Computing simple-path convex hulls in hypergraphs

Francesco M. Malvestuto *, Mauro Mezzini, Marina Moscarini

Computer Science Dept., Sapienza University of Rome, Italy

ABSTRACT

In a connected hypergraph a vertex set $X$ is simple-path convex (sp-convex, for short) if either $|X| \leq 1$ or $X$ contains every vertex on every simple path between two vertices in $X$ (Faber and Jamison, 1986 [7]), and the sp-convex hull of a vertex set $X$ is the minimal superset of $X$ that is sp-convex. In this paper, we give a polynomial algorithm to compute sp-convex hulls in an arbitrary hypergraph.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

While several convexity notions exist for graphs (e.g., g-convexity [7], m-convexity [5,7], ap-convexity [4], tp-convexity [3], Steiner convexity [2,10]), fewer convexity notions have been defined explicitly for hypergraphs. The first hypergraph convexity that has been introduced is simple-path convexity (sp-convexity, for short) [7], which is a generalization of ap-convexity. Recently [8], m-convexity has been generalized to hypergraphs and another hypergraph convexity, which is stronger than m-convexity and is called c-convexity, has been introduced; moreover, efficient algorithms to compute m-convex and c-convex hulls have been given [8]. On the other hand, no result on the complexity of the problem of computing the sp-convex hull of a vertex set exists except for the case that the family of sp-convex sets is a convex geometry, in which case an efficient algorithm can be easily derived from well-known properties of totally balanced hypergraphs [1,7]. In this paper we state a characterization of sp-convex sets, which leads to solve the sp-convex hull problem in an arbitrary hypergraph in $O(n^3ms)$ time where $n$ is the number of its vertices, $m$ is the number of its edges and $s$ is the sum of the cardinalities of its edges.

The rest of the paper is organized as follows. Section 2 contains basic notions on hypergraphs and simple-path convexity. In Section 3 we present an sp-convex hull algorithm for totally balanced hypergraphs. In Section 4 we first state a characterization of sp-convex sets in an arbitrary hypergraph and, then, give our sp-convex hull algorithm.

2. Definitions

In this section we recall some hypergraph-theoretic definitions from [6].

A hypergraph is a (possibly empty) set $H$ of nonempty sets; the elements of $H$ are called the (hyper)edges of $H$ and their union the vertex set of $H$, denoted by $V(H)$. The degree of a vertex of $H$ is the number of edges containing it.

A hypergraph is trivial if it has only one edge, and non-trivial otherwise. A partial hypergraph of hypergraph $H$ is a nonempty subset of $H$.

The subhypergraph of $H$ induced by a nonempty subset $X$ of $V(H)$ is the hypergraph $(A \cap X; A \in H \setminus \{\emptyset\})$.

A path between two vertices $a$ and $b$ of $H$ is a sequence $\pi = (a_0, A_1, a_1, \ldots, A_k, a_k), k \geq 0$, where $a_0 = a$, $a_k = b$, and if $k \geq 1$ the $a_i$’s are pairwise distinct vertices of $H$.

* Corresponding author.

E-mail address: malvestuto@di.uniroma1.it (F.M. Malvestuto).
A_i's are pairwise distinct edges of \( H \), and \([a_{i-1}, a_i] \subset A_i\) for \( 1 \leq i \leq k \); by \( V(\pi) \) and \( H(\pi) \) we denote the set of vertices and edges on the path \( \pi \), respectively, that is, \( V(\pi) = \{a_0, a_1, \ldots, a_k\} \) and \( H(\pi) = \{A_1, \ldots, A_k\} \). If \( H \) is a graph (i.e., every edge has cardinality less than 3), then path \( \pi = (a_0, A_1, a_1, \ldots, A_k, a_k) \) will be written simply as \((a_0, a_1, \ldots, a_k)\) and is chordless if no two non-consecutive vertices are adjacent in \( H \).

