamma_{c_j}^v$ and that every hyperedge of H containing an edge of $G_{c_j}^u$ is a subset of $V_{c_j}^u$ (Corollary 1). Moreover, by the definition of the Cartesian product of L2-sections, $\epsilon \in \mathcal{L}\square \cdots \square\mathcal{L}(\{\bar{v}, \bar{w}\})$ if and only if there exists $e \in \mathcal{L}(\{\bar{v}_j, \bar{w}_j\})$ such that $\epsilon = \{\bar{v}[j := w] : w \in e\}$. Hence, it comes:

$$
\begin{aligned}
\mathcal{L}(\{h_u(\bar{v}), h_u(\bar{w})\}) &= \mathcal{L}(\{g(\bar{v}_j), g(\bar{w}_j)\}) \\
&= \{g(e) : e \in \mathcal{L}(\{\bar{v}_j, \bar{w}_j\})\} \\
&= \{\{g(w) : w \in e\} : e \in \mathcal{L}(\{\bar{v}_j, \bar{w}_j\})\} \\
&= \{\{h_u(\bar{v}[j := w]) : w \in e\} : e \in \mathcal{L}(\{\bar{v}_j, \bar{w}_j\})\} \\
&= \{h_u(\{\bar{v}[j := w]) : w \in e\}) : e \in \mathcal{L}(\{\bar{v}_j, \bar{w}_j\})\} \\
&= \{h_u(\epsilon) : \epsilon \in \mathcal{L}\square \cdots \square\mathcal{L}(\{\bar{v}, \bar{w}\})\}. \quad \square
\end{aligned}
$$

We introduce now a lemma used to show that factors of L2-sections are necessarily Cartesian joins. In the lemma, the L2-sections denoted by $\Gamma_{(i)}$ are arbitrary L2-sections, the index (i) is only used to enumerate the factors.

Lemma 12. *Let $\Gamma = (G, \mathcal{L})$ be an L2-section. If $\Gamma = \Gamma_{(1)}\square\Gamma_{(2)}\square \cdots \square\Gamma_{(l)}$ is a prime decomposition of Γ, ($\Gamma_{(i)}$ non-trivial, for $i \in \{1, \ldots, l\}$) then $\Gamma_{(i)}$ is an $\mathcal{R}^*$-induced Cartesian join for all $i \in \{1, \ldots, l\}$, and so we can write $\Gamma = \Gamma_{c_1}\square\Gamma_{c_2}\square \cdots \square\Gamma_{c_l}$, with $\Gamma_{(i)} = \Gamma_{c_i}$, for $\{c_1, c_2, \ldots, c_l\} = \bar{k}/\mathcal{R}^*$ a partition of $\bar{k}$, the set of colors obtained from the prime factorization of G.*

Proof. Note first that if $\Gamma = \Gamma_{(1)}\square\Gamma_{(2)}\square \cdots \square\Gamma_{(l)}$, where $\Gamma_{(i)}$ is an L2-section for every $i \in \{1, \ldots, l\}$, then we have obviously $G = G_{(1)}\square G_{(2)}\square \cdots \square G_{(l)}$, where $\Gamma_{(i)} = (G_{(i)}, \mathcal{L}_{(i)})$, for every $i \in \{1, \ldots, l\}$. As G admits a prime factorization we can write $G = \prod_{i=1}^{k} G_i$, for some $k \geq l$. As this prime factorization is unique, each graph $G_{(i)}$ is a Cartesian product of the form $\prod_{j\in c_i} G_j$, where $c_i \subseteq \bar{k}$. Obviously, we have $c_i \neq \emptyset$ (otherwise $G_{(i)}$ would be the trivial graph), $c_i \cap c_{i'} = \emptyset$, for all distinct $i, i' \in \{1, \ldots, l\}$, and $\bigcup_{i=1}^{l} c_i = \bar{k}$. Hence, $\{c_i\}_{i\in\{1,\ldots,l\}}$ is a partition of $\bar{k}$.

