Counting Unit Circular Arc Intersections

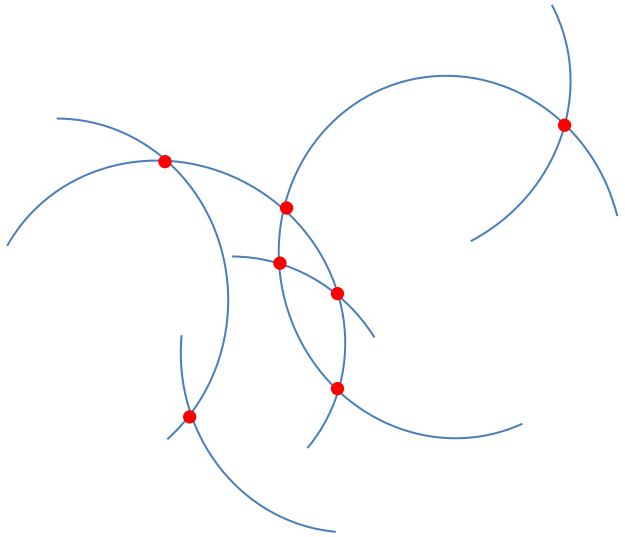
Haitao Wang

University of Utah

STACS 2026

Problem definition

- Input: a set S of n unit-circle arcs
- Output: the number of intersections among all arcs
- Previous work
  - $O(n^2)$ by brute force
  - $O(n^{4/3+\varepsilon})$, Agarwal, Pellegrini, Sharir 1993
  - $\Omega(n^{4/3})$, partition algorithm model, Erickson 1996
- Our result
  - $O(n^{4/3} \log^{16/3} n)$
  - $O(n^{1+\varepsilon} + K^{1/3} n^{2/3} (n^2/(n+K))^{\varepsilon} \log^{16/3} n)$, for K as the number of intersections
    - matches $O(n^{4/3} \log^{16/3} n)$ when $K = \Theta(n^2)$
    - $o(n^{4/3})$ for $K = O(n^{2-\varepsilon})$
    - $O(n^{1+\varepsilon})$ for $K = O(n)$
  - Main technique: divide and conquer using cuttings
    - Divide the problem into cases, subcases, subsubcases, subsubsubcases

