Studies on hypergraphs I: hyperforests*

Lianwen Zhang

Department of Computer Science, University of British Columbia, 6356 Agriculture Road, Vancouver, B.C., Canada V6T 1Z2

Received 26 September 1989
Revised 26 February 1991

Abstract


The theory of hyperforests has many applications in fields ranging from mathematical theories, computer sciences to industrial technologies. The purpose of this paper is to systematically outline this widely scattered theory by making use of definitions and theorems found by the author himself, so as to make it into a coherent and independent mathematical theory. The paper begins with the definition of hyperforests as hypergraphs without cycles, in a sense very much similar to that of forests being graphs without cycles. Then, the concept of decomposition is introduced for hypergraphs. The following questions are raised and answered: when is a hypergraph decomposable? How? Thereafter, twig sequences are introduced, and it is proved that if a hypergraph has a twig sequence, then all its maximal intersection sequences are twig sequences. We then prove the so-called fundamental theorem, which says that a hypergraph is a hyperforest if and only if it has twig sequences. After exposing a few equivalent conditions for a hypergraph to be a hyperforest, we give a simple but important property of hyperforests, which essentially explains why the theory of hyperforests has so many applications. After that, we show the reader a way of testing if a hypergraph is a hyperforest and of finding twig sequences for a hyperforest. The paper ends with the concept of Markov trees, which provides us with a clearer picture of hypertrees-connected hyperforests, and which provides a useful mechanism for the applications of the theory.

1. Introduction

Hyperforests constitute a subclass of hypergraphs. They are more commonly known as acyclic hypergraphs in the theory of relational databases (see, for

* This paper is partly supported by a program of China National Education Committee and by the Endowment Funds of University of Kansas.
example, Maier [20]). Some statisticians also call them decomposable hypergraphs [17]. Hyperforests are closely related to chordal or triangulated graphs which originate from the studies of Gaussian elimination on sparse symmetric matrices. As a matter of fact, a hypergraph is a hyperforest if and only if it is conformal and its 2-section\(^1\) is triangulated [25].

The theory of hyperforests has a wide range of applications. In addition to the fields of relational databases and Gaussian elimination on sparse symmetric matrices mentioned above, they are also important to the theory of contingency tables [7], to nonserial dynamic programming [6], to the study of the reliability of communication systems [2], to the studies on constraint satisfaction problems [8,23,24], to probability propagation [18,23] and to belief propagation [1,16,23].

The property all the applications want of hyperforests is essentially that hyperforests have twig sequences—i.e. all the hyperedges can be so listed that the resulting sequence of hyperedges possesses the so-called running intersection property.

A theory of hyperforests should deal with the following four basic topics:

1. How to define hyperforests from hypergraphs;
2. how to prove that hypergraphs with the defining properties do have twig sequences;
3. how to test if a hypergraph is a hyperforest; and
4. how to find a twig sequence for a hyperforest.

Other topics such as how to decompose a hypergraph as far as possible and how to find an optimal hyperforest cover for a hypergraph are beyond the scope of this paper.

The most straightforward way of defining hyperforests is to say that a hyperforest is a hypergraph which has twig sequences [23]. Other definitions using properties that are close to the wanted property include Lauritzen et al.’s [17] definition of decomposable hypergraphs, Maier’s [20] definition of acyclic hypergraphs in terms of Graham deletion and Arnborg et al.’s [2] definition of k-trees.

Because those definitions employ properties which are the same as or close to the wanted property, there is, on one hand, nothing or little to do in proving the wanted property from the defining properties. On the other hand, it is very difficult by starting from these definitions to come up with a practical method of testing if a hypergraph is a hyperforest and a practical method of finding a twig sequence for a hyperforest.

A more profound definition of hyperforest is by making use of the statement we made at the end of the first paragraph. That is to define a hyperforest as a conformal hypergraph whose 2-section is triangulated [25]. It becomes harder to prove the wanted property with this definition [13]. But, one can manage, though

\(^1\) The 2-section of a hypergraph is a graph in which two vertices are neighbors if and only if they are neighbors in the original hypergraph. See Tarjan and Yannakakis [25] for the definition of conformal hypergraphs.
difficult, to come up with ways for testing if a hypergraph is a hyperforest and for finding twig sequences for hyperforests [19,22,25]. The best result in this regard is the so-called maximum cardinality search first published by Tarjan and Yannakakis [25].

In this paper, we are going to define hyperforests as hypergraphs without cycles, in a sense very much similar to that of forests being graphs without cycles in standard graph theory. This definition has its advantages over the one given by Tarjan and Yannakakis [25] because: (1) it is a straightforward generalization of the definition of forests as graphs without cycles. The concept of 2-section is avoided and the language lies completely in the category of hypergraphs. And it reveals the exact sense in which a hyperforest is acyclic. (2) The proofs are more insightful, easier to understand and mathematically more elegant (see Sections 5 and 6). (3) A new method of testing if a hypergraph is a hyperforest and of finding twig sequences for hyperforests is established along with the proof of the wanted property from the definition (Section 8). This new method may be called maximal intersection search. It improves the maximum cardinality search of Tarjan and Yannakakis in the sense that the latter requires stricter conditions to be satisfied.

The purpose of this paper is to outline the theory of hyperforests by making use of the new definition. Efforts will be made to show the simplicity and generality of the theory. The coherence and usefulness of the theory will be further exposed when we pick up the topics of decompositions and of finding hyperforest covers for hypergraphs in forthcoming papers.

2. Graphs, trees and forests

In this beginning section, we will give definitions pertaining to graphs in a way that may appear different from those one finds in standard textbooks. This is to reader’s later convenience in understanding definitions pertaining to hypergraphs.

Suppose \( V \) is a finite set. A graph \( G \) over \( V \) is a set of doublets (two-element subsets) of \( V \). The members of \( G \) are called edges. An element of \( V \) is called a vertex.
of $G$ if it is contained in at least one of the edges of $G$. The set of all the vertices of $G$ is denoted by $\bigcup G$.

