CONVEX HULLS IN ARBITRARY DIMENSION

Vera Sacristán

Discrete and Algorithmic Geometry
Facultat de Matemàtiques i Estadística
Universitat Politècnica de Catalunya

CONVEX HULLS IN ARBITRARY DIMENSION

Let S be a set of points in E^d, a real Euclidean space of dimension d.

CONVEX HULLS IN ARBITRARY DIMENSION

Let S be a set of points in E^d, a real Euclidean space of dimension d.

A **linear combination** of points in S is $\sum_{i=1}^{k} \lambda_i p_i, \text{ where } k \in \mathbb{N},\ \lambda_i \in \mathbb{R},\ p_i \in S$.

CONVEX HULLS IN ARBITRARY DIMENSION

Let S be a set of points in E^d, a real Euclidean space of dimension d.

A **linear combination** of points in S is $\sum_{i=1}^{k} \lambda_i p_i, \text{ where } k \in \mathbb{N},\ \lambda_i \in \mathbb{R},\ p_i \in S.$

An **affine combination** of points in S is a linear combination such that $\sum_{i=1}^{k} \lambda_i = 1.$

- The set of all affine combinations of S is called **affine hull** of S.
- The affine hull of two points p and q is the line through them.
- $k+1$ points are called **affinely independent** if their affine hull is an affine space of dimension k.

CONVEX HULLS IN ARBITRARY DIMENSION

Let S be a set of points in E^d, a real Euclidean space of dimension d.

A **linear combination** of points in S is $\sum_{i=1}^{k} \lambda_i p_i, \text{ where } k \in \mathbb{N}, \lambda_i \in \mathbb{R}, p_i \in S$.

An **affine combination** of points in S is a linear combination such that $\sum_{i=1}^{k} \lambda_i = 1$.

- The set of all affine combinations of S is called **affine hull** of S.
- The affine hull of two points p and q is the line through them.
- $k+1$ points are called **affinely independent** if their affine hull is an affine space of dimension k.

A **convex combination** of points in S is an affine combination such that $\lambda_i \geq 0 \; \forall \, i$.

- The set of all convex combinations of S is called **convex hull** of S.
- The convex hull of two affinely independent points p and q is the line segment pq.
- The convex hull of three affinely independent points p, q and r is the triangle pqr.
- The convex hull of four affinely independent points p, q, r and s is the tetrahedron $pqrs$.

CONVEX HULLS IN ARBITRARY DIMENSION

Let S be a set of points in E^d, a real Euclidean space of dimension d.

A **linear combination** of points in S is $\sum_{i=1}^{k} \lambda_i p_i, \text{ where } k \in \mathbb{N},\ \lambda_i \in \mathbb{R},\ p_i \in S$.

An **affine combination** of points in S is a linear combination such that $\sum_{i=1}^{k} \lambda_i = 1$.

- The set of all affine combinations of S is called **affine hull** of S.
- The affine hull of two points p and q is the line through them.
- $k+1$ points are called **affinely independent** if their affine hull is an affine space of dimension k.

A **convex combination** of points in S is an affine combination such that $\lambda_i \geq 0 \; \forall \, i$.

- The set of all convex combinations of S is called **convex hull** of S.
- The convex hull of two affinely independent points p and q is the line segment pq.
- The convex hull of three affinely independent points p, q and r is the triangle pqr.
- The convex hull of four affinely independent points p, q, r and s is the tetrahedron $pqrs$.

A set is called **convex** if it is stable under convex combinations.

CONVEX HULLS IN ARBITRARY DIMENSION

Proposition: A set S is convex iff it contains segment pq for all $p, q \in S$.

Proof: If S is convex, it is stable under convex combinations and, in particular, it contains all segments with endpoints in S. The reciprocal is proved by induction on the number of points k of the convex combination. For $k = 2$, we have the hypothesis. Consider now a convex combination with $k + 1$ points. We have

$$\sum_{i=0}^{k} \lambda_i p_i = \lambda_0 p_0 + \sum_{i=1}^{k} \lambda_i p_i = \lambda_0 p_0 + (1 - \lambda_0) \sum_{i=1}^{k} \frac{\lambda_i}{\lambda_0} p_i \in S.$$

CONVEX HULLS IN ARBITRARY DIMENSION

Proposition: A set S is convex iff it contains segment pq for all $p, q \in S$.

Proof: If S is convex, it is stable under convex combinations and, in particular, it contains all segments with endpoints in S. The reciprocal is proved by induction on the number of points k of the convex combination. For $k = 2$, we have the hypothesis. Consider now a convex combination with $k + 1$ points. We have

$$\sum_{i=0}^{k} \lambda_i p_i = \lambda_0 p_0 + \sum_{i=1}^{k} \lambda_i p_i = \lambda_0 p_0 + (1 - \lambda_0) \sum_{i=1}^{k} \frac{\lambda_i}{\lambda_0} p_i \in S.$$

Corollary: The intersection of convex sets is convex.

Corollary: The convex hull $ch(S)$ of a set S is the smallest convex set containing S.

Corollary: The convex hull $ch(S)$ of a set S is the intersection of all the convex sets containing S.