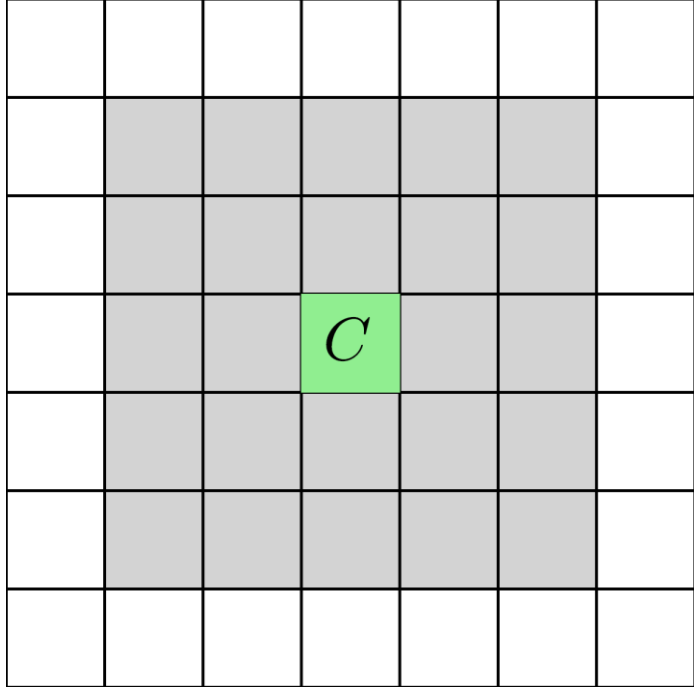


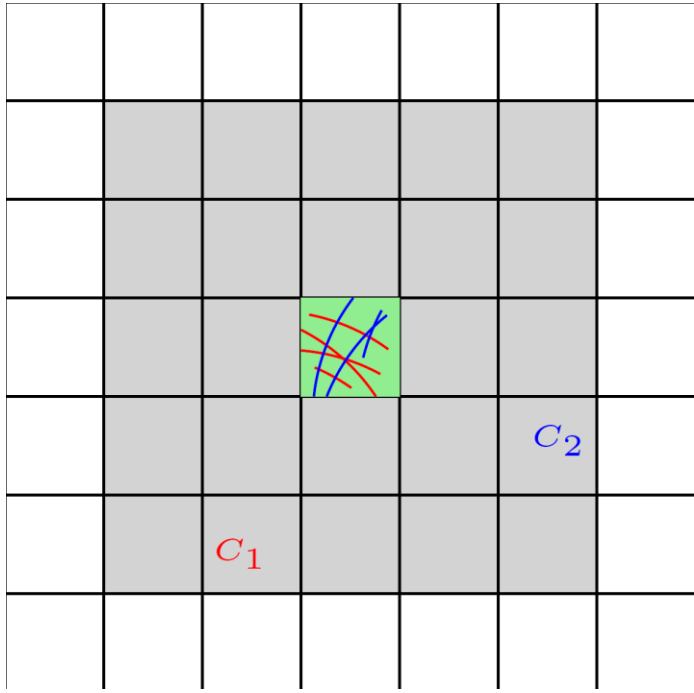
Related work

- Counting the intersections of line segments
  - Nice property: every two segments intersect at most once
  - $O(n^{1.695})$, Chazelle 1986
  - $O(n^{4/3+\varepsilon})$, Guibas, Overmars, Sharir 1989
  - $O(n^{4/3} \log^{1.78} n)$, Agarwal 1990
  - $O(n^{4/3} \log^{1/3} n)$, Chazelle 1993
  - $O(n^{4/3})$ time, Chan and Zheng 2022
    - Matching the lower bound $\Omega(n^{4/3})$, partition algorithm model, Erickson 1996
- Counting the intersections of unit circles
  - $O(n^{4/3} \log n)$, Katz and Sharir 1997
  - $O(n^{4/3} \log^{2/3} n)$, Agarwal, Aronov, Sharir, Suri 1993
  - $O(n^{4/3})$, Wang 2022
    - Matching the lower bound $\Omega(n^{4/3})$, partition algorithm model, Erickson 1996
- Counting the intersections of circular arcs of different radii
  - $O(n^{3/2+\varepsilon})$, Agarwal, Pellegrini, Sharir 1993
- Counting the intersections of algebraic arcs of constant complexity
  - $O(n^{3/2+\varepsilon})$, Chan, Cheng, Zheng 2024

Algorithm: using a grid

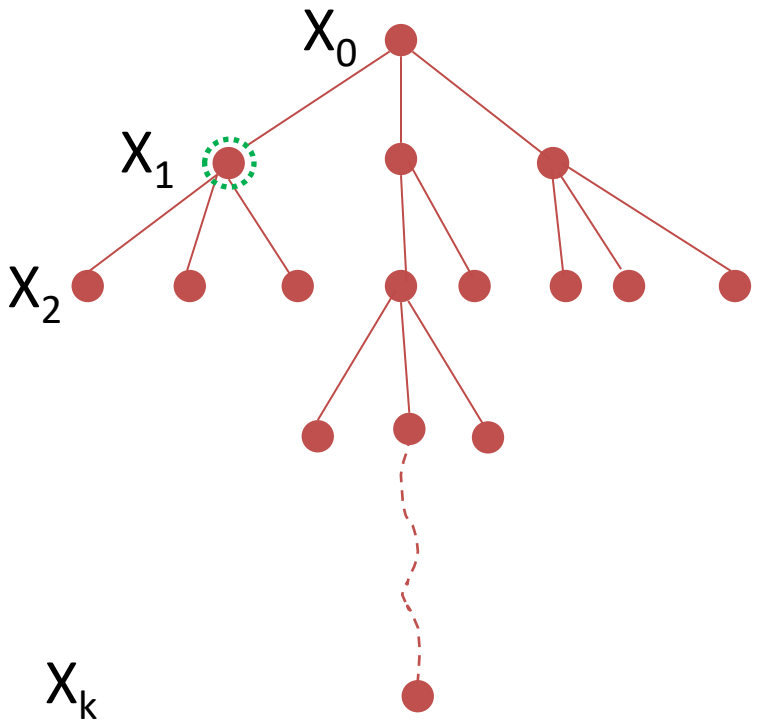
- Suppose the radius of a unit circle is 1
- Build a grid of side length ½
- For each cell C
  - if an arc intersects C, then its center must be in one of C's 25 neighboring cells
  - for every two of C's neighboring cells C_1 and C_2
    - S_1: arcs whose centers are in C_1
    - S_2: arcs whose centers are in C_2
    - key step: count the intersections of arcs of S_1 and S_2 in C
      - a bichromatic problem
      - only need to consider portions of arcs inside C
        - » a nice property: each such portion lies on a half-circle of a unit circle
  - Sum of the counts for all O(1) pairs of neighboring cells of C
- Sum of the counts for all "non-empty" cells C in the grid