Figure 1 shows two different ways of depicting the graph $G = \{\{v, x\}, \{x, w\}, \{x, y\}\}$. In Fig. 1(a), the edges are depicted by lines, which is the standard practice in graph theory textbooks. In Fig. 1(b), however, the edges are depicted by ovals, which will lead us naturally to hypergraphs.

Two vertices $v_1$ and $v_2$ of $G$ are called neighbors if $\{v_1, v_2\}$ is an edge of $G$. A sequence $v_1, v_2, \ldots, v_k$ of vertices is called a path (connecting $v_1$ and $v_k$) if $v_i$ and $v_{i+1}$ are neighbors for $i = 2$ to $k$. A graph is connected if every pair of vertices is connected by at least one path.

Suppose $V_i$ is a subset of $\bigcup G$. The restriction $G_{V_i}$ of $G$ to $V_i$ is the remaining graph of $G$ after removing all the edges which have at least one vertex outside $V_i$. In formula that is $G_{V_i} = \{e \in G \mid e \subseteq V_i\}$. For the graph $G$ in Fig. 1, when $V_i = \{u, w, x\}$, $G_{V_i} = \{\{u, x\}, \{x, w\}\}$; when $V_i = \{u, w\}$, $G_{V_i} = \emptyset$ — the empty graph.

A cycle is a connected graph in which every vertex is contained in exactly two edges. The graph $G = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_1\}\}$, for instance, is a cycle, while the graph in Fig. 1 is not.

To reveal the similarities between graphs and hypergraphs, we say that a graph induces a cycle if one of its restrictions is a cycle. Graphs which do not induce any cycles are called forests, and connected forests are called trees. The graph in Fig. 1 is a tree.

A vertex in a graph is called a leaf if it has only one neighbor. This unique neighbor is called its parent. For the graph in Fig. 1, the vertices $u$, $w$ and $y$ all are leaves, with the common parent $x$.

**Proposition 2.1.** Suppose $G$ is a tree (forest). Then

1. $G$ has at least two leaves. Hence there is always at least one leaf other than any predetermined vertex; and
2. suppose $v$ is a leaf of $G$, then $G_{-v}$ is again a tree (forest).

This property of trees (forests) is very important. In this paper, we are going to establish a similar result for hypertrees and hyperforests.

3. Hypergraphs, hypertrees and hyperforests

Suppose $V$ is a finite set. A hypergraph $H$ over $V$ is a class of nonempty non-singleton subsets of $V$. The members of $H$ are called hyperedges. An element of $V$ is called a vertex of $H$ if it is contained in at least one of the hyperedges of $H$. The set of all the vertices of $H$ is denoted by $\bigcup H$.

It is evident that graphs are hypergraphs. But a hypergraph may not be a graph. The hypergraphs in Fig. 2 are not graphs.
Two vertices $u_1$ and $u_2$ of $H$ are called neighbors if $\{u_1, u_2\}$ is contained in at least one hyperedge of $H$. A sequence $u_1, u_2, \ldots, u_k$ of vertices is called a path (connecting $u_1$ and $u_k$) if $u_{i-1}$ and $u_i$ are neighbors for $i = 2$ to $k$. A hypergraph is connected if every pair of vertices is connected by at least one path.

Suppose $V_1$ is a subset of $\bigcup H$. The restriction $H_{V_1}$ to $H$ over $V_1$ is defined by

$$H_{V_1} = \{h \cap V_1 \mid |h \cap V_1| > 1, h \in H\}.$$ (3.1)

Notice that restrictions are hypergraphs, they do not contain the empty subset nor any singletons. For the hypergraph in Fig. 2(a), if $V_1 = \{u, w, x, y\}$, then $H_{V_1} = \{\{u, x\}, \{x, w\}, \{x, y\}\}$, which is the same as the graph in Fig. 1. If $V_1 = \{t, u, w\}$, then $H_{V_1} = \{\{t, u\}\}$. When $V_1 = (\bigcup H) - \{v\}$ for some vertex $v$ of $H$, $H_{V_1}$ is denoted by $H_{-v}$.

Suppose $\mathcal{V} = \{V_1, V_2, \ldots, V_k\}$ is a partition of $\bigcup H$. For any hyperedge $h \in H$, define

$$h_{\mathcal{V}} = \{V_i \mid V_i \subseteq h, V_i \in \mathcal{V}\}.$$ (3.2)

The coarsening $H_{\mathcal{V}}$ of $H$ over $\mathcal{V}$ is a hypergraph given by

$$H_{\mathcal{V}} = \{h_{\mathcal{V}} \mid |h_{\mathcal{V}}| > 1, h \in H\}.$$ (3.3)

For the hypergraph $H$ in Fig. 2(a), if $\mathcal{V} = \{V_1, V_2, V_3, V_4\}$ where $V_1 = \{t, u\}$, $V_2 = \{x\}$, $V_3 = \{u, w\}$ and $V_4 = \{y\}$, then $H_{\mathcal{V}} = \{\{V_1, V_2\}, \{V_2, V_3\}, \{V_2, V_4\}\}$, which has the same layout as the graph in Fig. 1. We depict $H_{\mathcal{V}}$ in Fig. 3(a). For the hypergraph
$H$ in Fig. 2(b), if the coarsening $\mathcal{V} = \{V_1, V_2, V_3, V_4\}$ where $V_1 = \{u, v\}$, $V_2 = \{v\}$, $V_3 = \{w, x\}$ and $V_4 = \{y\}$, then $H_\tau = \{\{V_1, V_2\}, \{V_2, V_3\}, \{V_1, V_3, V_4\}\}$. We depict $H_\tau$ in Fig. 3(b).

A hypergraph may be a cycle in the sense of Section 2 because it may be a graph. Specifically, a hypergraph is a cycle if and only if every hyperedge consists of exactly two vertices and every vertex is contained in exactly two hyperedges. A hypergraph $H$ induces a cycle if one of the restrictions of one of the coarsenings of $H$ is a cycle. A hypergraph which does not induce any cycles is called a hyperforest, and a connected hyperforest is called a hypertree. The hypergraph in Fig. 2(a) is a hypertree, but the hypergraph in Fig. 2(b) is not a hypertree as made clear by Fig. 3(b).

