HIGHER DIMENSIONS

Let $S \subset \mathbb{R}^d$. The convex hull of S is the smallest convex set containing S. If S is finite, then conv S is a polytope, i.e. a polytope is bounded. The intersection of an arbitrary number of halfspaces may yield an unbounded region. Such a region is a polyhedral space.

A hyperplane H has the form $\langle x, a \rangle = c$, or as expanded

$$a_1 x_1 + a_2 x_2 + \ldots + a_d x_d = c.$$

a face of P.

A hyperplane H supports P iff $P \cap H \neq 0$ and $P \subset H^{\leq}$ where $H^{\leq}$ represents the halfspace $\langle x, a \rangle \leq c$.

Each face of P is a polytope.

0-faces vertices
1-faces edges
⋮
(d-3)-faces peaks
(d-2)-faces ridges
(d-1)-faces facets

The f-vector of a polytope has: $f_j(P) = \#\, j\text{-faces of } P$, $-1 \leq j \leq d$

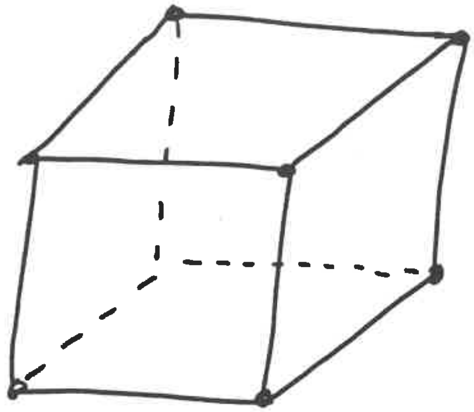
$\emptyset$ = the unique (-1)-face
P = the unique (d)-face

F(P) = set of all faces of P. Use partial order of inclusion to produce the face lattice (incidence graph) of P.

<u>OPEN QUESTION</u>: Given a lattice, is it the face lattice of some polytope P?

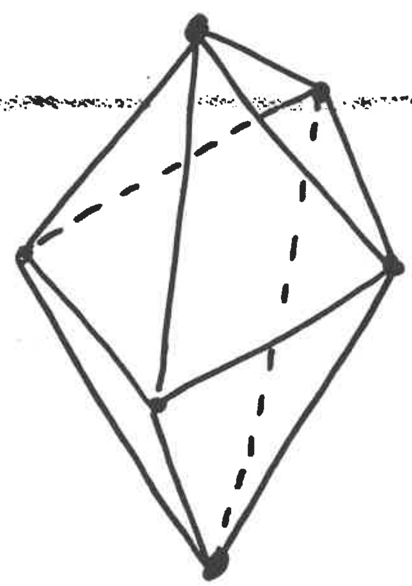
<u>COMMENT</u>: Take a face lattice and turn it upside down. The result is the face lattice of the dual polytope.

EXAMPLES



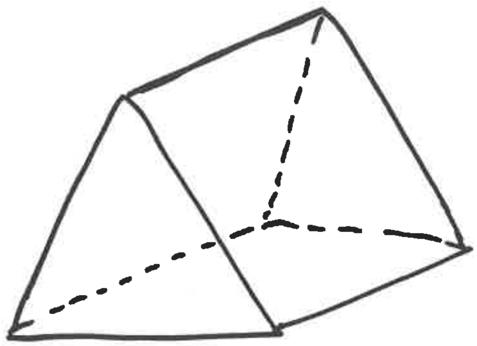
$v = 8$
$e = 12$
$f = 6$

vertex → face; edge → edge; face → vertex.

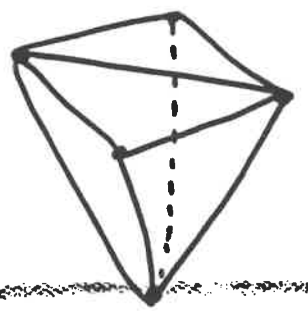


$v = 6$
$e = 12$
$f = 8$

EXAMPLES



$v = 6$
$e = 9$
$f = 5$



$v = 5$
$e = 9$
$f = 6$

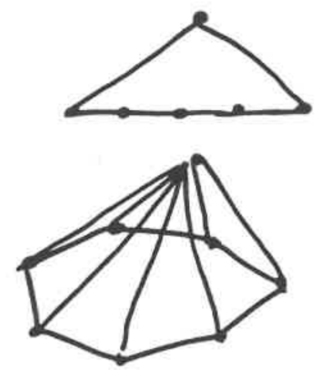
EXAMPLES OF POLYTOPES

① Simplex = convex hull of $d+1$ affinely independent points
S_d

triangle
tetrahedron
⋮

$$\Rightarrow f_j(S_d) = \binom{d+1}{j+1}$$

② Pyramid P



base → $B \subset \mathbb{R}^d$ k-polytope
apex → $x \in \mathbb{R}^d \setminus \text{aff } B$