Proposition: A set S is convex iff it contains segment pq for all $p, q \in S$.

Proof: If S is convex, it is stable under convex combinations and, in particular, it contains all segments with endpoints in S. The reciprocal is proved by induction on the number of points k of the convex combination. For $k = 2$, we have the hypothesis. Consider now a convex combination with $k + 1$ points. We have

$$\sum_{i=0}^{k} \lambda_i p_i = \lambda_0 p_0 + \sum_{i=1}^{k} \lambda_i p_i = \lambda_0 p_0 + (1 - \lambda_0) \sum_{i=1}^{k} \frac{\lambda_i}{\lambda_0} p_i \in S.$$

Corollary: The intersection of convex sets is convex.

Corollary: The convex hull $ch(S)$ of a set S is the smallest convex set containing S.

Corollary: The convex hull $ch(S)$ of a set S is the intersection of all the convex sets containing S.

The **dimension** of a convex set is defined as the dimension of its affine hull.

POLYTOPES

POLYTOPES

- The convex hull of a finite set of points is called **polytope**.
- A polytope of dimension k is called k-**polytope**.
- The convex hull of $k+1$ affinely independent points is a k-polytope called k-**simplex**.

CONVEX HULLS IN ARBITRARY DIMENSION

POLYTOPES

- The convex hull of a finite set of points is called **polytope**.
- A polytope of dimension k is called k-**polytope**.
- The convex hull of $k+1$ affinely independent points is a k-polytope called k-**simplex**.

Proposition: Polytopes are closed and bounded.

POLYTOPES

- The convex hull of a finite set of points is called **polytope**.
- A polytope of dimension k is called **k-polytope**.
- The convex hull of $k+1$ affinely independent points is a k-polytope called **k-simplex**.

Proposition: Polytopes are closed and bounded.

Let P be a polytope and H be a hyperplane in E^d:

- H **supports** P if $P \cap H \neq \emptyset$ and $(P \subset \overline{H^+}$ or $P \subset \overline{H^-})$.
- If H supports P, then we call $P \cap H$ a **face** of P.
- The 0-faces are called **vertices** of P.
- The 1-faces are called **edges** of P.
- The $(d-1)$-faces are called **facets** of P.

CONVEX HULLS IN ARBITRARY DIMENSION

POLYTOPES

- The convex hull of a finite set of points is called **polytope**.
- A polytope of dimension k is called k-**polytope**.
- The convex hull of $k+1$ affinely independent points is a k-polytope called k-**simplex**.

Proposition: Polytopes are closed and bounded.

Let P be a polytope and H be a hyperplane in E^d:

- H **supports** P if $P \cap H \neq \emptyset$ and ($P \subset \overline{H^+}$ or $P \subset \overline{H^-}$).
- If H supports P, then we call $P \cap H$ a **face** of P.
- The 0-faces are called **vertices** of P.
- The 1-faces are called **edges** of P.
- The $(d-1)$-faces are called **facets** of P.

Incidence

Two faces are called **incident** if one of them is a subset of the other one.

Adjacency

- Two vertices are **adjacent** if they are incident to the same edge.
- Two facets are **adjacent** if they are incident to the same $(d-1)$-face.

PROPERTIES OF POLYTOPES

PROPERTIES OF POLYTOPES

1. The boundary of a polytope is the union of all its faces.
2. Every polytope has a finite number of faces, and each one of them is a polytope.
3. Every polytope is the convex hull of its vertices.
4. Every polytope is the intersection of a finite set of closed halfspaces, namely, one for each of its $(d-1)$-faces.
5. The intersection of a finite number of closed halfspaces, if bounded, is a polytope.
6. Every face of a polytope P is a face of a $(d-1)$-face of P. Reciprocally, every face of a face of P is a face of P.
7. If P is a polytope, then
   (a) The intersection of any family of faces of P is a face of P.
   (b) Every $(d-2)$-face of P is the intersection of two $(d-1)$ faces of P.
   (c) If $j, k \in \mathbb{N}$, and $0 \leq j \leq k < d$, every j-face is the intersection of all the k-faces containining it.

COMBINATORICS OF POLYTOPES

COMBINATORICS OF POLYTOPES

Let P be a d-polytope.
Let n_k be its number of k-faces, for $-1 \leq k \leq d$ (the (-1)-face being $\emptyset$ and the d-face being P itself).

Euler's relation

$$\sum_{k=0}^{d-1} (-1)^k n_k = 1 - (-1)^d \qquad \text{or, equivalently,} \qquad \sum_{k=-1}^{d} (-1)^k n_k = 0.$$

COMBINATORICS OF POLYTOPES

Let P be a d-polytope.
Let n_k be its number of k-faces, for $-1 \leq k \leq d$ (the (-1)-face being $\emptyset$ and the d-face being P itself).

Euler's relation

$$\sum_{k=0}^{d-1}(-1)^k n_k = 1-(-1)^d \qquad \text{or, equivalently,} \qquad \sum_{k=-1}^{d}(-1)^k n_k = 0.$$

The Dehn-Sommerville relations

If P is simple (i.e., each of its vertices belongs exactly to d facets), then $\sum_{j=0}^{k}(-1)^j \binom{d-j}{d-k} n_j = n_k$.