**Proposition 3.1.** Any restrictions of a hyperforest are still hyperforests. In particular, if $H$ is a hyperforest and $v$ is a vertex of $v$, then $H_{-v}$ is again a hyperforest.

### 4. The decompositions of hypergraphs

Suppose $H$ and $H'$ are two hypergraphs. If $H' \subseteq H$, then we call $H'$ a subhypergraph of $H$. It is easy to see that a subhypergraph of $H$ may not be a restriction of $H$; and a restriction of $H$ may not be a subhypergraph of $H$ either.

**Definition 4.1.** Suppose $H$ is a hypergraph, and $H_1$ and $H_2$ are two nonempty subhypergraphs of $H$. If

1. $H_1 \cup H_2 = H$,
2. there are $h_1 \in H_1$ and $h_2 \in H_2$ such that
   $$ \left( \bigcup H_1 \right) \cap \left( \bigcup H_2 \right) \subseteq h_1 \cap h_2, \quad (4.1) $$
3. $\left( \bigcup H_i \right) \setminus \left( h_1 \cap h_2 \right) \neq \emptyset \quad (i = 1, 2), \quad (4.2)$

then we say that $H$ decomposes into $H_1$ and $H_2$, or into $\{H_1, H_2\}$, and that the ordered pair $\langle h_1, h_2 \rangle$ is the hinge for the decomposition.

For any hypergraph $H$, if it has two subhypergraphs $H_1$ and $H_2$ such that $H$ decomposes into $H_1$ and $H_2$, then we say that $H$ is decomposable.

The hypergraph $H$ depicted in Fig. 4 is decomposable. Actually, $H$ can be written as $H = \{\{u, w\}, \{w, x\}, \{x, u\}, \{x, y\}, \{y, u\}\}$. It is easy to verify that $H$ decomposes into $H_1 = \{\{u, w\}, \{w, x\}, \{x, u\}\}$ and $H_2 = \{\{x, u\}, \{x, y\}, \{y, u\}\}$ with the hinge being $\langle \{u, x\}, \{u, x\}\rangle$.

For a hypergraph $H$, if $\{H_1, H_2\}$ is a decomposition of $H$ and $\{H_{11}, H_{12}\}$ is in turn a decomposition of $H_1$, then $\{H_{11}, H_{12}, H_2\}$ is a decomposition of $H$. More

---

\[ \text{Footnote 4: The notion of decomposability defined here is very different from the one given by Lauritzen et al. [17]. See footnote 4 for more information.} \]
Fig. 4. An example of decomposable hypergraphs.

generally, when we say that \{H_1, H_2, \ldots, H_k\} is a decomposition of \(H\) we mean that we can reach \{H_1, H_2, \ldots, H_k\} from \(H\) by a series of binary decompositions.

A natural question here is: When is a hypergraph decomposable? To answer this question, we first notice that hypergraphs which are not connected are decomposable.

For any hypergraph \(H\), we define an equivalence relation \(\sim\) among all its vertices as follows: for any \(v, v' \in V(H)\), \(v \sim v'\) if and only if there is a path in \(H\) that connects them. The equivalence classes \(V_1, V_2, \ldots, V_k\) of \(V(H)\) under this relation constitute the connectivity partition of \(H\). The restrictions \(H_{V_i} (i = 1, 2, \ldots, k)\) are called the connected components of \(H\).

When \(H\) is not connected, the number \(k\) of connected components is larger than one. Hence \(H\) is decomposable and decomposes into \{\(H_{V_1}, H_{V_2}, \ldots, H_{V_k}\}\}. The following is devoted to the decomposability of connected hypergraphs.

Suppose \(V_0\) is a subset of \(V(H)\) for a connected hypergraph \(H\). Denote \((V(H)) - V_0\) by \(V_1\). If \(H_{V_i}\) is not connected, then \(V_0\) is a separator of \(H\). If two vertices \(v_i\) and \(v_2\) lie in different connected components of \(H_{V_i}\), then we say \(V_0\) separates \(v_i\) and \(v_2\). A subset of \(V(H)\) is complete in \(H\) if every pair of vertices in it are neighbors in \(H\). A complete separator is a separator which is complete. A minimal complete separator is a complete separator such that none of its proper subsets are separators.

Proposition 4.2. Suppose \(H\) is a connected hypergraph which decomposes into \(H_1\) and \(H_2\) with the hinge \(\langle h_1, h_2 \rangle\), then \(h_1 \cap h_2\) is a complete separator of \(H\).

Proof. Denote \(h_1 \cap h_2\) by \(d\), and denote \((V(H)) - d\) by \(V_1\). Because \(d \subseteq h_1\), \(d\) is complete. To see that \(H_{V_i}\) is not connected, we first notice that \((V(H_i)) - d \neq \emptyset (i = 1, 2)\) according to (4.2), and that \([V(H_i)) - d]\cap[(V(H_2)) - d] = \emptyset\) according to (4.1). Hence, assume \(H_{V_i}\) were connected, there would be vertices \(v_i \in [(V(H_i)) - d] (i = 1, 2)\) that are neighbors in \(H_{V_i}\), hence are neighbors in \(H\). So, there would be a hyperedge \(h\) of \(H\) that contains both \(v_1\) and \(v_2\). According to (1) of Definition 4.1, \(h\) must be either in \(H_1\) or in \(H_2\). If \(h \in H_1\), then \(v_2 \in (V(H_1)) \cap (V(H_2)) \subseteq d\). But \(v_2 \notin d\). A contradiction! A similar contradiction would occur if we assume \(h \in H_2\). So, \(h\) could not belong to either \(H_1\) or to \(H_2\). A contradiction! Therefore, \(H_{V_i}\) must be disconnected. Thus, the proposition is proved.
Another way to state this proposition is to say that a connected hypergraph is not decomposable if it has no complete separators. This leads us to the inverse question. If a connected hypergraph has a complete separator, will it be decomposable? Unfortunately, the answer is not always positive. To see an example, let us consider the hypergraph $H$ in Fig. 5. The set $\{u, x\}$ is a complete separator of $H$, yet there is no way to decompose the hypergraph. However, we have the following proposition.