The equality $\Gamma = \prod_{i=1}^{l} \Gamma_{(i)}$ implies $\mathcal{L}_{(i)}$ needs to be defined as $p_{c_i}\left(\mathcal{L}_{|\{e\in\mathbb{E}: e \text{ is a } c_i-\text{edge}\}}\right)$. The last mapping is an L_2-isomorphism as soon as each Cartesian join induced by c_i-edges $\Gamma_{c_i}^u$, $u \in V$, are pairwise $L2$-isomorphic, what is true as soon as c_i are unions of equivalence classes in $\bar{k}/\mathcal{R}^*$, by definition (otherwise, there exists $i \in c_i, j \notin c_i$ such that $i\mathcal{R}j$ or $j\mathcal{R}i$, so j-adjacent c_i-layers are not necessarily L2-isomorphic or c_j layers c_i-adjacent are not). Minimal such unions are elements of $\bar{k}/\mathcal{R}^*$. $\square$

Theorem 2. *Let $(G, \mathcal{L})$ be the L2-section of a hypergraph H. If $G = G_1 \square \cdots \square G_k$ then, for all vertex $u \in V$ and $c \in \bar{k}/\mathcal{R}^*$, the partial hypergraph H_c^u is prime.*

Proof. Let $u \in V$ and $c \in \bar{k}/\mathcal{R}^*$. By Lemma 4, it is sufficient to show that Γ_c^u is prime. Suppose it is not the case. By Lemma 12, if Γ_c^u has at least two prime factors then there exists a partition $\{c_1, c_2\}$ of c such that and $\Gamma_c^u = \Gamma_{c_1}^u \square \Gamma_{c_2}^u$. As we have $c_1, c_2 \subsetneq c \in \bar{k}/\mathcal{R}^*$, it comes that $c_1 \notin \bar{k}/\mathcal{R}^*$, $c_2 \notin \bar{k}/\mathcal{R}^*$, contradicting Lemma 12. □

Fig. 2 gives the prime decomposition of a hypergraph $\mathcal{H}$ which has the same 2-section as the hypergraph in Fig. 1. This 2-section can be factorized as $K_3 \square K_2 \square K_2$ and so the number k of its prime factors is 3. Nevertheless, and contrary to the hypergraph in Fig. 1, $\mathcal{H}$ cannot be factorized as a product of three prime factors. Indeed, the quotient of $\bar{k} = \{1, 2, 3\}$ by $\mathcal{R}^*$ contains two equivalent classes $c_1 = \{1, 2\}$ and $c_2 = \{3\}$ and so, according to the results above, $\mathcal{H}$ has only two factors corresponding to these classes.

6. Hypergraph prime factorization algorithm

6.1. A general prime factorization algorithm

We present now Algorithms 1 and 2. The first one computes the prime factorization of every hypergraph H using the second one. Algorithm 2 decides the relation $\mathcal{R}$. We suppose implemented the operations $[\_]_{L2}$ and $[\_]_{L2}^{-1}$.

Algorithm 1 Hypergraph prime decomposition

Require: A hypergraph $H = (V, E)$.
Return: The set of all prime factors of H.

1: Compute $\Gamma = (G; \mathcal{L})$, the L2-section of H.
2: Run the prime factorization algorithm of Imrich and Peterin on G and let $G_1, \ldots, G_k$ be its prime factors.
3: T is an array of length k connecting each color $i \in \bar{k}$ to its class index.
4: **For** $i = 1$ to k **do**
5:   $c_i = \{i\}$; $T[i] = i$
6: **EndFor**
7: **For** $i = 1$ to $k - 1$ and $j = i + 1$ to k **do**
8:   **if** $i\mathcal{R}j$ Or $j\mathcal{R}i$ **then**
9:     $i_0 = T[i]$
10:     $j_0 = T[j]$
11:     **if** $i_0 < j_0$ **then**
12:       $c_{i_0} = c_{i_0} \cup c_{j_0}$
13:       **For** all $l \in c_{j_0}$ **do**
14:         $T[l] = i_0$
15:       **EndFor**
16:     **end if**
17:     **if** $j_0 < i_0$ **then**
18:       $c_{j_0} = c_{i_0} \cup c_{j_0}$
19:       **For** all $l \in c_{i_0}$ **do**
20:         $T[l] = j_0$
21:       **EndFor**
22:     **end if**
23:   **end if**
24: **EndFor**
25: $\bar{k}/\mathcal{R}^* = \{c_j : \exists i \in \bar{k},\ T[i] = j\}$
26: **return** $\{H_c^u : c \in \bar{k}/\mathcal{R}^*\}$, where u can be any vertex of V