CONVEX HULLS IN ARBITRARY DIMENSION

COMBINATORICS OF POLYTOPES

Let P be a d-polytope.
Let n_k be its number of k-faces, for $-1 \leq k \leq d$ (the (-1)-face being $\emptyset$ and the d-face being P itself).

Euler's relation

$$\sum_{k=0}^{d-1}(-1)^k n_k = 1-(-1)^d \qquad \text{or, equivalently,} \qquad \sum_{k=-1}^{d}(-1)^k n_k = 0.$$

The Dehn-Sommerville relations

If P is simple (i.e., each of its vertices belongs exactly to d facets), then $\sum_{j=0}^{k}(-1)^j \binom{d-j}{d-k} n_j = n_k$.

The upper bound theorem

Any d-polytope with n vertices (or n facets) has at most $O(n^{\lfloor d/2 \rfloor})$ faces of all dimensions and $O(n^{\lfloor d/2 \rfloor})$ pairs of incident faces of all dimensions.

CONVEX HULLS IN ARBITRARY DIMENSION

COMBINATORICS OF POLYTOPES

Let P be a d-polytope.
Let n_k be its number of k-faces, for $-1 \leq k \leq d$ (the (-1)-face being $\emptyset$ and the d-face being P itself).

Euler's relation

$$\sum_{k=0}^{d-1}(-1)^k n_k = 1-(-1)^d \qquad \text{or, equivalently,} \qquad \sum_{k=-1}^{d}(-1)^k n_k = 0.$$

The Dehn-Sommerville relations

If P is simple (i.e., each of its vertices belongs exactly to d facets), then $\sum_{j=0}^{k}(-1)^j \binom{d-j}{d-k} n_j = n_k$.

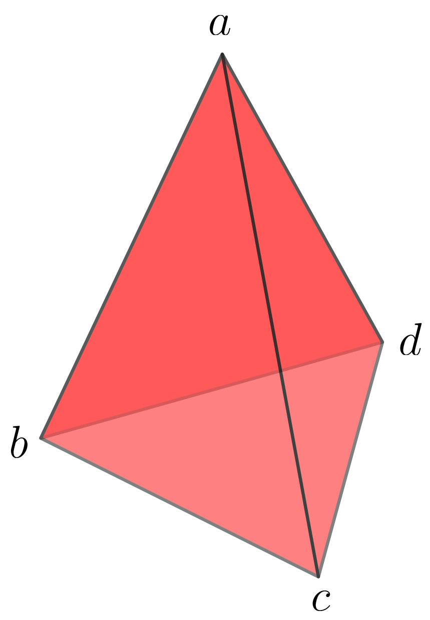
The upper bound theorem

Any d-polytope with n vertices (or n facets) has at most $O(n^{\lfloor d/2 \rfloor})$ faces of all dimensions and $O(n^{\lfloor d/2 \rfloor})$ pairs of incident faces of all dimensions.

Tightness of the bound

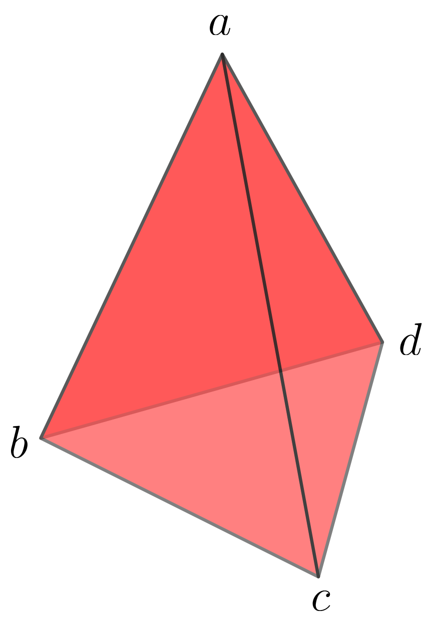
There exists a d-polytope with $\Omega(n^{\lfloor d/2 \rfloor})$ faces of all dimensions and $\Omega(n^{\lfloor d/2 \rfloor})$ pairs of incident faces of all dimensions.