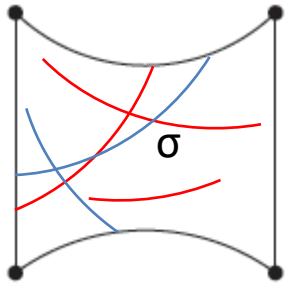




Counting the intersections of arcs of S_1 and S_2 in C
Divide & Conquer: A hierarchical cutting by Chazelle'93

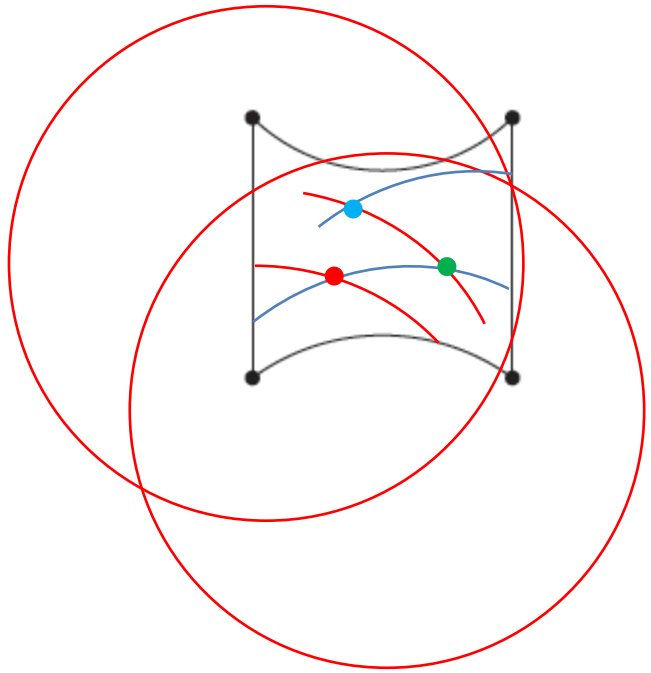
- B = S_1 (blue arcs)
- R = S_2 (red arcs)
- Compute a hierarchical (1/r)-cutting for all arcs of B and R: X_0, X_1,…,X_k, for a parameter $1 \le r \le n = |B \cup R|$
- X_0 is the entire plane
- Each cell of X_{i-1} is partitioned into O(1) child cells in X_i
  - Cells of all cuttings form a tree
  - Each cell σ is a "pseudo-trapezoid"
- X_k contains $O(r^2)$ cells such that each cell is crossed by at most n/r^2 arcs
- For each cell σ of X_k
  - key step: count the intersections between B and R inside σ
- Sum of the counts of all cells of X_k





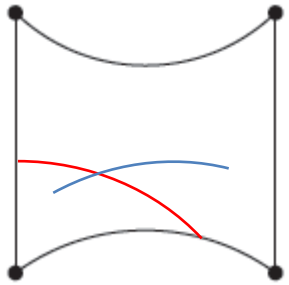
Counting the intersections of B and R in cutting cell σ

- Each unit arc can have at most two portions in σ
- Only keep the portions of each arc of BUR inside σ
  - # of arcs at most double
  - Still use B and R to represent the new portions
  - S = B U R
  - Let n = |S|
- Partition arcs of S into short and long arcs
  - Short: has at least one endpoint in σ
  - Long: both endpoints are on the boundary of σ
  - long/short red arcs vs. long/short blue arcs
  - m: # of all short arcs
- Four cases for arc intersections:
  - (1) long-red arc and long-blue arc intersections
    - Solve this case for all cutting cells in a global manner
  - (2) short-red arc and short-blue arc intersections
    - Apply the algorithm of Agarwal, Pellegrini, Sharir 1993: $O(m^{4/3+\varepsilon})$ time
  - (3) long-red arc and short-blue arc intersections
    - A new algorithm: $O(n \log^2 m + mn^{1/2} \log m)$ time
  - (4) short-red arc and long-blue arc intersections
    - Symmetric to case (3)