Two vertices \( a \) and \( b \) of a hypergraph are connected if there exists a path between \( a \) and \( b \). A hypergraph is connected if every two vertices are connected. The connected components of a hypergraph are its maximal connected partial hypergraphs.

A path \( \pi \) in \( H \) is simple [7] if \(|A \cap V(\pi)| = 2\) for each edge \( A \) of \( H(\pi) \). Note that in a graph every path is simple.

**Remark 2.** Let \( \pi = (a_0, A_1, a_1, \ldots, A_k, a_k) \) be a path between \( a \) and \( b \) in \( H \). Let \( i(1) = \max\{h : a_0 \in A_h\} \). Then, \( \pi' = (a_0, A_1, a_1, \ldots, A_i, A_{i+1}, a_{i+1}, \ldots, A_k, a_k) \) is a path between \( a \) and \( b \) in \( H \). Otherwise, let \( i(2) = \max\{h : h \leq k : a_1(1) \in A_h\} \). Then, \( \pi'' = (a_0, A_1, a_1, \ldots, A_{i(2)}, A_{i(2)+1}, a_{i(2)+1}, \ldots, A_k, a_k) \) is a path between \( a \) and \( b \) in \( H \). If \( i(2) = k \) then \( \pi'' \) is a simple path between \( a \) and \( b \) in \( H \). And so on. Thus, we can construct a simple path between \( a \) and \( b \) in \( H \).

**Remark 2.** Let \( \pi = (a_0, A_1, a_1, \ldots, A_k, a_k) \) be a simple path in \( H \). If \( H(\pi) \) contains a vertex \( c \) that is not in \( V(\pi) \) and has degree 2 or more, then \( c \) is on the simple path \( \pi' = (a_0, A_1, a_1, \ldots, A_i, c, A_{i+1}, a_{i+1}, \ldots, A_k, a_k) \) where \( i = \min\{h : c \in A_h\} \) and \( i' = \max\{h : c \in A_h\} \).

A simple circuit [7] is a sequence \( (a_0, A_1, a_1, \ldots, A_{k-1}, A_k, a_k) \), \( k \geq 2 \), where \((a_0, A_1, a_1, \ldots, A_{k-1}, a_k) \) is a simple path and \( A_k \cap \{a_0, a_1, \ldots, a_{k-1}\} = \{a_0, a_k\} \); the length of the simple circuit is the number \( k \) of its edges. A hypergraph \( H \) is totally balanced if \( H \) contains no simple circuit of length greater than 2.

A vertex of a hypergraph is a nest vertex [7] (corresponding to a simple row [1] of the vertex-edge incidence matrix of \( H \)) if the edges containing it form a nested (that is, totally ordered with respect to set-inclusion) family of sets. A hypergraph is totally balanced if and only if every induced subhypergraph of \( H \) has a nest vertex [7, 17]. Based on this characterization of totally balanced hypergraphs, Anstee and Farber [1] gave a recognition algorithm for totally balanced hypergraphs, which runs in \( O(n^2m) \) time if the input hypergraph has \( n \) vertices and \( m \) edges and consists in recursively deleting nest vertices. Let \( H \) be a connected hypergraph. The sp-interval between two vertices \( a \) and \( b \) of \( H \) is the set \( I(a, b) \) which consists of every vertex on any simple path between \( a \) and \( b \). A subset \( X \) of \( V(H) \) is sp-convex if either \( X \) is empty or \( X \) contains \( I(a, b) \) for every two vertices \( X \) in \( V(H) \). The sp-convex hull of a subset \( X \) of \( V(H) \) is the minimal superset of \( X \) that is sp-convex.

Let \( X \) be an sp-convex set of \( H \). A vertex \( v \) in \( X \) is an extreme point of \( X \) if the set \( X \setminus \{v\} \) is sp-convex. The family of sp-convex sets of \( H \) is a convex geometry if every sp-convex set equals the sp-convex hull of the set of its extreme points. In [7] it was proven that this is the case if and only if \( H \) is totally balanced.