**Proposition 4.3.** Suppose $d$ is a complete separator of a connected hypergraph $H$. Then the hypergraph $H' = H \cup \{d\}$ is decomposable. Denote $(\bigcup H) - d$ by $V_1$, and suppose $V_{i1}, V_{i2}, \ldots, V_{ik}$ is the connectivity partition of $V_i$ in $H_{V_i}$. Define $V_{id} = V_{i1} \cup d$ ($i = 1, 2, \ldots, k$), and $H_{id} = \{h \mid h \subseteq V_{i1}, h \in H'\}$. Then $\{H_{id}, H_{id}, \ldots, H_{id}\}$ is a decomposition of $H'$. We call it the natural decomposition of $H'$ on $d$.

**Proof.** Because (1) $\bigcup \{H_{id} \mid i = 1, 2, \ldots, k\} = H'$, (2) $d \in H_{id}$ and $d \in (\bigcup \{H_{id} \mid i = 2, \ldots, k\})$, and (3) $V_{i1} \cap (\bigcup \{V_{ii} \mid i = 2, \ldots, k\}) = d$, $H'$ decomposes into $H_{id}$ and $\bigcup \{H_{id} \mid i = 2, \ldots, k\}$. Similarly $\bigcup \{H_{id} \mid i = 2, \ldots, m\}$ decomposes into $H_{id}$ and $\bigcup \{H_{id} \mid i = 3, \ldots, k\}$. Continuing this argument, we can finally reach $\{H_{id}, H_{id}, \ldots, H_{id}\}$ by series of binary decompositions. Therefore, $\{H_{id}, H_{id}, \ldots, H_{id}\}$ is a decomposition of $H'$.

For the hypergraph $H$ in Fig. 5, if $d = \{u, x\}$, then $(\bigcup H) - d = \{w, y, z\}$. The connectivity partition of $(\bigcup H) - d$ in $H_{-d}$ is $\{V_{i1}, V_{i2}\}$, where $V_{i1} = \{w\}$ and $V_{i2} =$
So \( V_1 = \{u, w, x\} \) and \( V_2 = \{u, x, y, z\} \). Consequently \( H_1^d = \{\{u, w, x\}, \{u, x\}\} \) and \( H_2^d = \{\{u, x\}, \{x, y, z\}, \{z, u\}\} \). Therefore \( H \cup \{\{u, x\}\} \) decomposes into \( \{\{u, w, x\}, \{u, x\}\} \) and \( \{\{u, x\}, \{x, y, z\}, \{z, u\}\} \).

5. Twig sequences and maximal intersection sequences

A hyperedge \( h \) of a hypergraph \( H \) is a twig [23] if there is another hyperedge \( h^* \) which contains all the vertices that \( h \) has in common with the rest of the hypergraph, i.e.,

\[
h \cap (\bigcup_{i \neq h}(H - \{h\})) \subseteq h^*.
\]

The hyperedge \( h^* \) is called a branch to \( h \). If \( H \) is an ordinary graph, a twig is nothing but an edge containing a leaf.

For the hypergraph in Fig. 6, there are two twigs: \( \{s, t, u\} \) and \( \{w, y, z\} \). Twig \( \{s, t, u\} \) has the only branch \( \{t, u, w, x\} \); while twig \( \{w, y, z\} \) has two branches \( \{t, u, w, x\} \) and \( \{u, v, w\} \).

Let us define a listing \( h_1, h_2, \ldots, h_n \) of all the hyperedges of a hypergraph \( H \) to be a twig sequence if \( h_i \) is a twig of the subhypergraph \( \{h_1, h_2, \ldots, h_i\} \) of \( H \) for \( i \) from 2 to \( n \). Suppose \( h_j \) is a branch to \( h_i \) in \( \{h_1, h_2, \ldots, h_i\} \), then the mapping \( b: \{2, 3, \ldots, n\} \rightarrow \{1, \ldots, n - 1\} \) defined by \( b(i) = j \) is called a branching of the twig sequence.

A twig sequence \( h_1, h_2, \ldots, h_n \) for an ordinary graph \( G \) corresponds to a leaf sequence \( v_0, v_1, \ldots, v_n \) such that \( v_i \) is a leaf in the restriction of \( G \) to \( \{v_0, v_1, \ldots, v_{i-1}\} \) and that \( h_i = \{v_{i-1}, v_i\} \) for \( i = 1 \) to \( n \).

There are three hypergraphs in Fig. 7, all of them consist of three hyperedges and have the same set of vertices. Among them \( H_1 \) has six twig sequences, \( H_2 \) has four, while \( H_3 \) has none. And the hypergraph in Fig. 6 does not have any twig sequence either.

**Lemma 5.1.** Suppose \( h_1, h_2, \ldots, h_n \) is a twig sequence of a hypergraph \( H \). And suppose \( h_i \) is a branch to \( h_j \) in the subhypergraph \( H_j = \{h_1, \ldots, h_j\} \). Then moving \( h_j \) backward to any position in between \( h_i \) and \( h_j \) will result in another twig sequence of \( H \).

![Fig. 7. Hypergraphs with different number of twig sequences.](image-url)
Proof. Suppose $h_j$ is moved back to the position right after $h_i$, where $i \leq s < j$. We need only prove that for any $t, s \leq t < j, h_i$ is still a twig in the subhypergraph $H_i \triangleq \{h_1, \ldots, h_s, h_{s+1}, \ldots, h_t\}$ of $H$.