Theorem 3. *Algorithms* 1 *and* 2 *are sound and complete.*

Proof. Concerning Algorithm 2, it is clearly sufficient to show that, for given $i, j \in \bar{k}$, $i\mathcal{R}j$ if and only if $\{e[j := v_j] : e \in \mathcal{L}(\{u, w\})\} \neq \mathcal{L}(\{v, w[j := v_j]\})$, for some i-edge $\{u, w\}$ and j-edge $\{u, v\}$. This can be proved straightforwardly using Lemma 9, its corollary and the definition of $\mathcal{R}$.

Concerning Algorithm 1. The algorithm is designed in such a way that at each execution of the **For** loop in Line 7, the class indices $T[i]$ and $T[j]$ are set at the minimums i_0 and j_0 of the current classes of i and j. This is ensured by the initialization **For** loop in Line 4, by the fact that only the class indexed by the minimum of $\{i_0, j_0\}$ is updated in Line 12 or 18, and by the fact that either $T[l]$, for all $l \in c_{j_0}$, or $T[l]$, for all $l \in c_{i_0}$, are set to this minimum depending on the test of the **if** in Line 11 and 17. Since these changes are done only when $i\mathcal{R}j$ or $j\mathcal{R}i$, it is clear that if $T[i] = T[j]$ at the end of the execution then $i\mathcal{R}^*j$.

Algorithm 2 $\mathcal{R}$-test

Require: A Cartesianly colored L2-section $\Gamma = (G; \mathcal{L})$. For all $u \in V$ and $i \in \bar{k}$, $N_i(u) = \{w : \{u, w\} \texttt{ is an } i\texttt{-edge}\}$. Two distinct colors $i, j \in \bar{k}$.
Return: **true** if $i\mathcal{R}j$, **false** otherwise.
1: *NonChecked* contains the vertices which are not already checked.
2: Let *NonChecked* $= V$
3: **while** *NonChecked* $\neq \emptyset$ **do**
4:   Let $u \in$ *NonChecked*
5:   **For** all $w \in N_i(u)$ and $v \in N_j(u)$ **do**
6:     **if** $\{e[j := v_j] : e \in \mathcal{L}(\{u, w\})\} \neq \mathcal{L}(\{v, w[j := v_j]\})$ **then**
7:       **return true**
8:     **end if**
9:   **EndFor**
10:   *NonChecked* $=$ *NonChecked* $\setminus \{u\}$
11: **end while**
12: **return false**

Now, it is easy to check that, for each $i \in \bar{k}$, the function which associates the number of iterations of the **For** loop in Line 7 with the current value of $c_{T[i]}$ after these iterations is increasing (for inclusion). This implies in particular that $i \in c_{T[i]}$ at any time during the computation after the execution of the initialization **For** loop in Line 4. It is also easy to check that, for all $i, j \in \bar{k}$, if $T[i] = T[j]$ at some point during the computation, then $T[i]$ stays equal to $T[j]$ during the rest of the computation. Now, if $i\mathcal{R}^*j$ then there exists a sequence $i_1, \ldots, i_m$ such that $i_1 = i$, $i_m = j$ and $i_n\mathcal{R}i_{n+1}$, or $i_{n+1}\mathcal{R}i_n$, for every $n \in \bar{m}$. Using the previous facts, it is then straightforward to show by induction on m that if $i\mathcal{R}^*j$ then $T[i] = T[j]$ and $i, j \in c_{T[i]}$ at the end of the execution. □

6.2. Data structures and complexity

We present briefly the data structures and complexity of Algorithms 1 and 2. Let H be a hypergraph and $\Gamma = (G, \mathcal{L})$ be its L2-section, where $G = (V, \mathbb{E})$. We let m be the number of hyperedges of H, n the number of vertices, Δ the maximal degree of a vertex and r the rank of H.