**3. Background**

A brute-force method for constructing the sp-convex hull of a vertex set \( X \subseteq V(H) \) begins by setting \( Y = X \); then, till we can no longer enlarge \( Y \), we repeatedly add to \( Y \) the set \( I(a, b) \) for every two vertices \( a \) and \( b \) in \( Y \). Unfortunately, this procedure is not efficient because, for a given value of \( Y \) it is NP-hard to compute \( I(a, b) \) for two given vertices \( a \) and \( b \) in \( Y \). To see it, let \( G(H) \) be the bipartite graph with bipartition \((V(H), H)\) where there is an arc \((a, A) \), \( a \in V(H) \) and \( A \in H \), if and only if \( a \in A \). For convenience, we call the elements of \( V(H) \) and \( H \) the vertex-nodes and edge-nodes of \( G(H) \), respectively. Note that a path in \( H \) is simple if and only if it is a chordless path in \( G(H) \), that is, no edge-node on the path is adjacent to three vertex-nodes on the path. As proven in [9], given three vertices \( a, b \) and \( c \) of a bipartite graph it is NP-complete to decide whether or not \( c \) is on a chordless path between \( a \) and \( b \). In other words, it is NP-complete to decide whether or not \( c \) belongs to \( I(a, b) \).

In the special case that \( H \) is a totally balanced hypergraph (in which case the family of sp-convex sets of \( H \) is a convex geometry), the following result easily entails the problem of computing sp-convex hulls is polynomial.

**Proposition 1.** (See Corollary 1.5.8 in [7].) Let \( H \) be a totally balanced and connected hypergraph. A subset \( X \) of \( V(H) \), is sp-convex if and only if there is an ordering \( a_1, a_2, \ldots, a_m \) of the vertices of \( V(H) \setminus X \) such that, for all \( i = 1, \ldots, m \), \( a_i \) is a nest vertex of the subhypergraph of \( H \) induced by \( X \cup \{a_1, a_1, \ldots, a_m\} \).

**Corollary 1.** Let \( H \) be a totally balanced and connected hypergraph with \( n \) vertices and \( m \) edges, and let \( X \) be a subset of \( V(H) \). The sp-convex hull of \( X \) can be constructed in \( O(n^2m) \) time.

**Proof.** By Proposition 1, the sp-convex hull of \( X \) can be obtained by repeatedly deleting the nest vertices of \( H \) that do not belong to \( X \). Therefore, the sp-convex hull problem reduces to a selective deletion of nest vertices of \( H \), which can be done in \( O(n^2m) \) time using the above-mentioned Anstee–Farber algorithm. \( \square \)

**4. Computing sp-convex hulls**

In this section we shall state a characterization of sp-convex sets which leads to a polynomial algorithm for finding the sp-convex hull of a given vertex set in an arbitrary hypergraph. To achieve this, we need the following definition.

Let \( X \) be a subset of \( V(H) \). Two edges \( A \) and \( B \) of \( H \) are connected outside \( X \) (X-connected, for short), written \( A \equiv_X B \), if

\[
\left(A \cap B\right) \setminus X \neq \emptyset \quad \text{or}
\]

\[
A = B
\]
there exists an edge \( C \) of \( H \) such that
\[(A \cap C) \setminus X \neq \emptyset \quad \text{and} \quad C \equiv_X B.\]

The edge relation \( \equiv_X \) is an equivalence relation; the classes of the resultant partition of \( H \) will be referred to as the \( X \)-connected components of \( H \), and \( H \) is \( X \)-connected if it has exactly one \( X \)-connected component. For an \( X \)-connected component \( C \) of \( H \), we call the set \( X \cap V(C) \) the boundary of \( C \). In what follows, given two distinct vertices \( a \) and \( b \) in \( X \cap V(C) \), by \( C_{a,b} \) we denote the hypergraph obtained from \( C \) by deleting the vertices in \( X \setminus \{a, b\} \) and the edges that contain both \( a \) and \( b \). Note that \( C_{a,b} \) need not contain \( a \) (or \( b \)) (see the example below).

