Randomized Algorithm for the Detour of a Polygonal Chain

Reference: P. K. Agarwal, R. Klein, C. Knauer, S. Langerman, P. Morin, M. Sharir, and M. Soss. Computing the Detour and Spanning Ratio of Paths, Trees, and Cycles in 2D and 3D. (First two sections)

Consider a polygonal chain C

The *detour* $\delta_C(p, q)$ of C on the pair (p, q):

$$\delta_C(p, q) = \frac{|C_p^q|}{|pq|},$$

where C_p^q is the simple path from p to q in C.

The *detour* δ_C of C

$$\delta_C = \max_{p,q \in C} \delta_C(p, q).$$

For simplicity, we use $\delta(p, q)$ to represent $\delta_C(p, q)$

General Idea

- Target is to find the maximal detour pair $(p, q) \in V \times C$ instead of $C \times C$, where V is the set of polygonal vertices of C
- Orient C from p_0 to p_{n-1}.
- Develop a decision algorithm that for a given parameter $\kappa \geq 1$, determines whether for all pairs $(p, q) \in V \times C$, so that p lies before q, the inequality $\delta(p, q) \leq \kappa$ holds.
  (By reversing the orientation of C and repeating the same algorithm once more, we can also determine the case in which p lies after q.)
- Apply Chan's randomized technique to turn the decision algorithm into an optimization one.

For a point $p \in C$, we define the *weight* $w(p)$ of p

$$w(p) = \frac{|C_{p_0}^p|}{\kappa}.$$

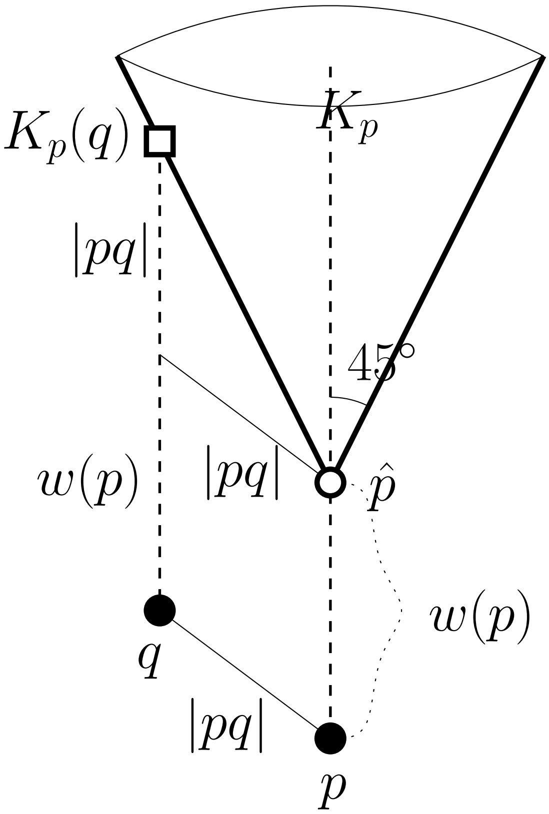
Trick:

$$\begin{aligned} &\delta(p,q) \le \kappa \\ \leftrightarrow\ &\frac{|C_p^q|}{|pq|} = \frac{|C_{p_0}^q| - |C_{p_0}^p|}{|pq|} \le \kappa \\ \leftrightarrow\ &\frac{|C_{p_0}^q|}{\kappa} \le |pq| + \frac{|C_{p_0}^p|}{\kappa} \\ \leftrightarrow\ &w(q) \le |pq| + w(p) \end{aligned}$$

Lead to a **geometric interpretation**

Geometric Interpretation:

- Let K denote the cone $z = \sqrt{x^2 + y^2}$ in $\mathbb{E}^3$.
- Map each point $p = (x_p, y_p) \in V$ to the cone $K_p = K + (x_p, y_p, w(p))$
- Also regard K_p as the graph of a bivariable function such that for any point $q \in \mathbb{E}^2$, $K_p(q) = |pq| + w(p)$. In other words, $K_p(q)$ is the distance between q and its vertical projection on K_p. Sometimes, $K_p(q)$ also means the vertical projection point from q onto K_p.
- Let $\mathfrak{K} = \{K_p \mid p \in V\}$
- Map all points $q = (x_q, y_q) \in C$ to the point $\hat{q} = (x_q, y_q, w(q))$ in $\mathbb{E}^3$.
- For any subchain π of C, we define $\hat{\pi} = \{\hat{q} \mid q \in \pi\}$