We suppose both the vertices and the hyperedges of H implemented as integers and E implemented as an array of length m, where $E[e]$ contains the list of the vertices in the hyperedge e. Now, each hyperedge $e \in E$ generates less than r^2 edges in G, and so the number m' of edges in G is less than mr^2. Moreover, the maximal degree Δ' of a vertex in G is clearly bounded by Δr. Note also that, since G is connected, we have $O(n^2) \subseteq O(nm')$ and so $O(n^2) \subseteq O(nmr^2)$.

Clearly, the adjacency matrix M of G can be produced from E in $O(n^2) + O(mr^2) \subseteq O(nmr^2)$ and we can suppose that it is implemented as required in [12]. Moreover, we can suppose that each time an edge $\{u, v\}$ of the 2-section is extracted from the hyperedge e, this hyperedge is appended to the list $\mathcal{L}(\{u, v\})$ of the labels of $\{u, v\}$. Note that this extrawork can be performed in constant time (for instance by using linked lists) and so it does not add to the overall time complexity of the construction of M. The adjacency list of G can then be obtained from M in $O(n^2) \subseteq O(nmr^2)$. Note finally that, since the number of hyperedges containing u is at most Δ, each edge of G is labeled by at most Δ hyperedges.

Thus, the prime decomposition $G_1\square \cdots \square G_k$ of G and the corresponding coloring can be obtained in $O(m') \subseteq O(mr^2)$ by applying the algorithm of Imrich and Peterin in [12]. We also suppose that the coordinatization algorithm described in [12] is implemented. This algorithm runs in $O(m') \subseteq O(mr^2)$ space and time and allows us to interpret each vertex of G as a sequence of vertices of $V_1 \times \cdots \times V_k$. For every vertex u, we let $u[]$ be the array implementing u as a vector. Note finally that, using the coloring of G, it is possible to generate a "colored" adjacency list A of G, that is, an array which associates, with each vertex u, the list of the neighborhoods $N_i(u)$, $i \in \bar{k}$. This adjacency list can clearly be built in $O(nm') \subseteq O(nmr^2)$ and is denoted by A in the sequel.

Hence, the overall construction of Γ, factorization of G and construction of the auxiliary data structures described above, can be computed in at most $2O(nmr^2) + 2O(mr^2) + O(nmr^2) = O(nmr^2)$.

We begin now the analysis of the algorithms above with few remarks on the complexity of Algorithm 2. We suppose the requisites of the algorithm fulfilled and we let $d_i(u) = |N_i(u)|$, for all vertex u and $i \in \bar{k}$. Suppose now i, j to be distinct colors. The set *NonChecked* can be built in $O(n)$. The **while** loop in Line 3 compares, for each vertex u, the i-edges and the j-edges containing u using Line 6. The number of such i-edge is at most $d_i(u)$ and the number of j-edges at most $d_j(u)$. Now the number of substitutions performed in Line 6 for each pair of edges is clearly bounded by $\Delta r + 1$. Indeed, $\mathcal{L}(\{u, w\})$ contains at most Δ hyperedges and each hyperedge is at most of size r. Moreover, for all vertices w, v and $i \subseteq \bar{k}$, the substitution $w[i := v_i]$ can be performed in constant time by the instruction $w[i] = v[i]$, and so the overall complexity of each execution of Line 6 is $O(\Delta r)$. Finally, clearly Line 4 and 10 can be performed in constant time using an adequate data structure to implement *NonChecked*.

We discuss now the complexity of Algorithm 1. Note first that, as seen above, the execution of Line 1 and 2, as well as the construction of the "colored" adjacency list A can be done $O(nmr^2)$. Note now that $k \leq d_0$, where d_0 is the minimal degree of a vertex u in G (since by connectivity, for each $i \in \bar{k}$, there is at least one i-edge starting from u). Hence, since the maximal degree of a vertex in G is bounded by Δr, we have $k \leq \Delta r$. Hence, the initialization **For** loop in Line 4 can be performed in $O(\Delta r)$.