**Theorem 1.** A vertex set \( X \) is \( sp \)-convex if and only if either \( |X| \leq 1 \) or, for every nontrivial \( X \)-connected component \( C \) of \( H \) with \( |X \cap V(C)| > 1 \) and for every two distinct vertices \( a \) and \( b \) in the boundary of \( C \), there exists no path between \( a \) and \( b \) in \( C_{a,b} \).

**Proof.** (only if) Assume that \( X \) is \( sp \)-convex. Let \( C \) be any nontrivial \( X \)-connected component of \( H \) with \( |X \cap V(C)| > 1 \), and let \( a \) and \( b \) be two distinct vertices in the boundary of \( C \). If \( a \) or \( b \) is not a vertex of \( C_{a,b} \) then trivially there exists no path between \( a \) and \( b \) in \( C_{a,b} \). Assume that both \( a \) and \( b \) are vertices of \( C_{a,b} \). By construction, if \( C_{a,b} \) and \( C \) are not adjacent in \( C_{a,b} \). Moreover, if \( a \) and \( b \) were connected in \( C_{a,b} \) then by Remark 1 there would exist a simple path \( \pi_{a,b} = (a_0, a_1, a_2, \ldots, a_k) \), \( k \geq 2 \), between \( a \) and \( b \) in \( C_{a,b} \). Therefore, there would exist a simple path \( \pi = (a_0, a_1, a_2, \ldots, a_k) \) between \( a \) and \( b \) in \( H \) where \( A_1 \) is an edge of \( C \) being the disjoint union of \( B_h \) with some subset of \( X \setminus \{a, b\} \), for all \( h \). But then one would have \( V(\pi) \setminus X \neq \emptyset \) which contradicts the hypothesis that \( X \) is \( sp \)-convex.

If (Assume that, for every nontrivial \( X \)-connected component \( C \) of \( H \) with \( |X \cap V(C)| > 1 \) and for every two distinct vertices \( a \) and \( b \) in the boundary of \( C \), there exists no path between \( a \) and \( b \) in \( C_{a,b} \). Suppose by contradiction that \( X \) is not \( sp \)-convex. Then, there would exist a simple path \( \pi \) between two vertices \( a \) and \( b \) in \( X \) such that \( V(\pi) \setminus X \neq \emptyset \). Let \( c \) be a vertex on \( \pi \) that does not belong to \( X \). Let \( u \) be the last vertex on \( \pi \) that is in \( X \) and precedes \( c \) in \( \pi \) and let \( v \) be the first vertex on \( \pi \) that is in \( X \) and follows \( c \). Then \( u, v \) and \( c \) are vertices of some nontrivial \( X \)-connected component \( C \) of \( H \); furthermore, \( u, v \) and \( c \) belong to the boundary of \( C \) and are connected in \( C_{u,v} \), which contradicts the hypothesis. \( \Box \)

**Example.** Let \( H = \{A_1, A_2, A_3, A_4, A_5\} \) where \( A_1 = \{1, 2\}, A_2 = \{1, 2, 3\}, A_3 = \{3, 4\}, A_4 = \{3, 4, 5\} \). The hypergraph \( H \) is shown in Fig. 1.

Let \( X = \{1, 3, 4\} \). The \( X \)-components of \( H \) are shown in Fig. 2 and \( C \) is the only \( X \)-component of \( H \) that is not a trivial hypergraph. The boundary of \( C \) is \( \{1, 3\} \). The hypergraph \( C_{13} \) is shown in Fig. 3.

