Heuristic k-NNS in polygonal domain of arbitrary dimension

March 5, 2025

Input

- Dimension d
- S: set of input sites in $\mathbb{R}^d$
- O: set of disjoint polytopes in $\mathbb{R}^d$
- Query point $q \in \mathbb{R}^d$
- Approximation factor ε
- Solution size k

Output

- Approximate k-nearest neighbors of q, each paired with the distance from q.

Sketch of the algorithm

- Quadtree Construction
  - Build a quadtree with point set $S \cup V_O$, where V_O is the set of polytope vertices.
  - Add $|S|/\varepsilon$ points at random. Every such point is contained in the free space (not in any polytope). Let V_{random} be the point set.
  - For each quadtree cell with no points so far, add a single point at random (if possible) while leaving the quadtree structure as is. Let V_{extra} be the set of added points.
- Graph Construction
  - The vertex set of the graph is $S \cup V_O \cup V_{\text{random}} \cup V_{\text{extra}}$.
  - The edges of the graph is defined as follows.
    Let (c_1, c_2) be a pair of incident quadtree cells with the corresponding point sets P_1 and P_2. Here, two cells are incident if the intersection of their boundaries is nonempty.
    For every $p_1 \in P_1$ and $p_2 \in P_2$, check if the linet segment p_1p_2 intersects any polytope in O. We add an edge between p_1 and p_2 if and only if no polytope intersects p_1p_2.

- k-NN
  - Find the quadtree cell containing q with a point location query.
  - Let $P = \{p_1, ..., p_n\}$ be the input sites in the cell such that for any $p \in P$, the line segment pq does not intersect any polytope.
  - Run Dijkstra's algorithm on the graph constructed above, with priority queue initialized as $\{(\text{dist}(q, p_1), p_1), ...(\text{dist}(q, p_n), p_n)\}$. Proceed until we find k nearest input sites (as vertices) on the graph.