Theoretical Computer Science 596 (2015) 124–141

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

An average study of hypergraphs and their minimal transversals

Julien David [a,*], Loïck Lhote [b,*], Arnaud Mary [c,*], François Rioult [b,*]

[a] LIPN, Université Paris 13, and CNRS, UMR 7030, 99, av. J.-B. Clément, 93430 Villetaneuse, France
[b] GREYC, CNRS, ENSICAEN and Université de Caen, Caen, France
[c] Université Lyon 1, CNRS, UMR5558 LBBE, France

A R T I C L E I N F O

Article history:
Received 31 October 2014
Received in revised form 19 May 2015
Accepted 23 June 2015
Available online 8 July 2015
Communicated by P.G. Spirakis

Keywords:
Hypergraph
Minimal transversal
Enumeration
Probabilistic analysis
Average complexity
Generic complexity

A B S T R A C T

In this paper, we study some average properties of hypergraphs and the average complexity of algorithms applied to hypergraphs under different probabilistic models. Our approach is both theoretical and experimental since our goal is to obtain a random model that is able to capture the real-data complexity. Starting from a model that generalizes the Erdös–Renyi model [10,11], we obtain asymptotic estimations on the average number of transversals, irredundants and minimal transversals in a random hypergraph. We use those results to obtain an upper bound on the average complexity of algorithms to generate the minimal transversals of a hypergraph. Then we make our random model more complex in order to bring it closer to real-data and identify cases where the average number of minimal transversals is at most polynomial, quasi-polynomial or exponential.

1. Introduction

A hypergraph is a pair $\mathcal{H} = (V, \mathcal{E})$ where $V = \{1, 2, \dots, n\}$ is the set of vertices and $\mathcal{E} = (E_1, \dots, E_m)$ is the collection of hyperedges with $E_i \subseteq V$ for all i.

A *transversal* is a set of vertices that intersects all the hyperedges. A set of vertices X is said to be *irredundant* if for all vertices $i \in X$, there exists a hyperedge H such that $H \cap X = \{i\}$. X is called a *minimal transversal* when it is a transversal and none of its subset is transversal. This is equivalent to being both irredundant and a transversal.

Given a hypergraph $\mathcal{H}$, the set of all its minimal transversals forms a hypergraph called the *transversal hypergraph*.

The Transversal Hypergraph Generation problem (for short, THG-problem) consists in computing the transversal hypergraph of a given hypergraph. In the same way, the associated decision problem (in short, THD-problem) consists in deciding if a first hypergraph $\mathcal{H}_1$ is the transversal hypergraph of a second one $\mathcal{H}_2$. This problem is known to be equivalent to the famous dualization of monotone boolean functions problem (see [8]). The Transversal Hypergraph Generation problem appears in very different domains: Artificial Intelligence and Logic [6,7], Biology [2], Datamining and Machine Learning [14], mobile communications [23], *etc.* We refer to [15] for a more complete list of applications.

Since a hypergraph may have an exponential number of minimal transversals, the THG-problem does not belong to the class of polynomial problems. However, a long standing question is to decide whether there exists an algorithm to solve

* Corresponding authors.
E-mail address: Julien.David@lipn.univ-paris13.fr (J. David).

http://dx.doi.org/10.1016/j.tcs.2015.06.052
0304-3975/

THG whose running time is a polynomial on the size of the hypergraph and on the number of minimal transversal. Such an algorithm is called an *output-polynomial* time algorithm.

The complexity of the THG-problem is closely related to its associated decision problem THD. Precisely, if an output-polynomial algorithm solves THG, then THD can be solved in polynomial time. In addition, THD is clearly in the class of co-NP problems but there is no evidence of its co-NP-completeness. If THD is co-NP-complete, then no output polynomial algorithm is likely to exist for the generation problem THG (unless P = co-NP) [6].

The best known algorithm to generate the transversal hypergraph is quasi-polynomial and is due to Fredman and Khachiyan in [13]. Its running time is of the form $N^{o(\log N)}$ with N the size of the input plus the output. Nevertheless, this algorithm is not efficient for practical applications. Other algorithmic solutions were proposed and a list of them can be found in [9]. In this article, we focus on the MTMiner algorithm defined by Hébert, Bretto and Crémilleux [16]. MTMiner is closely related to the mining of the frequent patterns in data mining and is clearly output-exponential in the worst-case. We will study both average complexity and generic-case [17] output-sensitive complexity of the algorithm.