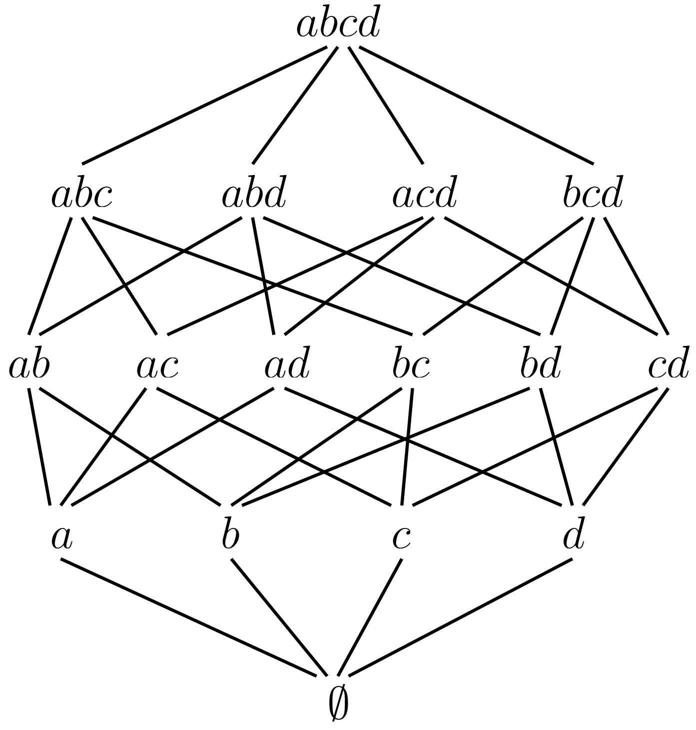
STORING A POLYTOPE



STORING A POLYTOPE

The **incidence graph** stores one node for each face and an edge for each pair of incident faces. The space used is $O(n^{\lfloor d/2 \rfloor})$.

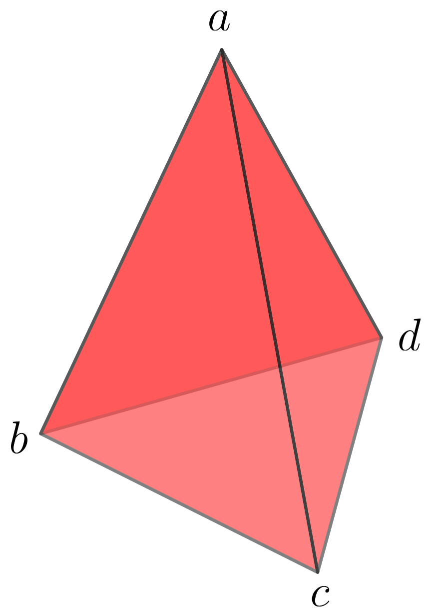


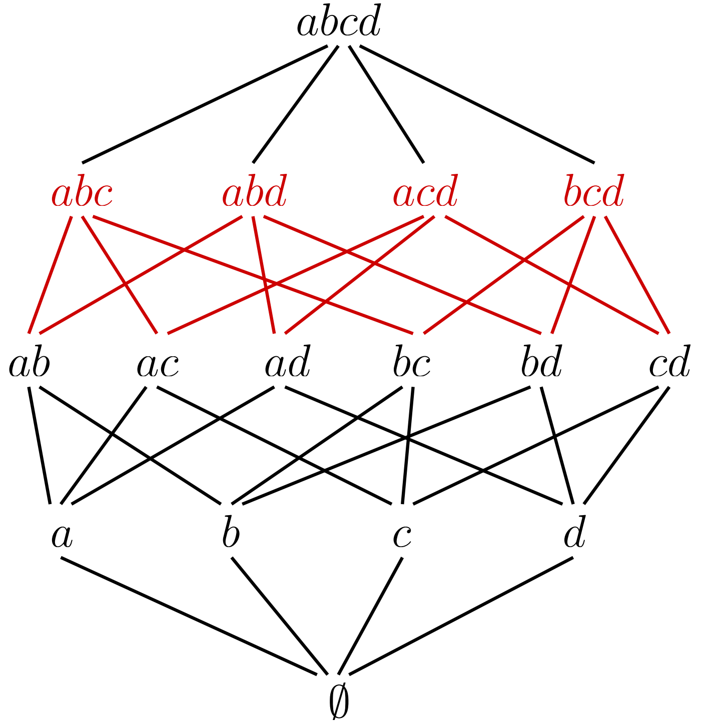


STORING A POLYTOPE

The **incidence graph** stores one node for each face and an edge for each pair of incident faces.
The space used is $O(n^{\lfloor d/2 \rfloor})$.

The incidence graph also encodes the **adjacency graph**, which has a node for each facet and an arc for each pair of adjacent facets: the arcs of the adjacency graph are in one-to-one correspondance with the $(d-2)$-faces.





COMPUTING d-DIMENSIONAL CONVEX HULLS

COMPUTING d-DIMENSIONAL CONVEX HULLS

Lower bound: Computing the convex hull of n points in E^d is $\Omega(n \log n + n^{\lfloor d/2 \rfloor})$.

Proof: If $d \geq 4$, then $\Omega(n \log n + n^{\lfloor d/2 \rfloor}) = \Omega(n^{\lfloor d/2 \rfloor})$, which is the size of the output. If $d = 2, 3$, then $\Omega(n \log n + n^{\lfloor d/2 \rfloor}) = \Omega(n \log n)$, which we know is a lower bound for the problem in dimension 2.

COMPUTING d-DIMENSIONAL CONVEX HULLS

Lower bound: Computing the convex hull of n points in E^d is $\Omega(n \log n + n^{\lfloor d/2 \rfloor})$.

Proof: If $d \geq 4$, then $\Omega(n \log n + n^{\lfloor d/2 \rfloor}) = \Omega(n^{\lfloor d/2 \rfloor})$, which is the size of the output. If $d = 2, 3$, then $\Omega(n \log n + n^{\lfloor d/2 \rfloor}) = \Omega(n \log n)$, which we know is a lower bound for the problem in dimension 2.

