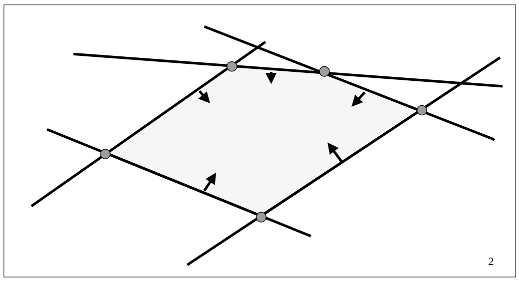
Randomized Methods in Computational Geometry

(Marco Pellegrini)

Lecture 4: Intersection of halfplanes and halfspaces

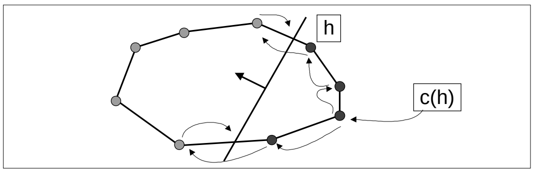
Construction of intersection of halfplanes.

- We are given a set S of halfplanes on the plane and we want to construct its intersection H(S).



Incremental algorithm.

- To build $H(S_{i+1})$ from $H(S_i)$ we add a random halfplane $h \in S \setminus S_i$. We use an auxiliary function returning a vertex of $H(S_i)$ in conflict with h, (contained in its complement).
- From there we can walk around an delete conflict vertices, generate two new vertices.



Time analysis

- We estimate the expected cost of adding h by backward analysis.
- Since the deletion of a vertex costs O(1) and a vertex is deleted only once we can charge the deletion cost to the creation.
- Let m(h) be the number of vertices incident to h. We want to bound the expectes structural change:

$$E[Str_i] = \frac{1}{i+1} \sum_{h \in S_{i+1}} m(h) = 2$$

- m(h) is always 2 in the case of polygons; we have a constant amortized cost for the i-th phase.
- We need now to maintain the conflict lists.

Maintaining conflict lists

- Suppose that in phase i we delete a vertex r with halfplane h in its conflict list. Let v_1, v_2 be the 2 new vertices created.
- Case 1: h is not in conflict with v_1 or v_2. If this happens, h will cease to be in any conflict list. This happens only once and it can be charged to the first assignment of h to a new vertex.
- Case 2. h is in conflict with say v_1. We assign h to v_1. The total cost of this case is bounded by the size of the conflict lists for v_1 and v_2.
- Question: what is the expected size of the conflict list of a vertex created in phase i?

Analysis, ctd. 2

- Intuiyion: At the beginning we have lists of size O(n), at the end of size 0, so the average list size decreases continuously in the algorithm.
- Def: $k'(S_j,e,h)$ = number of vertices in S_j, j=i+1 adjacent to e, the last inserted edge, and in conflict with h.
- Def: $k''(S_j, h)$ = number of vertices in S_j in conflict with h.
- Clearly $Cost(S_j,e,h) <= k'(S_j,e,h)$, thus also the expected cost over the choices of e, taking (in backward fashion) each s in S_j equally likely is bounded by

$$\frac{1}{j} \sum_{s \in S_j} k'(S_j, s, h) \le \frac{2}{j} k''(S_j, h)$$

Thus summing over the I's, the expected size of new conflict lists is bounded by:

$$Cost_i = \frac{2}{j} \sum_{h \in S \setminus S_j} k''(S_j, h)$$

On the other hand each element of in $S \setminus S_j$ is as likely to be the next selected halfplane $e=s_{j+1}$.

$$E[k''(S_j, s_{j+1})] = \frac{1}{n-j} \sum_{h \in S \setminus S_j} k''(S_j, h)$$

Now we express $Cost_i$ in terms of expectation on k".

$$Cost_i = \frac{2(n-j)}{j} E[k''(S_j, s_{j+1})]$$

7

Analysis Cntd. 3

- Def: DV(i) = number of vertices destroyed at phase i
  CV(i) = number of vertices created at phase i
- Note that $E[k''(S_j, s_{j+1})]$ is also the expected number of vertices destroyed in the next phase E[DV(j+1)].

$$Cost = \sum_{i=1}^{n} Cost_i = \sum_{i=1}^{n} \frac{2(n-i)}{i} E[DV(i+1)]$$

- We take the expectation out of the summation obtaining:

$$Cost = E\left[\sum_{i=1}^{n} \frac{2(n-i)}{i} DV(i+1)\right]$$

8

Analysis Cntd. 4

- Let v be any vertex created during the lifetime of the algorithm and f(v) the phase in which it is deleted.

$$\sum_{i=1}^{n} \frac{2(n-i)}{i} DV(i+1) = \sum_{v} \frac{2(n-f(v)-1)}{f(v)-1}$$

- Def: b(v) the phase in which v is created. Clearly b(v) <= f(v) - 1. Thus

$$\frac{2(n-f(v)-1)}{f(v)-1} \le \frac{2(n-b(v))}{b(v)}$$

- Thus the Total expectedcost is bounded by:

$$Cost = E\left[\sum_{v} \frac{2(n-b(v))}{b(v)}\right] = E\left[\sum_{i=1}^{n} \frac{2(n-i)}{i} CV(i)\right]$$

9

This bringing the expectation inside:

$$Cost = \sum_{i=1}^{n} \frac{2(n-i)}{i} E[CV(i)]$$

For polygons E[CV(j)]=2, so the whole summation is O(n log n)

Conclusion: we can build the intrsection of a set of halfplanes H(S) in time O(n log n).

10