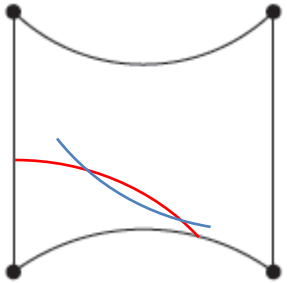


Case (3): long-red arc and short-blue arc intersections

- Two subcases:
  - (3.1) a pair of a long red arc and a short blue arc that intersect only once

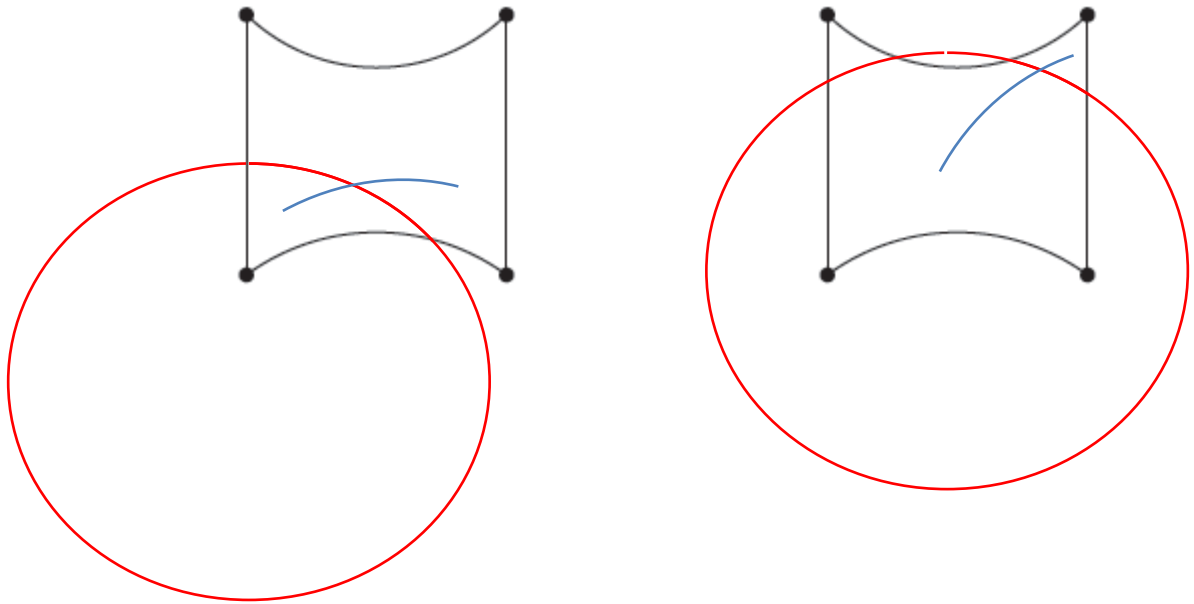


  - (3.2) a pair of a long red arc and a short blue arc that intersect twice



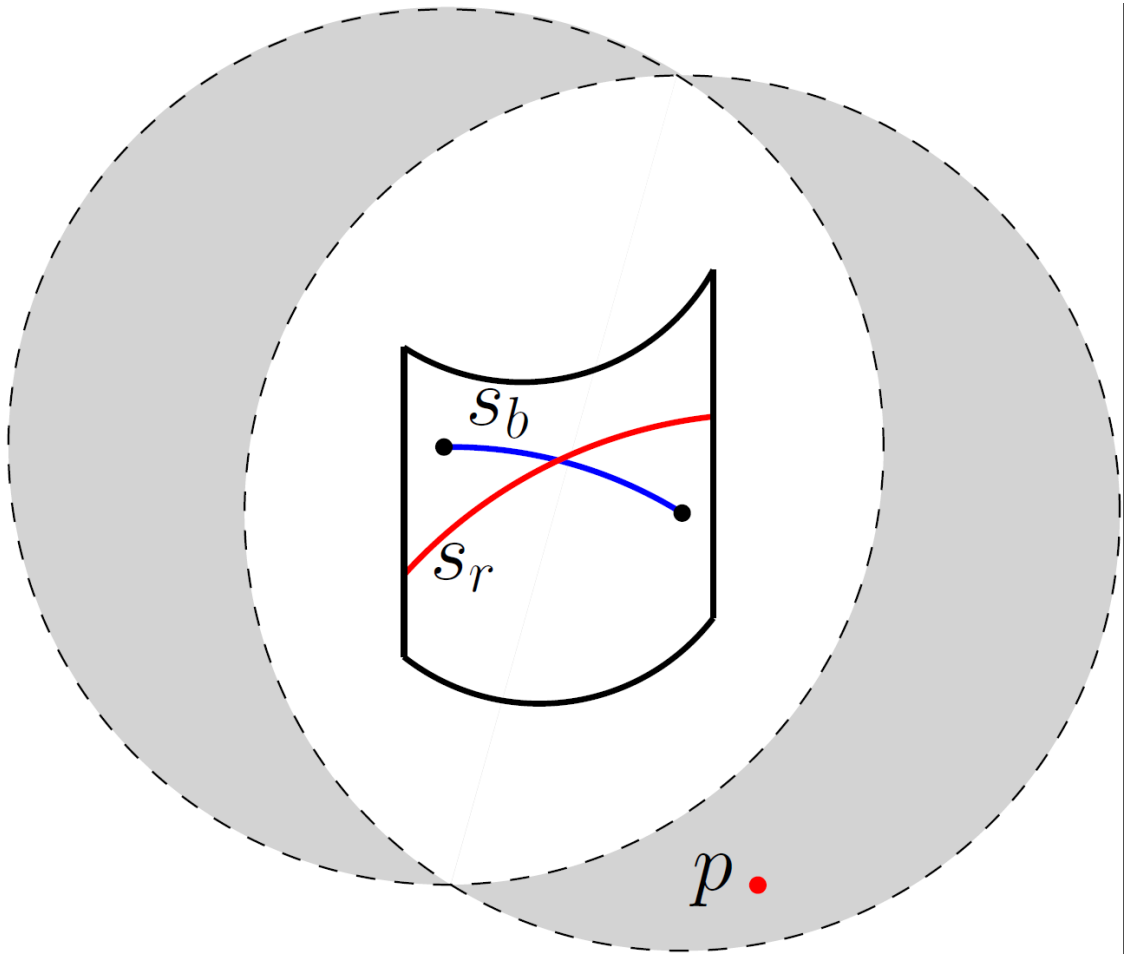
Case (3.1): a long red arc and a short blue arc that intersect only once

- For a long arc s, the circle containing s has at most two portions inside σ
  - s is a full arc if there is exactly one portion
  - s is a partial arc if there are two portions (s is one of them)
- Two subsubcases
  - (3.1.1) a long red full arc and a short blue arc that intersect only once
  - (3.1.2) a long red partial arc and a short blue arc that intersect only once



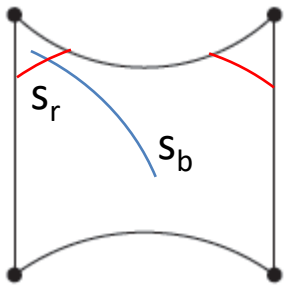
Case (3.1.1): a long red full arc and a short blue arc that intersect only once

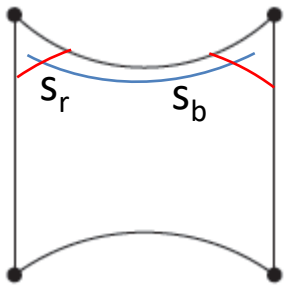
- Observation: For a short blue arc s_b and a long red full arc s_r, they intersect exactly once if and only if the center p of s_r lies in lune(s_b)
  - lune(s_b): the union of the two unit disks centered at the two endpoints of s_b minus their intersection, i.e., the symmetric difference of the two unit disks
- Algorithm: use a hierarchical cuttings on the boundary arcs of the unit disks centered at the endpoints of all short blue arcs
  - $O(m^2/\log m + n \log m)$ time