Notation

- Denote by H_F the supporting hyperplane of each facet F of a given polytope. Among the two closed subspaces determined by H_F, let $\overline{H_F^+}$ be the one containing the polytope, and $\overline{H_F^-}$ the one not containing the polytope.
- Let p be a point exterior to a convex polytope C, and suppose that p does not belong to any hyperplane supporting a facet of C. Then:
  - Facets F such that $p \notin \overline{H_F^+}$ are called red.
  - Facets F such that $p \in \overline{H_F^+}$ are called blue.
  - The color of any remaining face is the intersection (red, blue or purple) of the colors of the facets incident to it.

CONVEX HULLS IN ARBITRARY DIMENSION

COMPUTING d-DIMENSIONAL CONVEX HULLS

Let C be a convex polytope and p a point in general position with respect to C.

Lemma 1: Every face of $ch(C \cup \{p\})$ is either a blue or purple face of C or the convex hull $ch(G \cup \{p\})$ of p and a purple face G of C.

COMPUTING d-DIMENSIONAL CONVEX HULLS

Let C be a convex polytope and p a point in general position with respect to C.

Lemma 1: Every face of $ch(C \cup \{p\})$ is either a blue or purple face of C or the convex hull $ch(G \cup \{p\})$ of p and a purple face G of C.

Proof: If P belongs to C, every face is blue and the result follows.
Otherwise:

- Every blue facet F of C is a facet of $ch(C \cup \{p\})$ because H_F supports $ch(C \cup \{p\})$ and $H_F \cap ch(C \cup \{p\}) = F$. The remaining blue faces being the intersection of blue facets, the result holds for all blue faces.
- If G is a purple face of C, then G must belong at least to one red facet F_1 and one blue facet F_2. Then $p \in H^+_{F_2}$ and $H_{F_2} \cap ch(C \cup \{p\}) = G$. Therefore, G is a face of $ch(C \cup \{p\})$. Since $p \in H^-_{F_2}$, any hyperplane H rotating about $H_{F_1} \cap H_{F_2}$, will eventually hit p. Then H supports $ch(C \cup \{p\})$ and $ch(C \cup \{p\}) \cap H \supset ch(G \cup \{p\})$. Therefore, $ch(G \cup \{p\})$ is a face of $ch(C \cup \{p\})$.
- Any face of $ch(G \cup \{p\})$ that does not contain p must be a blue or purple face of C. Any face of $ch(C \cup \{p\})$ that contains p must be of the form $ch(G \cup \{p\})$, for some face of C that must be purple. (Note: in particular, p itself is a face of $ch(C \cup \{p\})$ because the empty face of C must be purple.)

COMPUTING d-DIMENSIONAL CONVEX HULLS

Let C be a convex polytope and p a point in general position with respect to C.

Lemma 1: Every face of $ch(C \cup \{p\})$ is either a blue or purple face of C or the convex hull $ch(G \cup \{p\})$ of p and a purple face G of C.

Lemma 2: Incidences in $ch(C \cup \{p\})$ are:

– All blue–blue, blue–purple and purple–purple incidences from C.

– G and $ch(G \cup \{p\})$, for all purple G in C.

– $ch(F \cup \{p\})$ and $ch(G \cup \{p\})$, for all purple F and G incident in C.

COMPUTING d-DIMENSIONAL CONVEX HULLS

Let C be a convex polytope and p a point in general position with respect to C.

Lemma 1: Every face of $ch(C \cup \{p\})$ is either a blue or purple face of C or the convex hull $ch(G \cup \{p\})$ of p and a purple face G of C.

Lemma 2: Incidences in $ch(C \cup \{p\})$ are:

- All blue–blue, blue–purple and purple–purple incidences from C.
- G and $ch(G \cup \{p\})$, for all purple G in C.
- $ch(F \cup \{p\})$ and $ch(G \cup \{p\})$, for all purple F and G incident in C.

Lemma 3: The set of red facets is connected. The set of blue facets is connected too.

Proof: If $d = 2$, this is a well known fact. Otherwise, let r_1 and r_2 be two points in E^d on two different red (blue) facets of C. The plane π spanned by p, r_1, r_2 intersects C in a 2-polytope, where the set of all red (blue) edges is connected. Therefore, there exists a red (blue) path connecting r_1 and r_1 in $\pi \cap C \subset C$.

COMPUTING d-DIMENSIONAL CONVEX HULLS

Let C be a convex polytope and p a point in general position with respect to C.

Lemma 1: Every face of $ch(C \cup \{p\})$ is either a blue or purple face of C or the convex hull $ch(G \cup \{p\})$ of p and a purple face G of C.

Lemma 2: Incidences in $ch(C \cup \{p\})$ are:

- All blue–blue, blue–purple and purple–purple incidences from C.
- G and $ch(G \cup \{p\})$, for all purple G in C.
- $ch(F \cup \{p\})$ and $ch(G \cup \{p\})$, for all purple F and G incident in C.