Lemm 1

For any point $q \in C$ and a vertex $p \in V$ that lies before q on C,

$$\delta(p, q) \leq \kappa$$

if and only if

$\hat{q}$ lies below the cone K_p

proof

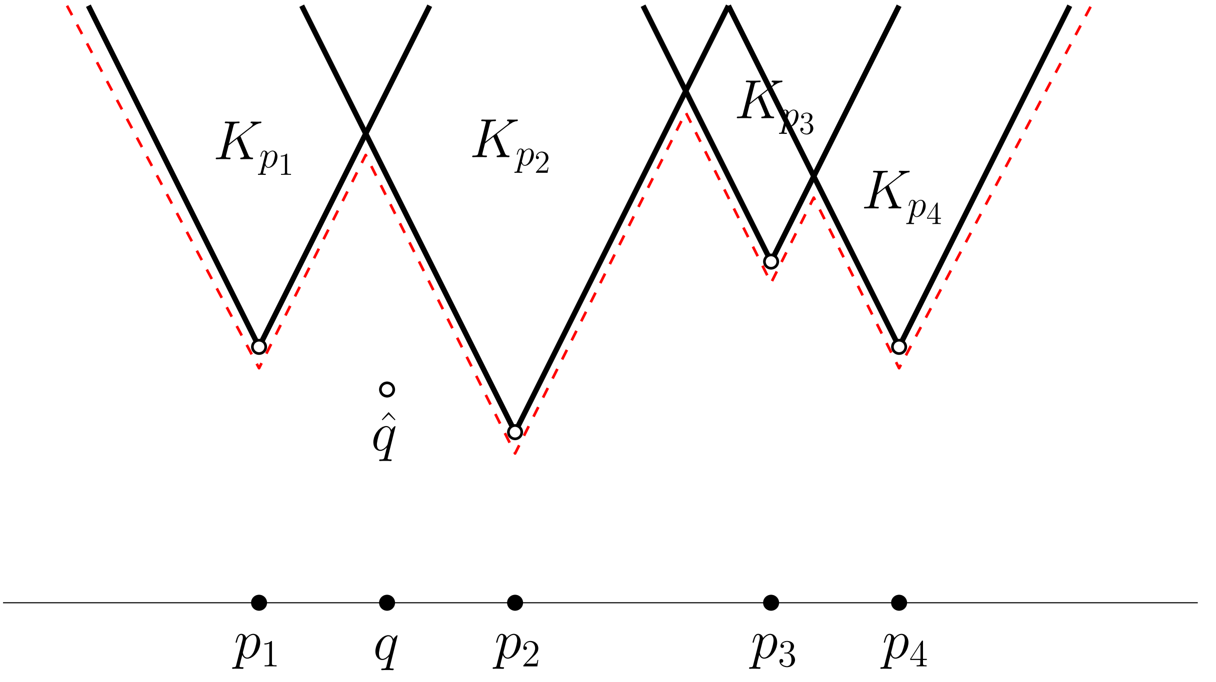
$$\delta(p, q) \leq \kappa \Leftrightarrow w(q) \leq w(p) + |pq|$$
$$\Leftrightarrow \hat{q} \text{ lies below } K_p(q)$$

Lemma 1 implies:

$\delta(\{q\}, V_q) \leq \kappa$, where V_q denotes the set of all vertices $p \in V$ that precedes q along C, if and only if $\hat{q}$ lies on or below each of the cones in $\mathcal{K}$, i.e., if and only if $\hat{q}$ lies on or below the lower envelope of $\mathcal{K}$

- Since the cones K_p are erected on the chain $\hat{C}$, the point $\hat{q}$, for any $q \in C$, always lies below all the cones erected on vertices appearing after q on P.

Cones and Lower Evenlope



Fact
$\delta_C \leq \kappa$ if and only if all points of $\hat{C}$ lie below the lower evenlope of $\mathcal{K}$.

Additively Weighted Voronoi diagram:
Given a set S of n sites with weights $w(p)$ for $\forall p \in S$, the additively weighted Voronoi diagram $\mathrm{V}_w(S)$ partitions the plane into Voronoi regions $\mathrm{VR}_w(p, S)$ such that all points in $\mathrm{VR}_w(p, S)$ share the same nearest site in S under the weighted distance.

- For a point $x \in \mathbb{R}^2$ and a site $p \in S$, the weighted distance $d_w(x, p)$ is $d(x, p) + w(p)$
- $\mathrm{V}_w(V)$ is the projection of the lower evenlope of $\mathcal{K}$ onto the xy-plane.
- $O(n \log n)$ construction time [Steven Fortune, A sweep-line algorithm for Voronoi diagrams, Algorithmica 2, pp. 153–174, 1987]

Fact
A point $q \in C$ lies in $\mathrm{VR}_w(p, V)$ if and only if $K_p(q) = \min_{p' \in V} K_{p'}(q)$

Partition C into a family E of maximal connected subchains so that each subchain lies within a single Voronoi region of $\mathrm{V}_w(V)$.