Suppose $h_j$ is a branch to $h_i$ in the subhypergraph $H_i \triangleq \{h_1, \ldots, h_i\}$. We claim that $h_i$ remains a branch to $h_i$ in $H_i'$. To verify, we need only show that $h_j \cap h_i \subseteq h_i'$. In fact, we have $h_j \cap h_i \subseteq h_i'$, because $h_i$ is a branch to $h_j$ in $H_i$ and $t < j$.

Hence $h_j \cap h_i \subseteq h_i'$. But $h_j$ is a branch to $h_i$ in $H_i$ and $i < s < t$, so we also have $h_j \cap h_i \subseteq h_i'$. Therefore we have $h_j \cap h_i \subseteq h_i'$.

Lemma 5.2. Suppose $h_1, \ldots, h_n$ is a twig sequence of a hypergraph $H$. And suppose there are integers $i$ and $j$ with $1 \leq i < j \leq n$, such that for $t$ from $i$ to $j-1$,

1. $h_j$ is a branch to $h_i$ in $H_{i,t+1} \triangleq \{h_1, \ldots, h_{t+1}\}$, and

2. $h_j \cap (h_1 \cup \cdots \cup h_{t-1}) = h_j \cap (h_1 \cup \cdots \cup h_{t-1})$.\(^3\)

Then $h_1, \ldots, h_{t-1}, h_j, h_{j-1}, \ldots, h_t, h_{j+1}, \ldots, h_n$ is also a twig sequence of $H$.

Proof. We need only prove that for any $t, i \leq t \leq j, h_i$ is a twig in $H_{i,t+1} \triangleq \{h_1, \ldots, h_{t+1}, h_{t+2}, \ldots, h_i\}$.

When $t = j$, because $H_{i,j} = \{h_1, \ldots, h_{t+1}, h_j\}$, $h_i \cap (h_1 \cup \cdots \cup h_{t-1}) = h_j \cap (h_1 \cup \cdots \cup h_{t-1})$ and $h_i$ is a twig in $\{h_1, \ldots, h_{t+1}, h_i\}$, $h_j$ is a twig in $H_{i,j}$.

Generally, we claim that $h_{i+1}$ is a branch to $h_i$ in $H_{i,j}$. To verify, we need to show that $h_j \cap h_i \subseteq h_{t+1}$ for any $s$ such that $s < i$ or $t+1 \leq s \leq j$. When $s < i$, $h_i \cap h_i \subseteq (h_1 \cup \cdots \cup h_{t-1}) \cap h_i = (h_1 \cup \cdots \cup h_{t-1}) \cap h_{t+1} \subseteq h_{t+1}$. So the claim is true in this case.

We know that $h_{j-1}$ is a branch to $h_j$ in $H_j$ and $t < j$. So, $h_j \cap h_i \subseteq h_{j-1}$. Hence $h_j \cap h_i \subseteq h_{j-1} \cap h_i$. Because $h_{j-2}$ is a branch to $h_{j-1}$ in $H_{j-1}$, if $t < j-1$, then $h_{j-1} \cap h_i \subseteq h_{j-2}$. Hence $h_j \cap h_i \subseteq h_{j-1} \cap h_i \subseteq h_{j-2} \cap h_i$. Continuing this argument, we finally can get $h_j \cap h_i \subseteq h_{j-1} \cap h_i \subseteq \cdots \subseteq h_{t+1} \cap h_i \subseteq h_{t+1}$. So, $h_i \cap h_i \subseteq h_{t+1}$ is also true for the case of $t+1 \leq s \leq j$. □

A listing $h_1, h_2, \ldots, h_n$ of all the hyperedges of a hypergraph $H$ is called a maximal intersection sequence if for any $i$ from 2 to $n$, there is no $j > i$ such that $h_i \cap (h_1 \cup \cdots \cup h_{j-1})$ is a proper subset of $h_j \cap (h_1 \cup \cdots \cup h_{j-1})$. In other words, $h_i$ has maximal intersection with $h_1 \cup \cdots \cup h_{j-1}$ among the hyperedges $h_i, h_{i+1}, \ldots, h_n$. For the hypergraph in Fig. 8, $h_1, h_2, h_3$ is a maximal intersection sequence, while $h_1, h_3, h_2$ is not.

Theorem 5.3. If a hypergraph has twig sequences, then all its maximal intersection sequences are twig sequences.

Proof. The proof can be completed by induction if we can show the following fact: if there is a twig sequence in the form of $h_1, \ldots, h_{i-1}, h_i, h_{i+1}, \ldots, h_n$, where $k_i \geq i$

\(^3\) Note: $h_0 = \emptyset$. 

L. Zhang
Fig. 8. A hypergraph illustrating the concept of intersection sequence.

To prove the fact, we first notice that if $k_1$ is $i$, then there is nothing to prove. If $k_1 > i$, we can find a branch, say $h_{k_1}$, for $h_{k_i}$ in $\{h_1, h_2, \ldots, h_{k_i}\}$. If $k_2$ is again larger than $i$, we can find a branch, say $h_{k_2}$, for $h_{k_2}$ in $\{h_1, h_2, \ldots, h_{k_2}\}$, and so on. Eventually, we can have $h_{k_1}, \ldots, h_{k_i}$, with $k_i \leq i$ and with $h_{k_{i+1}}$ a branch for $h_{k_i}$ in $H_{k_i} \supseteq \{h_1, h_2, \ldots, h_{k_i}\}$ for $s$ from 1 to $t-1$.

By Lemma 5.1, we can first move $h_{k_{i-1}}$ to the position right after $h_{i-1}$, and then move $h_{k_i}$ to the position right after $h_{k_{i-1}}$, and so on. Eventually, we can get a twig sequence in the form of $h_1, \ldots, h_{i-1}, h_{k_{i-1}}, h_{k_{i-2}}, \ldots, h_{k_1}, h_n$.