Since \( 3 \) is not a vertex of \( C_{13} \), there exists no path joining 1 and 3 in \( C_{13} \). By Theorem 1 the set \( X \) is \( sp \)-convex, which is confirmed by the fact that the only simple paths joining two vertices in \( X \) are: \( (1, A_2, 3), (1, A_2, 3, A_3, 4), (1, A_2, 3, A_4, 4), (3, A_3, 4), (3, A_4, 4) \).

Using Theorem 1 we easily obtain a polynomial algorithm for computing the \( sp \)-convex hull of a given vertex set \( X \). However, we can speed up the construction of the \( sp \)-convex hull of \( X \) using Remark 2. Suppose that \( C \) is a nontrivial \( X \)-connected component of \( H \) and \( \pi = (a_0, A_1, a_1, A_2, \ldots, A_k, a_k) \) is a simple path between two distinct vertices \( a \) and \( b \) in the boundary of \( C \) and assume that \( C_{a,b}(\pi) \) contains a vertex \( c \) of degree 2 or more which is not in \( X \). From Remark 2 we know that another simple path between \( a \) and \( b \) in \( C_{a,b} \) is given by \( \pi' = (a_0, A_1, a_1, \ldots, A_i', c, A_i'\ldots, A_k, a_k) \) where \( i' = \min \{h < k: c \in A_h\} \) and \( h' = \max \{h < k: c \in A_h\} \). Thus, we obtain the following algorithm.

**SPCH algorithm**

**Input:** a connected hypergraph \( H \) and a subset \( X \) of \( V(H) \).

**Output:** the \( sp \)-convex hull of \( X \) in the variable \( Y \).

**begin**
\[ Y := \emptyset; \]
\[ Z := X; \]

while \( Y \neq Z \) do

**begin**
\[ Y := Z; \]

for every nontrivial \( Y \)-connected component \( C \) of \( H \) do

for every two distinct vertices \( a \) and \( b \) in the boundary of \( C \) that are connected in \( C_{a,b} \) do

find a simple path \( \pi \) between \( a \) and \( b \) in \( C_{a,b} \);

add to \( Y \) the vertices of \( C_{a,b}(\pi) \) with degree 2 or more

end

end

end

**end**

We will evaluate the complexity of the SPCH algorithm in terms of the number \( n \) of vertices of \( H \), of the number \( m \) of edges of \( H \) and of the size \( s = \sum_{A \in H} |A| \) of \( H \).
We make use of the bipartite graph $G(H)$ to represent $H$. Thus, $G(H)$ is connected and has $m + n$ nodes and $s$ arcs.

For a given value of $Y$, we mark the vertex-nodes of $G(H)$ that belong to $Y$. Then, we can construct the $Y$-connected components of $H$ with their boundaries in $O(s)$ time and their number is $O(m)$. For a given $Y$-connected component $C$ of $H$ there exist $O(n^2)$ pair of vertices in the boundary of $C$. Let $\{a, b\}$ be a pair of vertices in the boundary of $C$. In the bipartite graph $G(C)$ we unmark $a$ and $b$ and we mark the edge-nodes adjacent to both $a$ and $b$. Thus, we can construct $G(C_{a,b})$ by ignoring the marked nodes of $G(C)$ and, if $a$ and $b$ are connected in $G(C_{a,b})$, in $O(s)$ time we can construct a shortest path $\pi$ between $a$ and $b$ in $G(C_{a,b})$ and find the set $Y'$ of vertex-nodes of $G(C_{a,b}(\pi))$ with degree 2 or more. Note that $\pi$ is also a chordless path in $G(C_{a,b})$ and, hence, a simple path between $a$ and $b$ in $C_{a,b}$. Finally, we can add $Y'$ to $Z$ in $O(n)$ time. Therefore, since $n < s$, processing a given value of $Y$ requires $O(n^2ms)$ time. Since $Y$ can assume $O(n)$ distinct values the complexity of the SPCH algorithm is $O(n^3ms)$.

References