- If $\mathrm{VR}_w(p, V)$ is non-empty, p lies in $\mathrm{VR}_w(p, V)$.
- Every subchain in E either is a segment or consists of two segment incidents to $p \in V$.
- For each segment $e \in E$, if e lies in $\mathrm{VR}_w(p, V)$, it takes $O(1)$ to decide whether $\hat{e}$ lies fully below K_p
- Since $|E|$ is quadratic in the worst case, this method still takes $O(n^2)$ time

Fact (a review)
The maximum detour can be attained by a co-visiale vertex-edge cut.

- Let $\mathfrak{A}$ be the arrangement formed by C and $\mathrm{Vor}_w(V)$
- For each $p \in V$, let f_p be the face in $\mathfrak{A}$ containing p.
- For each $p \in V$, let E_p be the edges in $\mathfrak{A}$ surrounding f_p

Lemma $\hat{C}$ lies below the lower evenlope of $\mathcal{K}$ if and only if $\bigcup_{e \in E_p, p \in V} \hat{e}$ lies below the lower evenlope of $\mathcal{K}$

An $O(n \log n)$-time decision algorithm to decide whether $\delta_C \leq \kappa$

1. Compute $\mathrm{V}_w(V)$ in $O(n \log n)$ time.
2. Compute E_p for $\forall p \in V$ in $O(n \log n)$ time. (L. J. Guibas, M. Sharir, S. Sifrony. On the general motion planning problem with two degress of freedom. Discrete Computational Geometry, vol 4., pp. 491–521, 1989.)
3. For each vertex $p \in V$ and each edge $e \in E_p$, we determine whether $\hat{e}$ lies below C_p in $O(1)$ time
4. Reverse the orientation of C and repeat step 1–3 once.

In order to apply Chan's randomized technique, we need to partition a problem instance.

- Let W be a subset of V, let Q be a subchain of C, and let m be $|W| + |Q|$
- Let $\delta(W, Q)$ be $\min_{p \in W, q \in Q} \delta(p, q)$. (Remember $\delta_C = \delta(V, C)$)
- However, the maximum dilation pair (p, q) for $\delta(W, Q)$ is not necessarily a co-visible pair.
- Let $\delta^*(W, Q)$ be $\sup_{(p,q) \in W \times Q, \overline{pq} \cap Q = \emptyset} \delta(p, q)$.
- $\delta^*(p, q) \leq \delta(p, q)$ and If $\delta(W, Q) = \delta(P)$, $\delta^*(W, Q) = \delta(W, Q)$
- it takes $O(m \log m)$ time to determine if $\delta^*(W, Q) \leq t$.

Compute a pair $(\xi, \eta) \in W \times Q$ such that $\delta^*(W, Q) \leq \delta(\xi, \eta) \leq \delta(W, Q)$

- If $\delta(C) = \delta(W, Q)$, $\delta(\xi, \eta) = \delta(C)$
- If $|W|$ or $|Q|$ is a constant, we can compute $\delta(W, Q)$ in constant time and select a pair (ξ, η).
- Otherwies, we parition W into W_1 and W_2 of roughly equal size and Q into Q_1 and Q_2 of roughly equal size

$$\delta(W, Q) = \max\{\delta(W_1, Q_1), \delta(W_2, Q_1), \delta(W_1, Q_2), \delta(W_2, Q_2)\}$$

$$\delta^*(W, Q) = \max\{\delta^*(W_1, Q_1), \delta^*(W_2, Q_1), \delta^*(W_1, Q_2), \delta^*(W_2, Q_2)\}$$

Recursive Algorithm for (W, Q)

1. If $|W|$ or $|Q|$ is a constant, compute $\delta(W, Q)$ and return a pair $(\xi, \eta) \in W \times Q$ with $\delta(\xi, \eta) = \delta(W, Q)$
2. Parition W into W_1 and W_2 and Q into Q_1 and Q_2 such that $|W_1| = |W_2|$ and $|Q_1| = |Q_2|$.
3. Let (ξ, η) be $(\emptyset, \emptyset)$ and let κ be ∞
4. For $1 \leq i \leq 2$ and $1 \leq j \leq 2$
5.   If $\delta^*(W_i, Q_j) > \kappa$ (Apply the decision algorithm)
6.   let κ be $\delta(W_i, Q_j)$ (Apply this recursive algorithm)
7.   let (ξ, η) be a pair satisfying $\delta(W_i, Q_j)$
8. return (ξ, η)

Apply the recursive algorithm on (V, P) will compute $\delta(P)$ in $O(n \log n)$ expected time