ICAI 2023
12^{th} International Conference on Applied Informatics
Eger, Hungary, 2023

An Incremental Algorithm for Computing the Transversal Hypergraph

Laszlo Szathmary

University of Debrecen, Faculty of Informatics
H-4002 Debrecen, Pf. 400, Hungary
szathmary.laszlo@inf.unideb.hu

Abstract

In this paper we present an incremental algorithm for computing the transversal hypergraph. Our algorithm is an optimized version of Berge's algorithm [1] for solving the transversal hypergraph problem. The original algorithm of Berge is the simplest and most direct scheme for generating all minimal transversals of a hypergraph. Here we present an optimized version of Berge's algorithm that we call *BergeOpt*. We show that *BergeOpt* can significantly reduce the number of expensive inclusion tests. Experimental results show that *BergeOpt* outperforms the original algorithm of Berge.

Basic Concepts of Hypergraphs

In this subsection we mainly rely on [2]. Hypergraph theory [1] is an important field of discrete mathematics with many relevant applications in applied computer science. A hypergraph is a generalization of a graph, where edges can connect arbitrary number of vertices. Formally:

Definition 1 (hypergraph)**.** A *hypergraph* is a pair $(V,\mathcal{E})$ of a finite set $V = \{v_1, v_2, \ldots, v_n\}$ and a family $\mathcal{E}$ of subsets of V. The elements of V are called vertices, the elements of $\mathcal{E}$ edges.

Note that some authors, e.g. [1], state that the edge-set as well as each edge must be non-empty and that the union of all edges results in the vertex set.

Definition 2 (partial hypergraph)**.** Let $\mathcal{H} = \{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_m\}$ be a hypergraph. The partial hypergraph $\mathcal{H}_i$ of $\mathcal{H}$ $(i = 1, \ldots, n)$ is the hypergraph that contains the first i edges of $\mathcal{H}$, i.e. $\mathcal{H}_i = \{\mathcal{E}_1, \ldots, \mathcal{E}_i\}$.

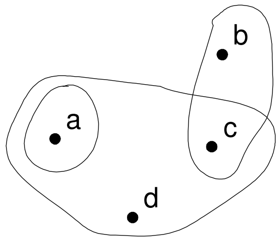

Figure 1. A sample hypergraph $\mathcal{H}$, where $V = \{a, b, c, d\}$ and $\mathcal{E} = \{\{a\}, \{b, c\}, \{a, c, d\}\}$.

A hypergraph is *simple* if none of its edges is contained in any other of its edges. Formally:

Definition 3 (simple hypergraph)**.** A hypergraph is called *simple* if it satisfies $\forall \mathcal{E}_i, \mathcal{E}_j \in \mathcal{E} : \mathcal{E}_i \subseteq \mathcal{E}_j \Rightarrow i = j$.

EXAMPLE. The hypergraph $\mathcal{H}$ in Figure 1 is not simple because the edge $\{a\}$ is contained in the edge $\{a, c, d\}$.

Definition 4. Let $\mathcal{H} = (V, \mathcal{E})$ be a hypergraph. Then $min(\mathcal{H})$ denotes the set of minimal edges of $\mathcal{H}$ w.r.t. set inclusion, i.e. $min(\mathcal{H}) = \{E \in \mathcal{E} \mid \nexists E' \in \mathcal{E} : E' \subset E\}$, and $max(\mathcal{H})$ denotes the set of maximal edges of $\mathcal{H}$ w.r.t. set inclusion, i.e. $max(\mathcal{H}) = \{E \in \mathcal{E} \mid \nexists E' \in \mathcal{E} : E' \supset E\}$.

Clearly, for any hypergraph $\mathcal{H}$, $min(\mathcal{H})$ and $max(\mathcal{H})$ are simple hypergraphs. Moreover, every partial hypergraph of a simple hypergraph is simple, too.

EXAMPLE. In the case of hypergraph $\mathcal{H}$ in Figure 1, $\min(\mathcal{H}) = \{\{a\}, \{b, c\}\}$ and $\max(\mathcal{H}) = \{\{b, c\}, \{a, c, d\}\}$.