Because $k_1 > i$, and $h_{k_1}$ has maximal intersection with $h_1 \cup \cdots \cup h_{i-1}$ among $h_1, h_{i+1}, \ldots, h_n$, and because $h_{k_{i+1}}$ is a branch for $h_{k_i}$ in $H_{k_i} \supseteq \{h_1, h_2, \ldots, h_{k_i}\}$ for $s$ from 1 to $t-1$, we know that $h_{k_i} \cap (h_1 \cup \cdots \cup h_{i-1}) = h_{k_i} \cap (h_1 \cup \cdots \cup h_{i-1})$ for $s$ from 2 to $t-1$. According to Lemma 5.2, we know there is a twig sequence of $H$ which is in the form of $h_1, \ldots, h_{i-1}, h_{k_i}, h_{k_{i+1}}, \ldots, h_n$. □

The following corollaries will be useful later.

**Corollary 5.4.** If a hypergraph has twig sequences, then it has twig sequences beginning with any predetermined hyperedge.

**Corollary 5.5.** Suppose $d$ is a subset of one of the hyperedges of a hypergraph $H$. Then $H$ has twig sequences if and only if $H \cup \{d\}$ has.

**Corollary 5.6.** Suppose a hypergraph $H$ decomposes into $H_1$ and $H_2$. Then $H$ has twig sequences if and only if both $H_1$ and $H_2$ have.

### 6. Hyperforests and twig sequences

**Theorem 6.1** (the fundamental theorem). A hypergraph is a hyperforest if and only if it has twig sequences.
In the case of ordinary graphs, this theorem says that a graph is a forest if and only if it has a leaf sequence. See Proposition 2.1.

The main task in this section is to prove this important theorem. But before that, we need to introduce several lemmas.

A sequence of hyperedges \( h_1, h_2, \ldots, h_i \) of a hypergraph \( H \) is a hypercycle if there is no hyperedge in \( H \) which contains at least three of the intersections \( h_i \cap h_2, h_2 \cap h_3, \ldots, h_{i-1} \cap h_i, h_i \cap h_1 \) [20].

**Lemma 6.2** (Maier). *If a hypergraph is not a hyperforest, then it has hypercycles.*

**Lemma 6.3** (The infinite argument). *If a hypergraph has twig sequences, then it does not have hypercycles.*

**Proof.** Suppose \( h_1, h_2, \ldots, h_n \) is a twig sequence of \( H \). Assume \( H \) had a hypercycle, say \( h_{i_1}, h_{i_2}, \ldots, h_{i_j} \) with \( i_1 < i_2 < \cdots < i_j \). Hence, there would be a \( j_1 < i_1 \) such that \( h_{j_1} \) is a branch to \( h_{i_1} \) in \( \{ h_{i_1}, h_{i_2}, \ldots, h_{i_j} \} \). The definition of branch tells us that \( h_{j_1} \cap h_{i_1} \supseteq h_{j_1} \cap h_{i_1} \cap h_{i_{j_1}} \cap h_{i_{j_1}} \cap h_{i_{j_1}} \cap h_{i_{j_1}} \). Thus, \( h_{j_1}, h_{i_1}, \ldots, h_{i_{j_1}} \) is a new hypercycle of \( H \).

We rewrite the new hypercycle as \( h_{k_1}, h_{k_2}, \ldots, h_{k_k} \), where \( k_1 < k_2 < \cdots < k_j \). Then the same argument would give us yet another new hypercycle. And this process can be carried on endlessly, which means, in particular, that \( H \) is infinite, and hence that the set \( V \) over which \( H \) is defined is infinite. A contradiction! \( \Box \)

**Lemma 6.4** [13]. *If \( H \) is a hypertree, then any minimal separator of \( H \) is complete.*

**Proof.** Suppose \( d \) is a minimal separator of \( H \), and suppose \( v_1 \) and \( v_2 \) are two vertices of \( H \) separated by \( d \). Denote \((\bigcup H) - d\) by \( V_1 \), and suppose \( H_1 \) and \( H_2 \) are the connected components of \( H_{V_1} \) that contain \( v_1 \) and \( v_2 \) respectively. Since \( d \) is minimal each \( v \in d \) has at least one neighbor in both \( H_1 \) and \( H_2 \). Therefore, for any two vertices \( x \) and \( y \) in \( d \), there exist paths \( x, a_1, \ldots, a_r, y \) and \( y, b_1, \ldots, b_r, x \) with \( a_i \in (\bigcup H_1) \) and \( b_j \in (\bigcup H_2) \), such that these paths are chosen to be of the smallest length possible. If \( x \) and \( y \) were not neighbors in \( H \), then the restriction of \( H \) to \( \{ v_1, v_2, v_3 \} \) would be a cycle. This contradicts the fact that \( H \) is a hypertree. So, \( x \) and \( y \) must be neighbors. Consequently, \( d \) must be complete. \( \Box \)

**Lemma 6.5.** *Suppose \( H \) is a hypertree. If a subset \( d \) of \( \bigcup H \) is complete, then \( d \) is a subset of some hyperedge of \( H \).*

**Proof.** List all the vertices in \( d \) as \( v_1, v_2, \ldots, v_f \). There must be a hyperedge in \( H \) that contains \( \{ v_1, v_2, v_3 \} \). Because otherwise, the restriction of \( H \) to \( \{ v_1, v_2, v_3 \} \) would be a cycle. Similarly, there must be a hyperedge in \( H \) that contains \( \{ v_1, v_2, v_4 \} \). Those two results lead us to the further conclusion that there must be a hyperedge in \( H \) that contains \( \{ v_1, v_2, v_3, v_4 \} \). Because otherwise the restriction of the coarsening \( H_y \) of \( H \) to \( \{ \{ v_1, v_2 \}, \{ v_3 \}, \{ v_4 \} \} \) would be a cycle, where \( Y \) is the
partition $\mathcal{U} = \{\{v_1, v_2\} \cup \{v' \mid v' \in (\bigcup H), v' \neq v_1, v_2\}$. Continue this argument, we can eventually arrive at the conclusion that there is a hyperedge in $H$ that contains $d = \{v_1, v_2, \ldots, v_l\}$ as a subset. $\square$