$$P = \text{conv}(B \cup \{x\})$$

③ Cyclic Polytope

Consider the curve $\gamma(t) = (t, t^2, \ldots, t^d)$
Any d points on the curve span a hyperplane

[The intersection of a hyperplane $H: \langle a, x \rangle = c$ with $\gamma(t)$ has the form $\sum_{i=1}^{d} a_i t^i - c = 0$.
This is a d^{th} degree equation and has $\leq d$ zeros

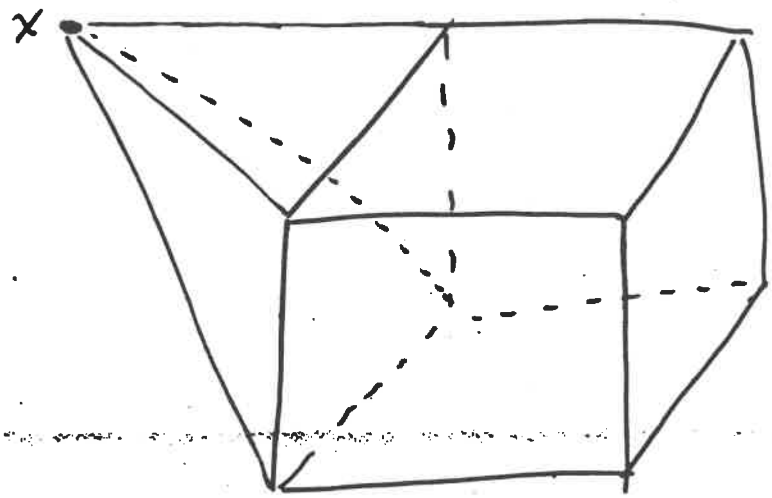
Choose n points $\gamma(t_1), \gamma(t_2) \ldots \gamma(t_n)$ to form a set S where $t_1 < t_2 < \ldots < t_n$

$C(n,d) = \text{conv } S$ = cyclic polytope (simplicial)
convex hull of n vert. in d dim.

<u>Can prove</u>: $f_{j-1}(C(n,d)) = \binom{n}{j}$ $j \leq \lfloor \frac{n}{2} \rfloor$.

$$f_j = O(n^{\lfloor \frac{d}{2} \rfloor})$$ (Upper bound)

<u>and</u> $C(n,d)$ maximizes the total number of faces.


x

Incremental Algorithm (Simplified)

Assume $S \subset \mathbb{R}^d$, $|S| = n$ and conv S simplicial.

Presort the points along one direction.

Maintain a description of the affine hull of the points encountered so far.

Let p_i represent the next point.

If $p_i \notin \text{aff}\{p_1, \dots p_{i-1}\}$, perform pyramidal updat

Else perform nonpyramidal update.

Non-pyramidal update

A facet F of P is $\begin{cases} \text{white iff hyperplane } H \supset F \text{ separates } p_i \text{ and conv}\{p_1, \dots \\ \text{black otherwise} \end{cases}$

A face G of P is $\begin{cases} \text{white iff contained only in white fac} \\ \text{black iff contained only in black fac} \\ \text{gray otherwise} \end{cases}$

F black $\Rightarrow$ F is a face of conv$\{p_1, \dots p_i\}$

F gray $\Rightarrow$ F is a face of conv$\{p_1, \dots p_i\}$

$F' = \text{conv}(F \cup \{p_i\})$ is a face also!

FACT "White facets form a connected set in the graph where facet = node, ridge = edge

FACT "One of the facets that contains p_{i-1} must be white."