Now, the heart of the algorithm is the **For** loop in Line 7. Let N be the number of times this loop is iterated during the computation. Clearly, $N < k^2 \leq \Delta^2 r^2$. It is also easy to check that the instructions from Line 9 to Line 20 can all be performed in $O(k^2) \subseteq O(\Delta^2 r^2)$, since any sequence of classes $c_{i_1}, \ldots, c_{i_n}$ contains at most k colors. Hence, the **For** loop in Line 7, if we except the $\mathcal{R}$-checking in Line 8, can be performed in $O(N\Delta^2 r^2) \subseteq O(\Delta^4 r^4)$. Finally, Algorithm 2 is called $2N$ times in Line 8. These calls induce $2N$ initializations of *NonChecked* in Line 2 of Algorithm 2 and so, according to the remarks above concerning this algorithm, the total cost of these calls for Line 2 is $O(n2N) = O(nN)$. The calls induce also $2N$ executions of the **while** loop in Line 3. Since *NonChecked* is initialized at V, these executions induce $2Nn$ iterations of Line 4 and 10 of Algorithm 2, for a total cost of $O(2Nn) + O(2Nn) = O(nN)$. Finally, for each vertex u, $\sum_{i \in \bar{k}} d_i(u) \leq \Delta' \leq \Delta r$ and so the $2N$ executions of the **while** loop induce, for each vertex, at most $2\Delta r$ iterations of the test in Line 6 of Algorithm 2. Hence the total cost for all vertices of the $2N$ executions of the **while** concerning Line 6 is at most $O(2Nn2\Delta r\Delta r) = O(Nn\Delta^2 r^2)$. So the total cost of the $2N$ calls for Algorithm 2 is at most $O(nN) + O(nN) + O(Nn\Delta^2 r^2) \subseteq O(Nn\Delta^2 r^2)$. Hence, the total cost of the **For** loop in Line 7 is $O(\Delta^4 r^4) + O(Nn\Delta^2 r^2) \subseteq O(Nn\Delta^4 r^4) \subseteq O(\Delta^2 r^2 n \Delta^4 r^4) = O(\Delta^6 r^6 n)$.

The instruction in Line 25 can easily be computed in $O(k^2) \subseteq O(\Delta^2 r^2)$ by building first $J = \{j : \exists i \in \bar{k},\ T[i] = j\}$ and by letting $\bar{k}/\mathcal{R}^* = \{c_j : j \in J\}$.

Finally, in order to execute Line 26, we pick a vertex root u and, for every $j \in J$, we build a (possibly non-connected) graph F_j by removing from G all the i-edges where $i \notin c_j$. Then, clearly $G^u_{c_j}$ is the connected component of F_j containing u. Now, to find a partial hypergraph of the form H^u_c, where $c \subseteq \bar{k}$, we do not need to use explicitly its L2-section. A simple way to get H^u_c is to note that it has the same vertices as G^u_c and that a hyperedge $e \in E$ is in H^u_c if and only if at least two vertices $v, w \in e$ belong to V^u_c. Indeed, if $v, w \in e$ then $\{v, w\} \in \mathbb{E}$ and $e \in \mathcal{L}(\{v, w\})$. Hence, in particular, $\{v, w\} \in \mathbb{E}^u_c$ if and only if $v, w \in V^u_c$ by definition of G^u_c. Moreover, since all edges extracted from e appear in the same layer of G (Corollary 1), $\{v, w\} \in \mathbb{E}^u_c$ if and only if there exists an edge extracted from e in $\mathbb{E}^u_c$.