**Proof of Theorem 6.1.** The *if* part follows directly from Lemmas 6.2 and 6.3.

We now set out to prove the *only if* part. Assume that any hyperforests with less than $k$ vertices have twig sequences, and that $H$ is a hyperforest with exactly $k$ vertices. If there is a hyperedge $h_0$ in $H$ that contains all the vertices of $H$, then any listings of all the hyperedges of $H$ beginning with $h_0$ are twig sequences. If no such hyperedges exist, then $H$ has at least one minimal separator, say $d$. According to Lemmas 6.4 and 6.5, $d$ is complete and is a subset of some hyperedge of $H$. By Proposition 4.3, we know that $H \cup \{d\}$ is decomposable. Hence $H \cup \{d\}$ has twig sequences because of the induction hypothesis and Corollary 5.6. Finally, by making use of Corollary 5.5 we get that $H$ has twig sequences. $\square$

7. Properties of hyperforests

In this section, we will expose explicitly the picture for hyperforests the paper has so far provided us.

If a hypergraph $H$ has a decomposition $\{H_1, H_2, \ldots, H_k\}$ such that there is a hyperedge in every $H_i$ that contains all the vertices of $H_i$, then we say that $H$ is completely decomposable$^4$.

**Proposition 7.1.** For any hypergraph $H$, the following conditions are equivalent:

1. $H$ is a hyperforest.
2. $H$ does not have any hypercycles.
3. $H$ has twig sequences.
4. $H$ is completely decomposable.

**Proof.** The equivalence of the first three conditions has been made clear in Section 6. So, it suffices to show the equivalence between (3) and (4).

Assume that any hyperforests with less than $k$ hyperedges are completely decomposable, and that $H$ is a hyperforest with exactly $k$ hyperedges. If there is a hyperedge in $H$ that contains all the vertices of $H$, then $H$ is completely decomposable by definition. Otherwise, $H$ can decompose into some $H_1$ and $H_2$, while both of them have twig sequences. By the induction hypothesis, both $H_1$ and $H_2$ are completely decomposable. Hence, $H$ is too.

On the other hand if $H$ completely decomposes into $\{H_1, H_2, \ldots, H_k\}$, by definition, each $H_i$ has twig sequences. Therefore, by repetitive uses of Corollary 5.6, we can reach the conclusion that $H$ has twig sequences. $\square$

$^4$ This concept of complete decomposability is the same as Lauritzen et al.'s [17] concept of decomposability.
In correspondence to Proposition 2.1 about forests, we have the following proposition about hyperforests.

**Proposition 7.2.** For any hyperforest $H$:

1. There are at least two twigs in $H$. Hence there is at least one twig other than any predetermined hyperedge; and
2. for any twig $h$ of $H$, $H - \{h\}$ is again a hyperforest.

**Proof.** Because $H$ is a hyperforest, it has at least one twig sequence. The last hyperedge in a twig sequence, say $h$, is a twig of $H$. According to Corollary 5.4, $H$ has at least one twig sequence beginning with $h$. The last hyperedge in this new twig sequence is again a twig of $H$, and which is different from $h$. Therefore (1) is proved.

Fact (2) is obvious if one notices that a twig sequence of $H$ ending with $h$ becomes a twig sequence of $H - \{h\}$ if $h$ is taken away from the sequence. $\square$

8. The maximal intersection search

This section is to answer the following two questions: (1) How can one tell if a hypergraph is a hyperforest? (2) How can one find a twig sequence and a branching mapping for a hyperforest?

Theorem 6.1 tells us that a hyperforest has twig sequences, and Theorem 5.3 tells us that if a hypergraph has twig sequences, then any of its maximal intersection sequences are twig sequences. So, we can tell if a hypergraph is a hyperforest by generating a maximal intersection sequence and checking if it is a twig sequence. In the case of hyperforest, this process also produces a twig sequence and a branching mapping. We call this the maximal intersection search.

**[MAXIMAL INTERSECTION SEARCH]**

**Input:** $H$ a hypergraph.

**Output:** “No” if $H$ is not a hyperforest; $\{h_1, h_2, \ldots, h_k\}$ a twig sequence of $H$ and $b$ a branching for the twig sequence if $H$ is a hyperforest.

1. Select an arbitrary hyperedge of $H$ as $h_1$.
2. Set $i = 2$, $H^* = \{h_1\}$.
3. If $H^* = H$, output $\{h_1, h_2, \ldots, h_k\}$ and the mapping $b$.
4. Else find from $H - H^*$ a hyperedge and name it $h_i$ such that $h_i$ has maximal intersection with $\bigcup H^*$ among all the hyperedges in $H - H^*$.
5. Find a hyperedge, say $h_j$, from $H^*$ such that $h_j \cap (\bigcup H^*) \subseteq h_j$.
6. If no such $h_j$ exists, output “No” and end.
7. Else set $b(i) = j$, $H^* = H^* \cup \{h_j\}$, $i = i + 1$ and go to (3).

One can easily see that the complexity of this algorithm is $O(k^2)$ (where $k$ is the number of hyperedges) in terms of hyperedge operations: union, intersection and comparison.
9. Markov trees

This section is devoted to the concept of Markov trees, which presents us with a clearer picture for hypertrees. And it also facilitates applications with a very convenient mechanism.

Suppose \( V \) is a finite set. We use \( \mathcal{P}(V) \) to denote the power set of \( V \). A graph \( \mathcal{T} \) over \( \mathcal{P}(V) \) is a graph whose vertices are subsets of \( V \).

**Definition 9.1** [23]. A tree \( \mathcal{Y} \) over \( \mathcal{P}(V) \) is called a *Markov tree* if:

1. for any vertices \( h \) and \( h' \) of \( \mathcal{Y} \), if they are neighbors in \( \mathcal{Y} \), then \( h \cap h' \neq \emptyset \), and
2. (the Markov property) if two different vertices \( h \) and \( h' \) of \( \mathcal{Y} \) contain a same \( u \in V \), then the path \( h, h_1, \ldots, h_k, h' \) in \( \mathcal{Y} \) that connects \( h \) and \( h' \) is such that \( u \in h_i \) for all \( i \).