Algorithm

$O(i^{\lfloor\frac{d-1}{2}\rfloor})$	1) Identify one white facet by considering the facets containing p_{i-1}.
$O(D_i)$	2) Determine remaining white facets using DFS (each node (facet) has degree d).
$O(D_i+I_i)$	3) Determine all white and gray faces
$O(D_i)$	4) Delete all white faces
$O(I_i)$	5) Make new faces from gray ones

Let i = # vertices of P
D_i = # faces of P deleted
I_i = # faces created ← each one contains the new vertex
$\Rightarrow I_i = O(i^{\lfloor\frac{(d-1)}{2}\rfloor})$

[take hyperplane and cut pt p_i off from rest of polytope $\Rightarrow$ (d-1) dim polytope.
$\Rightarrow$ use (d-1) dim version of upper bound thm

<u>Time complexity</u>: $\sum_{i<n} O(D_i+I_i) \leq \sum_{i<n} O(I_i)$
$= \sum O(i^{\lfloor\frac{d-1}{2}\rfloor}) = O(n^{\lfloor\frac{d+1}{2}\rfloor})$

Entire Incremental Algorithm:
$O(n\log n + n^{\lfloor\frac{d+1}{2}\rfloor})$

worst case optimal for even d.

gift-wrapping Chand-Kapur '70

Every ridge is contained in exactly two facets
Every edge joins exactly two points

KEY FACT: Given a facet F and a ridge R, one can find the "other" facet that contains R in $O(n)$ time

Algorithm

1) Find some facet.
   Call its ridges "open ridges"
2) While there exists an open ridge R,
3) giftwrap over R to discover new facet F
4) Update set of open ridges

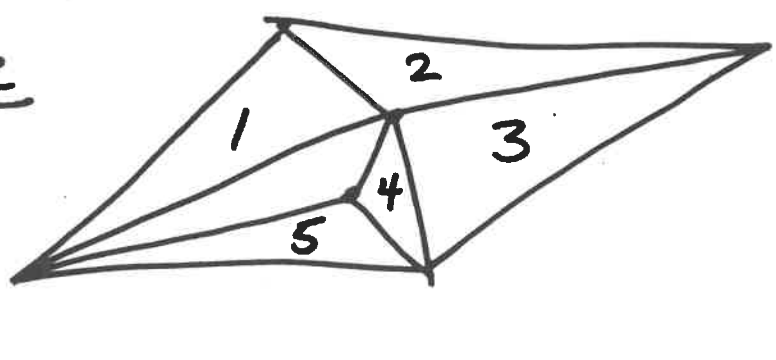
Time Complexity

$O(nF)$ where $F = f_{d-1}(\mathrm{conv}\, S)$

$O(n^{\lfloor \frac{d+2}{2} \rfloor})$ worst case.

Question: Can some revision to gift wrapping achieve better complexity?

Note



After 4, all vertices have been identified
After 5, all edges have been found.
All work thereafter is redundant.

Linear Programming

Linear programming in d variables and n constraints can be solved in $O(n)$ time when d is fixed using the prune and search technique [Megiddo and Dyer].*