In the previous quoted results, the complexity of the THG-problem and associated algorithms were mostly studied with the worst-case point of view. Indeed, very specific entries were exhibited in order to obtain worst-case lower or upper bounds on the behavior of the algorithms. But these entries do not generally occur in practice, and the existing worst-case analyses are then not sufficient to understand the practical complexity of THG. In this article, we adopt a probabilistic point of view. Though analytic combinatorics is often used to conduct an average case study, the symbolic method [12] (a generic method used to describe the recursive decomposition of combinatorial objects) does not seem to be relevant in our case, as it cannot be used to describe the patterns we are interested in.

The study of random hypergraphs under various distributions is quite common and one of the most popular is the uniform distribution on k-uniform hypergraphs [1,5,19] (in which all hyperedges have the same cardinal k). In [22], the authors study the creation and the growth of components with a continuous random hypergraph process. Recently, De Panafieu [3] studies random non-uniform hypergraphs and their structures near the birth of the complex components. In [24], the authors prove that under the uniform distribution over all the simple hypergraphs with n vertices, the THG problem is output-polynomial with probability close to 1. In fact, under this distribution, the size of the transversal hypergraph is with high probability exponential in n and even the naive algorithm that goes through the whole search space is almost surely output-polynomial. To the best of our knowledge, this is the only study on the average complexity of the THG problem.

In this paper, we consider two random models in which the number n of vertices and the number m of hyperedges are given and suppose that m is a polynomial in n. The results we obtain are original and do not intersect with [24].

Plan of the paper: Section 2 is devoted to the probabilistic models we consider. In Section 2.1, we introduce a *single-parameter* model that generalizes the Erdös–Renyi model [10,11]. In Section 2.2, we make our random model more complex so that the probability that each vertex appears in a hyperedge is given by a function. Section 3 summarizes the asymptotic results we obtain with the single-parameter model on the average number of transversals, irredundants and minimal transversals. Section 4 is devoted to the results with the second probabilistic model. Section 5 is devoted to algorithms analysis. We study the average complexity of the MTMiner algorithm and the generic-case complexity of the THG-problem. The average complexity of MTMiner is closely related to the average number of irredundants: we obtain upper bounds on the average complexity for both models. Section 6 is devoted to experimental results. Using hypergraphs obtained from real datasets, we discuss the consistency of our random models. Section 7 contains all the proofs of the main results. Conclusion is devoted to perspectives and indications on a random model that might be interesting for a future work.

2. Probabilistic models for hypergraphs

In this paper, we study the average properties of hypergraphs under two probabilistic models. For both models we suppose that:

- The number of hyperedges m is at most polynomial in the number of vertices n, precisely $\ln m = \Theta(\ln n)$. Some of our results do not require this assumption and can therefore be extended to cases where m is exponential in n ($\ln m = \Theta(n)$). In this case, most questions we study in this paper become trivial.
- A hypergraph with n vertices and m hyperedges can be seen as a binary matrix $M(\mathcal{H}) = (m_{i,j}(\mathcal{H}))_{i=1..m, j=1..n}$. Each row in the matrix encodes a hyperedge. The value $m_{i,j}$ at line i and column j is equal to 1 if the vertex j belongs to the hyperedge encoded in row i, $m_{i,j} = 0$ otherwise.
- The variables $(m_{i,j}(\mathcal{H}))_{i=1..m, j=1..n}$ form an independent family of random variables. In other words, the event that a given vertex v appears in a given hyperedge is independent from the event that a vertex u appears in the same hyperedge or from the event that v appears in another hyperedge.

2.1. The single-parameter model

We first introduce our simplest model which is close to the Erdös–Renyi model [10,11] for graphs.

Definition 1 *(*$\mathbf{HG(n, m, p)}$ *random model).* The $HG(n, m, p)$ model supposes that the family of random variables $(m_{i,j})_{i=1..m, j=1..n}$ forms an independent and identically distributed family of random variables following the same Bernoulli law of parameter p ($0 < p < 1$) with $1 - e^{-\frac{1}{\ln n}} < p < e^{-\frac{1}{\ln n}}$.