Now, for each $j \in J$, clearly F_j can be computed in $O(m'|c_j|)$. Moreover, it is well known that the connected components of a graph can be computed in linear time using either breadth-first search or depth-first search. Hence, the construction of $G^u_{c_j}$ can be made in $O(m'|c_j|)$, for every $j \in J$. Note now that $\sum_{j \in J} |c_j| = k$ and so the overall complexity to build the $G^u_{c_j}$'s is $O(m'k) \subseteq O(mr^2\Delta r) = O(mr^3\Delta)$. Now, since $\sum_{j \in J} |V^u_{c_j}| \leq n$, the construction of the $H^u_{c_j}$'s can clearly be done in $O(nm)$ by checking if the two first vertices of each hyperedge $e \in E$ belong to some $V^u_{c_j}$. So the overall cost of the execution of Line 26 is $O(mr^3\Delta) + O(nm) \subseteq O(nmr^3\Delta)$. Hence, the total cost of Algorithm 1 is

$$O(nmr^2) + O(\Delta r) + O(\Delta^6 r^6 n) + O(\Delta^2 r^2) + O(nmr^3\Delta) \subseteq O(nm\Delta^6 r^6).$$

Hence, if we suppose H with a bounded-rank and a bounded-degree, Algorithm 1 runs in $O(nm)$.

References

[1] Thomas Andreae, Michael Nölle, Gerald Schreiber, Embedding cartesian products of graphs into de bruijn graphs, J. Parallel Distrib. Comput. 46 (2) (1997) 194–200.
[2] C. Berge, Graphs, North Holland, 1987.
[3] C. Berge, Hypergraphs, North Holland, 1989.
[4] A. Bretto, Introduction to hypergraph theory and its use in engineering and image processing, Adv. in Imaging and Electron Phys. 131 (2004).
[5] A. Bretto, Hypergraphs and the helly property, Ars Combin. 78 (2006) 23–32.
[6] Alain Bretto, Yannick Silvestre, Factorization of cartesian products of hypergraphs, in: COCOON, 2010, pp. 173–181.
[7] Alain Bretto, Yannick Silvestre, Thierry Vallee, Cartesian product of hypergraphs: properties and algorithms, EPTCS 4 (2009) 22–28.
[8] Alain Bretto, Stéphane Ubéda, Janez Zerovnik, A polynomial algorithm for the strong helly property, Inf. Process. Lett. 81 (1) (2002) 55–57.
[9] Yongxi Cheng, A new class of antimagic cartesian product graphs, Discrete Math. 308 (24) (2008) 6441–6448.
[10] Mounir Hamdi, Siang W. Song, Embedding hierarchical hypercube networks into the hypercube, IEEE Trans. Parallel Distrib. Syst. Arch. 8 (9) (1997) 897–902.
[11] W. Imrich, Kartesisches produkt von mengensystemen und graphen, Studia Sci. Math. Hungar 2 (1967) 285–290.
[12] W. Imrich, I. Peterin, Recognizing cartesian products in linear time, Discrete Math. 307 (2007) 472–483.
[13] Wilfried Imrich, Sandi Klavžar, Douglas F. Rall, Topics in Graph Theory: Graphs and Their Cartesian Product, A K Peters, Wellesley, MA, 2008.
[14] L. Ostermeier, M. Hellmuth, P. F. Stadler, The cartesian product of hypergraphs, J. Graph Theory (2011). http://dx.doi.org/10.1002/jgt.20609.
[15] Iztok Peterin, Game chromatic number of cartesian product graphs, Electron. Notes Discrete Math. 29 (2007) 353–357.
[16] G. Sabidussi, Graphs with given group and given graph-theoretical properties, Canad. J. Math. 9 (1957).
[17] G. Sabidussi, Graph multiplication, Math. Z. 72 (1960).
[18] A. Vesel, Channel assignment with separation in the cartesian product of two cycles, in: Proceedings of the 24th International Conference on Information Technology Interfaces, 2002.
[19] V. G. Vizing, The cartesian product of graphs, Vychisl. Sistemy 9 (1963) 30–43.
[20] Tao-Ming Wang, Toroidal grids are anti-magic, in: LNCS, vol. 3595, 2005, pp. 671–679.
[21] Yuchen Zhang, Xiaoming Sun, The antimagicness of the cartesian product of graphs, Theoret. Comput. Sci. 410 (8–10) (2009) 727–735.