The problem that is of high interest for us concerns hypergraph transversals. A transversal of a hypergraph $\mathcal{H}$ is a subset of the vertex set of $\mathcal{H}$ which intersects each edge of $\mathcal{H}$. A transversal is *minimal* if it does not contain any transversal as proper subset. Formally:

Definition 5 (transversal)**.** Let $\mathcal{H} = (V, \mathcal{E})$ be a hypergraph. A set $T \subseteq V$ is called a *transversal* of $\mathcal{H}$ if it meets all edges of $\mathcal{H}$, i.e. $\forall E \in \mathcal{E} : T \cap E \neq \emptyset$. A transversal T is called *minimal* if no proper subset T' of T is a transversal.

Note that Pfaltz and Jamison call transversal (resp. minimal transversal) as *blocker* (resp. *minimal blocker*) in [5]. Outside hypergraph theory, a transversal is usually called a *hitting set*.

EXAMPLE. The hypergraph $\mathcal{H}$ in Figure 1 has two minimal transversals: $\{a, b\}$ and $\{a, c\}$. For instance, the sets $\{a, b, c\}$ and $\{a, c, d\}$ are transversals but they are not minimal.

Definition 6 (transversal hypergraph)**.** The family of all minimal transversals of $\mathcal{H}$ constitutes a simple hypergraph on V called the *transversal hypergraph* of $\mathcal{H}$, which is denoted by $Tr(\mathcal{H})$.

EXAMPLE. Considering the hypergraph $\mathcal{H}$ in Figure 1, $Tr(\mathcal{H}) = \{\{a, b\}, \{a, c\}\}$.

The Algorithm of Berge

The Algorithm of Berge is the most simple and direct scheme for generating all minimal transversals of a hypergraph. The algorithm starts with the computation of $Tr(\mathcal{H}_1)$, which is a trivial case ($\mathcal{H}_1$ has one edge only, $\mathcal{E}_1$, whose minimal transversals are its vertices). Then, the algorithm adds one by one the rest of the edges, computing at each step the set of minimal transversals of the new partial hypergraph. At each step, non-minimal elements are removed. The algorithm terminates when the last edge $\mathcal{E}_n$ is added. The algorithm of Berge outputs at the end all minimal transversals of the input hypergraph $\mathcal{H}$ [1].

BergeOpt: An Optimized Version of Berge's Algorithm

In [4], Le Floc'h *et al.* presented an algorithm called *JEN* whose goal is to efficiently extract generators from a concept lattice [3] for mining exact and approximate association rules. As part of *JEN*, the aforementioned authors presented a simple algorithm without a name for calculating all the minimal transversals of a hypergraph. Our algorithm is an extended and completed version of this algorithm. In addition to [4], **(i)** we show that this algorithm is actually an optimization of Berge's original algorithm (hence the name *BergeOpt*), and **(ii)** we provide a proposition and its proof.

References

[1] C. BERGE: *Hypergraphs: Combinatorics of Finite Sets*, Amsterdam: North Holland, 1989.

[2] T. EITER, G. GOTTLOB: *Identifying the Minimal Transversals of a Hypergraph and Related Problems*, SIAM Journal on Computing 24.6 (1995), pp. 1278–1304, ISSN: 0097-5397, DOI: http://dx.doi.org/10.1137/S0097539793250299.

[3] B. GANTER, R. WILLE: *Formal concept analysis: mathematical foundations*, Berlin/Heidelberg: Springer, 1999, p. 284, ISBN: 3540627715.

[4] A. LE FLOC'H, C. FISETTE, R. MISSAOUI, P. VALTCHEV, R. GODIN: *JEN : un algorithme efficace de construction de générateurs pour l'identification des règles d'association*, Spec. num. of Revue des Nouvelles Technologies de l'Information 1.1 (2003), pp. 135–146.

[5] J. L. PFALTZ, R. E. JAMISON: *Closure Systems and their Structure*, Information Sciences 139.3–4 (2001), pp. 275–286.