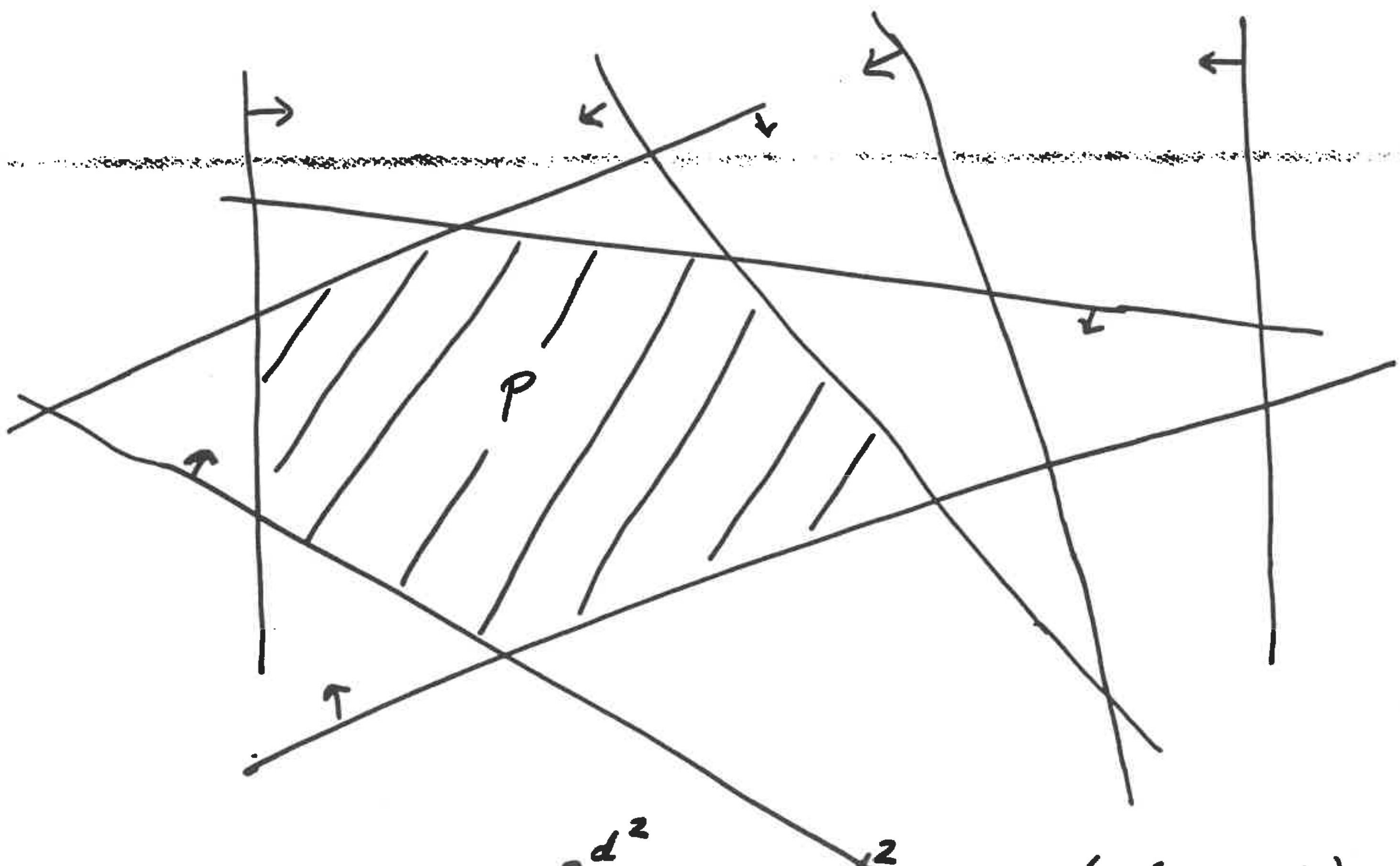
$$LP_1(n,d) \le 2^{O(2^d)} \cdot n$$

generic 2-D Linear Programing Problem

$$\begin{cases} \text{minimize} & ax+by \\ \text{subject to} & a_i x + b_i y + c_i \le 0 \quad i=1,2,\ldots N \end{cases}$$

By a change of variables, setting $Y = ax+by$ and $X = x$, we get

$$\begin{cases} \text{minimize} & Y \\ \text{subject to} & \alpha_i X + \beta_i Y + c_i \le 0 \quad i=1,2,\ldots N. \end{cases}$$



$$LP_2(n,d) \le \frac{2^{d^2}}{\prod_{k=1}^{d-2} k!}\, n \log^{d^2} n = O(\lambda(n,d))$$

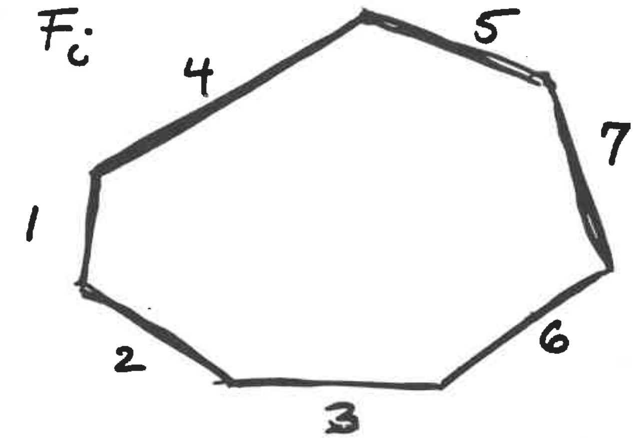
* Also can obtain a result polynomial in n and d

Shelling Algorithm: (Seidel)