Lemma 3: The set of red facets is connected. The set of blue facets is connected too.

Lemma 4: The set of purple faces is isomorphic to a $(d-1)$-politope of at most n vertices.

Proof: Any hyperplane H separating p from C intersects all the faces of $ch(C \cup \{p\})$ containing p (except for the vertex p) and those faces only. The trace in H of these faces is a $(d-1)$-polytope whose incidences correspond to incidences in C.

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \dots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \dots, p_n\})$

CONVEX HULLS IN ARBITRARY DIMENSION

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \ldots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \ldots, p_n\})$

Algorithm

1. Lexicographically sort the points.
2. Initialize $C_{d+1} = simplex\,(p_1, \ldots, p_{d+1}) = ch(\{p_1, \ldots, p_{d+1}\})$
3. Construct $C_{i+1} = ch(\{p_1, \ldots, p_{i+1}\})$ from $C_i = ch(\{p_1, \ldots, p_i\})$.
    1. Identify a red facet of C_i as seen from p_{i+1}.
    2. Construct three lists, respectively containing all red facets, all $(d-2)$ red faces, and all $(d-2)$ purple faces.
    3. Construct two more lists, respectively containing all remaining red faces, and all remaining purple faces.
    4. Update the incidence graph.

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \dots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \dots, p_n\})$

Algorithm

1. Lexicographically sort the points.
2. Initialize $C_{d+1} = simplex\,(p_1, \dots, p_{d+1}) = ch(\{p_1, \dots, p_{d+1}\})$
3. Construct $C_{i+1} = ch(\{p_1, \dots, p_{i+1}\})$ from $C_i = ch(\{p_1, \dots, p_i\})$.
    1. Identify a red facet of C_i as seen from p_{i+1}.

       Explore all facets incident to p_i (the last inserted point). Due to the lexicographical order of the points, at least one of them needs to be red.

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \dots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \dots, p_n\})$

Algorithm

1. Lexicographically sort the points.
2. Initialize $C_{d+1} = simplex\,(p_1, \dots, p_{d+1}) = ch(\{p_1, \dots, p_{d+1}\})$
3. Construct $C_{i+1} = ch(\{p_1, \dots, p_{i+1}\})$ from $C_i = ch(\{p_1, \dots, p_i\})$.
    1. Identify a red facet of C_i as seen from p_{i+1}.
    2. Construct three lists, respectively containing all red facets, all $(d-2)$ red faces, and all $(d-2)$ purple faces.

       A deptf-first search allows to find all red facets and classify all $(d-2)$-faces into red and purple.

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \ldots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \ldots, p_n\})$

Algorithm

1. Lexicographically sort the points.
2. Initialize $C_{d+1} = simplex\,(p_1, \ldots, p_{d+1}) = ch(\{p_1, \ldots, p_{d+1}\})$
3. Construct $C_{i+1} = ch(\{p_1, \ldots, p_{i+1}\})$ from $C_i = ch(\{p_1, \ldots, p_i\})$.
    1. Identify a red facet of C_i as seen from p_{i+1}.
    2. Construct three lists, respectively containing all red facets, all $(d-2)$ red faces, and all $(d-2)$ purple faces.
    3. Construct two more lists, respectively containing all remaining red faces, and all remaining purple faces.

    Once all red and purple faces have been classified from dimension $d-1$ down to dimension $k+1$, all k-subfaces of purple $(k+1)$-faces are declared purple. Once this done, all unclassified k-subfaces of red $(k+1)$-faces are declared red.

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \ldots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \ldots, p_n\})$

Algorithm

1. Lexicographically sort the points.
2. Initialize $C_{d+1} = simplex\,(p_1, \ldots, p_{d+1}) = ch(\{p_1, \ldots, p_{d+1}\})$
3. Construct $C_{i+1} = ch(\{p_1, \ldots, p_{i+1}\})$ from $C_i = ch(\{p_1, \ldots, p_i\})$.
    1. Identify a red facet of C_i as seen from p_{i+1}.
    2. Construct three lists, respectively containing all red facets, all $(d-2)$ red faces, and all $(d-2)$ purple faces.
    3. Construct two more lists, respectively containing all remaining red faces, and all remaining purple faces.
    4. Update the incidence graph.

       Fist, all red faces are eliminated from the incidence graph (both nodes and incident arcs). Then, starting from $k = 0$, if F is a purple k-face a new node is created for the $(k+1)$-face $ch(F \cup \{p\})$. This new node is connected to F and also to all the k-faces of the form $ch(G \cup \{p\})$, where G is a $(k-1)$-subface of F.

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \ldots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \ldots, p_n\})$

Algorithm

1. Lexicographically sort the points.
2. Initialize $C_{d+1} = simplex\,(p_1, \ldots, p_{d+1}) = ch(\{p_1, \ldots, p_{d+1}\})$
3. Construct $C_{i+1} = ch(\{p_1, \ldots, p_{i+1}\})$ from $C_i = ch(\{p_1, \ldots, p_i\})$.
    1. Identify a red facet of C_i as seen from p_{i+1}.
    2. Construct three lists, respectively containing all red facets, all $(d-2)$ red faces, and all $(d-2)$ purple faces.
    3. Construct two more lists, respectively containing all remaining red faces, and all remaining purple faces.
    4. Update the incidence graph.