Figure 9 shows two examples of Markov trees.

For any graph \( \mathcal{G} \) over \( \mathcal{P}(V) \), the set \( \bigcup \mathcal{G} \) of all the vertices of \( \mathcal{G} \) is a hypergraph over \( V \). For the Markov tree in Fig. 9(a), the set of the vertices is the hypergraph \( \{ \{W, X\}, \{X, Y\}, \{Y, Z\} \} \).

**Proposition 9.2** [23]. Suppose \( \mathcal{Y} \) is a Markov tree over \( \mathcal{P}(V) \), then \( \bigcup \mathcal{Y} \) is a hypertree over \( V \). Moreover, if \( h \) is a leaf of \( \mathcal{Y} \), then it is a twig of \( \bigcup \mathcal{Y} \).

**Proof.** Suppose \( h \) is a leaf of \( \mathcal{Y} \), and \( h' \) is its parent. To show that \( h \) is a twig of \( \mathcal{Y} \), we need to prove that for any other \( h^* \in (\bigcup \mathcal{Y}) \), there holds \( h \cap h^* \subseteq h' \). In fact, for any \( v \in h \cap h^* \), the path \( h, h_1, \ldots, h_k, h^* \) that connects \( h \) and \( h^* \) is such that \( v \in h_i \) for all \( i \). Because \( h' \) is the only neighbor of \( h \) in \( \mathcal{Y} \), \( h_1 \) must be \( h' \). Therefore we have \( v \in h' \).

To show that \( \bigcup \mathcal{Y} \) is a hypertree, we first notice that it is connected by (1) of Definition 9.1. The fact that \( \bigcup \mathcal{Y} \) has twig sequences can be proved by induction and by making use of Proposition 2.1 and the first part of this proposition. 

Markov trees are more commonly known as join trees in the theory of relational database literature. See, for example, Maier [20].
The Markov tree $\mathcal{T}$ in the proposition is called a Markov tree representative of the hypertree $\bigcup \mathcal{T}$. Two questions follow. Are there Markov tree representatives for all hypertrees? If yes, how can one construct a Markov tree representative for a particular hypertree? The following proposition answers those two questions.

**Proposition 9.3** [23]. Suppose $H$ is a hypertree, and $h_1, \ldots, h_n$ is a twig sequence of $H$ with a branching $b$. Then the graph over $\mathcal{P}(V)$ defined as follows is a Markov tree representative of $H$,

$$\mathcal{T} \equiv \{ \{ h_i, h_b(i) \} \mid i = 2, 3, \ldots, n \}. \quad (9.1)$$

**Proof.** The following four statements need to be proved: (1) $\bigcup \mathcal{T} = H$; (2) $\mathcal{T}$ is a tree; (3) $h_i \cap h_b(i) \neq \emptyset$ for $i = 2, 3, \ldots, n$; and (4) $\mathcal{T}$ possesses the Markov property. We will induct on $n$. And we denote the $\mathcal{T}$ for the case of $n$ by $\mathcal{T}_n$.

For the case of $n = 2$, there is nothing to prove. Suppose all the statements are true for the case of $n - 1$. Consider now the case of $n$. (1) is again obvious. (2) is true because the node in $\mathcal{T}_n$ but not in $\mathcal{T}_{n-1}$ is a leaf in $\mathcal{T}_n$. The fact that there are no cycles in $\mathcal{T}_{n-1}$ implies that there are no cycles in $\mathcal{T}_n$. For (3), it suffices to show that $h_n \cap h_b(n) \neq \emptyset$. This is evident because $h_n \cap h_b(n) \supseteq h_k \cap (\bigcup \{ H - \{ h_n \} \})$ and $H$ is connected. To prove (4), suppose $v \in h_n \cap h_k$ for some $k$. Since $h_n \cap h_k \subset h_b(n)$, we have $v \in h_b(n)$. By the induction hypothesis, $\mathcal{T}_{n-1}$ possesses the Markov property. Hence there is a path $h_k, h'_1, \ldots, h'_n, h_b(n)$ in $\mathcal{T}_{n-1}$ connecting $h_k$ and $h_b(n)$ such that

![Fig. 10. A hypertree and its three Markov tree representatives.](image)

(a) (b) (c) (d)
Proposition 9.3 shows us a way of constructing a Markov tree representative for a hypertree. For any hypertree, we can find a twig sequence with a branching mapping by using the maximal intersection search, and then use (9.1) to define a tree over \( \mathcal{P}(V) \). The resulting tree is nothing but a Markov tree representative of the hypertree we began with. Figure 10 shows a hypertree and its three possible Markov tree representatives.

To end this paper, we list the equivalent conditions for a connected hypergraph to be a hypertree.

**Proposition 9.4.** Suppose \( H \) is a connected hypergraph. Then, the following conditions are equivalent.

1. \( H \) is a hypertree.
2. \( H \) does not have hypercycles.
3. \( H \) has twig sequences.
4. \( H \) is completely decomposable.
5. \( H \) has Markov tree representatives.

**Summary**

In this paper, hyperforests are defined as hypergraphs without cycles. By manipulating twig sequences, we have proved that a hypergraph is a hyperforest if and only if it has a twig sequence. We have also proved that a maximal intersection sequence of a hyperforest is a twig sequence. The maximal intersection search algorithm is designed to tell whether a hypergraph is a hyperforest, and if yes, to find a twig sequence for it. Finally, the correspondence between hypertrees and Markov (join) trees is established.

**Acknowledgement**

This paper has benefited from talks with and help of Professor Yan Shijian, Professor Glenn Shafer and Professor Prakash Shenoy, to whom I feel very much indebted. I am also grateful to the referee who carefully read through the paper and gave me many valuable suggestions for revising the paper.

**References**

[27] L. Zhang, Three models for uncertainties in AI, Ph.D. Dissertation, Mathematics Department, Beijing Normal University, Beijing.