Try to discover the facets in a straight-line shelling

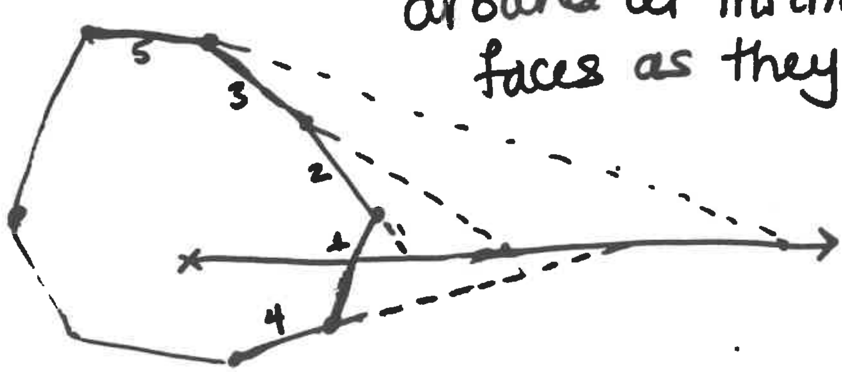
Definition: Let P = d-polytope. $F_1, \dots F_m$ is an enumeration of all facets of P. This enumeration is called a shelling iff $1 < i < m$, $F_i \cap \bigcup_{j<i} F_j$ yields the nonempty initial portion of some shelling of F_i

e.g.



Topologically, the union of the first j facets is a d-1 dimensional ball. The intersection of the j-th facet with the union of the first j-1 is a (d-2)- dimensional ball, topologically.

To obtain a straight-line shelling of P, start at a point interior to P; move in a straight line; enumerate each face as it is seen from the outside; Wrap around at infinity; enumerate remaining faces as they disappear!



[When facet i appears at p pick $q \in F_i$. shell F_i along line $\overrightarrow{qp}$].

Shelling Algorithm to Compute Convex Hull of $S \subset \mathbb{R}$

- Pick "origin" as convex combination of the points.
  Pick shelling line L through the origin: $x(t) = \frac{-1}{t} a$

Invariant:

(d-2) faces → the horizon ridges "form" a (d-1)-polytope whose facets are the ridges themselves.

(d-3) faces → the horizon peaks are each shared by exactly 2 horizon ridges

Next facet F in shelling order

Case I: F intersects the horizon in >1 ridge
⇒ F contains some horizon peak G and its two H-ridges R_1, R_2

But $\dim(R_1 \cup R_2) = \dim R_1 + \dim R_2 - \dim(\overbrace{R_1 \cap R_2}^{G})$
$= (d-2) + (d-2) - (d-3) = \underbrace{d-1}_{\text{hyperplan}}$

- Generate a hyperplane H_G for each horizon peak G.
- Keep a priority queue of the intersections of the H_G with the shelling line.

Case II F intersects the horizon in exactly 1 ridge
$\Rightarrow$ F contains some vertex that has never been seen

We need preprocessing to compute for each p,
"when is p seen for the first time, if ever?"
generic answer: Linear programming.

We want hyperplane $H \ni p \in H$ and $\forall q \in S$, $q \in H^+$
and H intersects L "first"

i.e. need to find $n = (n_1, n_2, \ldots n_d)$ such that

$$\langle p-q, n\rangle \geq 0. \quad [\text{Can insist } \langle p, n\rangle = 1.]$$

$$\langle p + \tfrac{1}{t}a, n\rangle = 0 \quad [\text{Remember } L: x(t) = -\tfrac{1}{t}a$$

$$1 = \langle p, n\rangle = -\tfrac{1}{t}\langle a, n\rangle$$

$$t = -\langle a, n\rangle$$

LP

$\min t = -\langle a, n\rangle$
$\langle p-q, n\rangle \geq 0$
$\langle p, n\rangle = 1$

If for some p, the LP is infeasible, p is interior to conv S. Otherwise, you can discover the first facet in shelling order containing p

Keep another priority queue telling you when the first facet for a new vertex should be output

TOTAL TIME COMPLEXITY: $O(n^2 + F \log n)$

If F is superlinear, this beats gift wrapping.
Worst case best in odd dimensions,
but not provably optimal.