Analysis

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \ldots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \ldots, p_n\})$

Algorithm

1. Lexicographically sort the points.
2. Initialize $C_{d+1} = simplex\,(p_1, \ldots, p_{d+1}) = ch(\{p_1, \ldots, p_{d+1}\})$
3. Construct $C_{i+1} = ch(\{p_1, \ldots, p_{i+1}\})$ from $C_i = ch(\{p_1, \ldots, p_i\})$.
    1. Identify a red facet of C_i as seen from p_{i+1}.
    2. Construct three lists, respectively containing all red facets, all $(d-2)$ red faces, and all $(d-2)$ purple faces.
    3. Construct two more lists, respectively containing all remaining red faces, and all remaining purple faces.
    4. Update the incidence graph.

Analysis

$O(n \log n)$

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \ldots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \ldots, p_n\})$

Algorithm	Analysis
1. Lexicographically sort the points.	$O(n \log n)$
2. Initialize $C_{d+1} = simplex\,(p_1, \ldots, p_{d+1}) = ch(\{p_1, \ldots, p_{d+1}\})$	$O(1)$
3. Construct $C_{i+1} = ch(\{p_1, \ldots, p_{i+1}\})$ from $C_i = ch(\{p_1, \ldots, p_i\})$.	

1. Identify a red facet of C_i as seen from p_{i+1}.
2. Construct three lists, respectively containing all red facets, all $(d-2)$ red faces, and all $(d-2)$ purple faces.
3. Construct two more lists, respectively containing all remaining red faces, and all remaining purple faces.
4. Update the incidence graph.

CONVEX HULLS IN ARBITRARY DIMENSION

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \ldots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \ldots, p_n\})$

Algorithm	**Analysis**
1. Lexicographically sort the points.	$O(n \log n)$
2. Initialize $C_{d+1} = simplex\,(p_1, \ldots, p_{d+1}) = ch(\{p_1, \ldots, p_{d+1}\})$	$O(1)$
3. Construct $C_{i+1} = ch(\{p_1, \ldots, p_{i+1}\})$ from $C_i = ch(\{p_1, \ldots, p_i\})$.	
1. Identify a red facet of C_i as seen from p_{i+1}.	Linear in #facets created in step i.
2. Construct three lists, respectively containing all red facets, all $(d-2)$ red faces, and all $(d-2)$ purple faces.	
3. Construct two more lists, respectively containing all remaining red faces, and all remaining purple faces.	
4. Update the incidence graph.	

CONVEX HULLS IN ARBITRARY DIMENSION

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \ldots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \ldots, p_n\})$

Algorithm	Analysis
1. Lexicographically sort the points.	$O(n \log n)$
2. Initialize $C_{d+1} = simplex\,(p_1, \ldots, p_{d+1}) = ch(\{p_1, \ldots, p_{d+1}\})$	$O(1)$
3. Construct $C_{i+1} = ch(\{p_1, \ldots, p_{i+1}\})$ from $C_i = ch(\{p_1, \ldots, p_i\})$.	
1. Identify a red facet of C_i as seen from p_{i+1}.	Linear in #facets created in step i.
2. Construct three lists, respectively containing all red facets, all $(d-2)$ red faces, and all $(d-2)$ purple faces.	Linear in #red facets of C_i and their adjacencies.
3. Construct two more lists, respectively containing all remaining red faces, and all remaining purple faces.	
4. Update the incidence graph.	

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \dots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \dots, p_n\})$

Algorithm	**Analysis**
1. Lexicographically sort the points.	$O(n \log n)$
2. Initialize $C_{d+1} = simplex\,(p_1, \dots, p_{d+1}) = ch(\{p_1, \dots, p_{d+1}\})$	$O(1)$
3. Construct $C_{i+1} = ch(\{p_1, \dots, p_{i+1}\})$ from $C_i = ch(\{p_1, \dots, p_i\})$.	
1. Identify a red facet of C_i as seen from p_{i+1}.	Linear in #facets created in step i.
2. Construct three lists, respectively containing all red facets, all $(d-2)$ red faces, and all $(d-2)$ purple faces.	Linear in #red facets of C_i and their adjacencies.
3. Construct two more lists, respectively containing all remaining red faces, and all remaining purple faces.	Linear in #red-red and red-purple incidences in C_i.
4. Update the incidence graph.	

CONVEX HULLS IN ARBITRARY DIMENSION

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \ldots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \ldots, p_n\})$

Algorithm	**Analysis**
1. Lexicographically sort the points.	$O(n \log n)$
2. Initialize $C_{d+1} = simplex\,(p_1, \ldots, p_{d+1}) = ch(\{p_1, \ldots, p_{d+1}\})$	$O(1)$
3. Construct $C_{i+1} = ch(\{p_1, \ldots, p_{i+1}\})$ from $C_i = ch(\{p_1, \ldots, p_i\})$.	
1. Identify a red facet of C_i as seen from p_{i+1}.	Linear in #facets created in step i.
2. Construct three lists, respectively containing all red facets, all $(d-2)$ red faces, and all $(d-2)$ purple faces.	Linear in #red facets of C_i and their adjacencies.
3. Construct two more lists, respectively containing all remaining red faces, and all remaining purple faces.	Linear in #red-red and red-purple incidences in C_i.
4. Update the incidence graph.	Linear in #red faces and their incidences and purple faces and their purple incidences.

CONVEX HULLS IN ARBITRARY DIMENSION

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \dots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \dots, p_n\})$

Algorithm

1. Lexicographically sort the points.
2. Initialize $C_{d+1} = simplex\,(p_1, \dots, p_{d+1}) = ch(\{p_1, \dots, p_{d+1}\})$
3. Construct $C_{i+1} = ch(\{p_1, \dots, p_{i+1}\})$ from $C_i = ch(\{p_1, \dots, p_i\})$.
    1. Identify a red facet of C_i as seen from p_{i+1}.
    2. Construct three lists, respectively containing all red facets, all $(d-2)$ red faces, and all $(d-2)$ purple faces.
    3. Construct two more lists, respectively containing all remaining red faces, and all remaining purple faces.
    4. Update the incidence graph.

Analysis

$O(n \log n)$

$O(1)$

Proportional to the number of faces and incidences created along the algorithm.

CONVEX HULLS IN ARBITRARY DIMENSION

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \dots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \dots, p_n\})$

Algorithm

1. Lexicographically sort the points.
2. Initialize $C_{d+1} = simplex\,(p_1, \dots, p_{d+1}) = ch(\{p_1, \dots, p_{d+1}\})$
3. Construct $C_{i+1} = ch(\{p_1, \dots, p_{i+1}\})$ from $C_i = ch(\{p_1, \dots, p_i\})$.
    1. Identify a red facet of C_i as seen from p_{i+1}.
    2. Construct three lists, respectively containing all red facets, all $(d-2)$ red faces, and all $(d-2)$ purple faces.
    3. Construct two more lists, respectively containing all remaining red faces, and all remaining purple faces.
    4. Update the incidence graph.

Analysis

$O(n \log n)$

$O(1)$

Proportional to the number of faces and incidences created along the algorithm.

$$\sum_{i=1}^{n} O(i^{\lfloor \frac{d-1}{2} \rfloor}) = O(n^{\lfloor \frac{d+1}{2} \rfloor})$$

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \ldots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \ldots, p_n\})$

Algorithm

1. Lexicographically sort the points.
2. Initialize $C_{d+1} = simplex\,(p_1, \ldots, p_{d+1}) = ch(\{p_1, \ldots, p_{d+1}\})$
3. Construct $C_{i+1} = ch(\{p_1, \ldots, p_{i+1}\})$ from $C_i = ch(\{p_1, \ldots, p_i\})$.
    1. Identify a red facet of C_i as seen from p_{i+1}.
    2. Construct three lists, respectively containing all red facets, all $(d-2)$ red faces, and all $(d-2)$ purple faces.
    3. Construct two more lists, respectively containing all remaining red faces, and all remaining purple faces.
    4. Update the incidence graph.

Analysis

1. $O(n \log n)$
2. $O(1)$
3. Proportional to the number of faces and incidences created along the algorithm.

$$\sum_{i=1}^{n} O(i^{\lfloor \frac{d-1}{2} \rfloor}) = O(n^{\lfloor \frac{d+1}{2} \rfloor})$$

And the space used is $O(n^{\lfloor \frac{d}{2} \rfloor})$

CONVEX HULLS IN ARBITRARY DIMENSION

COMPUTING d-DIMENSIONAL CONVEX HULLS

Input: $p_1, \ldots, p_n \in E^d$
Output: Incidence graph of $ch(\{p_1, \ldots, p_n\})$

Algorithm

1. Lexicographically sort the points.
2. Initialize $C_{d+1} = simplex\,(p_1, \ldots, p_{d+1}) = ch(\{p_1, \ldots, p_{d+1}\})$
3. Construct $C_{i+1} = ch(\{p_1, \ldots, p_{i+1}\})$ from $C_i = ch(\{p_1, \ldots, p_i\})$.
    1. Identify a red facet of C_i as seen from p_{i+1}.
    2. Construct three lists, respectively containing all red facets, all $(d-2)$ red faces, and all $(d-2)$ purple faces.
    3. Construct two more lists, respectively containing all remaining red faces, and all remaining purple faces.
    4. Update the incidence graph.

Analysis

$O(n \log n)$

$O(1)$

Proportional to the number of faces and incidences created along the algorithm.

$$\sum_{i=1}^{n} O(i^{\lfloor \frac{d-1}{2} \rfloor}) = O(n^{\lfloor \frac{d+1}{2} \rfloor})$$

And the space used is $O(n^{\lfloor \frac{d}{2} \rfloor})$

This is optimal when n is even.

FURTHER READING

J.-D. Boissonat. M. Yvinec, **Algorithmic Geometry**, Cambridge University Press, 1998.