3-D Divide and Conquer Algorithm (Preparata & Hong)

Presort the points of S with respect to x_1-coordinate.
Let $P = \{p_1, p_2, \ldots p_n\}$ represent the resulting order;
Call ConvexHull(P, n)

ConvexHull(P, n)

If $n \le 7$ then construct $\mathcal{P} = \text{conv}\, P$ by brute force.
Else

DIVIDE: set $k = \lfloor n/2 \rfloor$
define $P_1 = \{p_1, \ldots p_k\}$
define $P_2 = \{p_{k+1}, \ldots p_n\}$

RECUR: ConvexHull(P_1, k)
ConvexHull$(P_2, n-k)$

MERGE: $\mathcal{P} = \text{merge}(\mathcal{P}_1, \mathcal{P}_2)$
(i.e. wrap $\mathcal{P}_1$ and $\mathcal{P}_2$ into one polytope $\mathcal{P}$.

NOTE: $\exists\, H_0 \perp x_1$-axis such that H_0 separates $\mathcal{P}_1, \mathcal{P}_2$

H_0 intersects $\mathcal{P} = \text{conv}(\mathcal{P}_1, \mathcal{P}_2)$ in a 2-D convex polygon.

Each facet and edge of $\mathcal{P}$ which is not a facet or edge of $\mathcal{P}_1$ or $\mathcal{P}_2$ MUST intersect H_0.

These faces define a "sleeve."

Assume that all facets will be triangles.

Which facets of P_1 (P_2) should be removed?

1) Any facet F of P_1 for which there is a vertex q of P_2 such that q is on the "wrong" side of the plane containing F. Call such a facet "red."

2) edge e of P_1 if it is contained only in red facets unless e is a border edge. Call these edges "red" edges

3) vertex v of P_1 if it is only incident to red edges unless v is part of the border

Claim 1 The red facets and edges of P_1 form a connected component. [Consider dual graph] Proof by induction

Claim 2 One red facet of P_1 can be found in $O(|P_1|)$ time. [Take vertex v of P_1 with max x_1-coordinate. Take some vertex of P_2. Look for a facet of P_1 that contains v and for which w is beyond. Linear time].

Claim 3 If the border of the sleeve is known, all red facets of P_1 can be found in $O(|P_1|)$.
[Use claim 2 to find one red facet. Perform DFS. Backtrack at border edge].

How to determine the sleeve

1. Orthogonally project P_1, P_2 onto $x_1 - x_2$ plane X.
   Produce convex polygons Q_1, Q_2 separated by $X \cap H = \ell$
2. Find bridge over ℓ.
   This bridge is the projection of a "new" edge of the sleeve. Call this sleeve edge (p_{init}, q_{init})

3. Assume $e = (p, q)$ is a non-border edge of the sleeve. Want to find the "new" sleeve facet that contains e.
4. Let $ccw(p) = \{p_0, p_1, \dots p_\lambda\}$ be vertices of P_1 adjacent to p in ccw order
   Let $cw(q) = \{q_0, q_1, \dots q_\rho\}$ be vertices of P_2 adjacent to q in cw order.
5. Let $p_t \in ccw(p)$; Let $q_t \in cw(q)$
6. Search $ccw(p)$ starting at p_t for the first $p_i \ni$ the hyperplane H_1 spanned by p, q, p_i keeps p_{i-1} and p_{i+1} on the same side
7. Search $cw(q)$ starting at q_t for the first $q_j \ni$ the hyperplane H_2 spanned by p, q, q_j keeps q_{j-1}, q_{j+1} on the same side.

(H_1 tangent to P_1 at $\overline{pp_i}$
H_2 tangent to P_2 at $\overrightarrow{qq_j}$

8. Pick the winner of H_1, H_2.

Pick H_1 iff q_j is on the same side of H_1 as p_i.
Pick H_2 iff p_i is on the same side of H_2 as q_j

Assume H_1 wins. We know

$\{p, q, p_i\}$ spans a facet of the sleeve.
$\{p, p_i\}$ is a border edge
$\{p_i, q\}$ is a "new" edge of the sleeve. (non-border)

Gift wrap over this new edge.
In $CCW(p_i)$ start search at p.
In $CCW(q)$ start search at q_j

Repeat until (p_{init}, q_{init}) is reached.

Claim: for every vertex p (q) on sleeve, $CCW(p)$ and $CW(q)$ is traversed in total at most once. [Convince yourself!]

Therefore, time to find sleeve is

$$O(\sum \deg(p) + \sum \deg(q)) = O(|P_1| + |P_2|)$$

$\Rightarrow$ merge takes linear time.

Theorem: The 3-D CH problem can be solved in $O(n \log n)$ time.