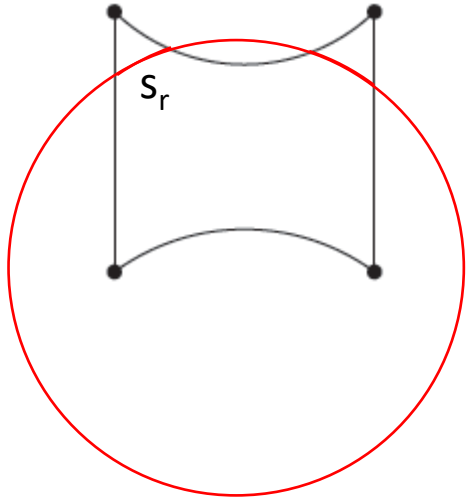


Case (3.1.2): a long red partial arc and a short blue arc that intersect only once

- For a red partial arc s_r, the circle containing it has another arc in σ and call it the coupled arc of s_r
- For a long red partial arc s_r and a short blue arc s_b that intersect only once, two subsubsubcases
  - (3.1.2.1) s_b does not intersect the coupled arc of s_r
  - (3.1.2.2) s_b intersects the coupled arc of s_r

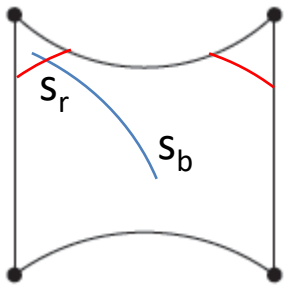


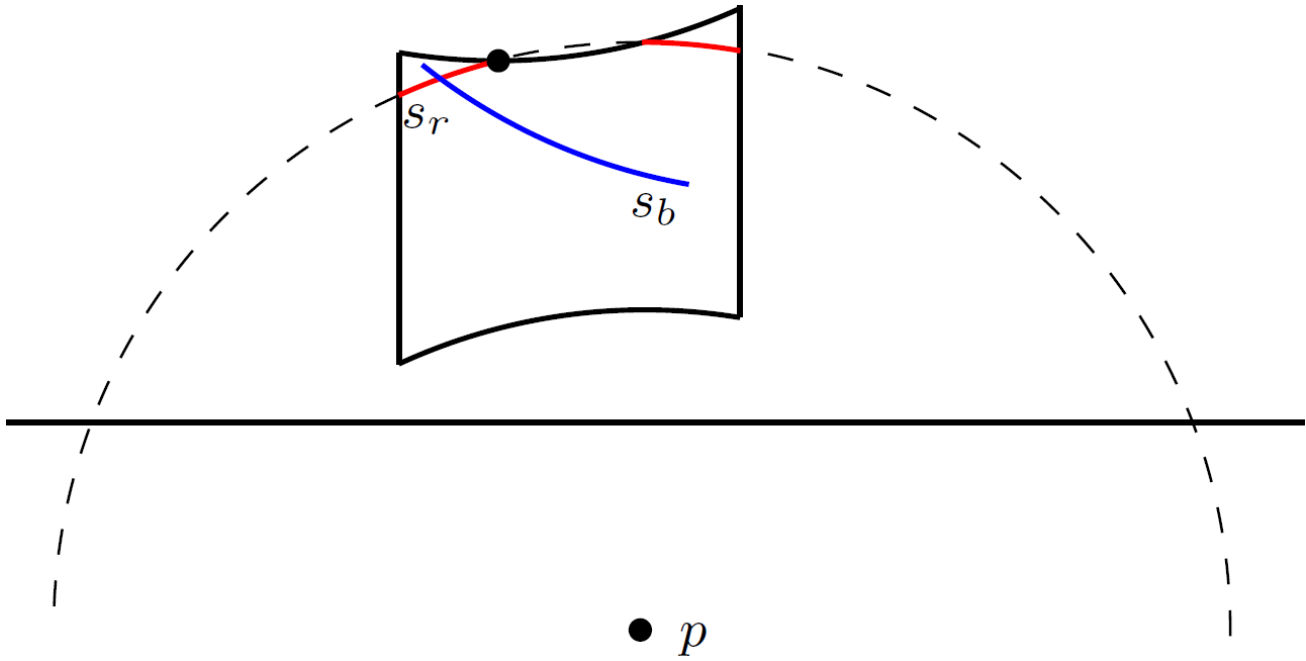




Case (3.1.2): a long red partial arc and a short blue arc that intersect only once

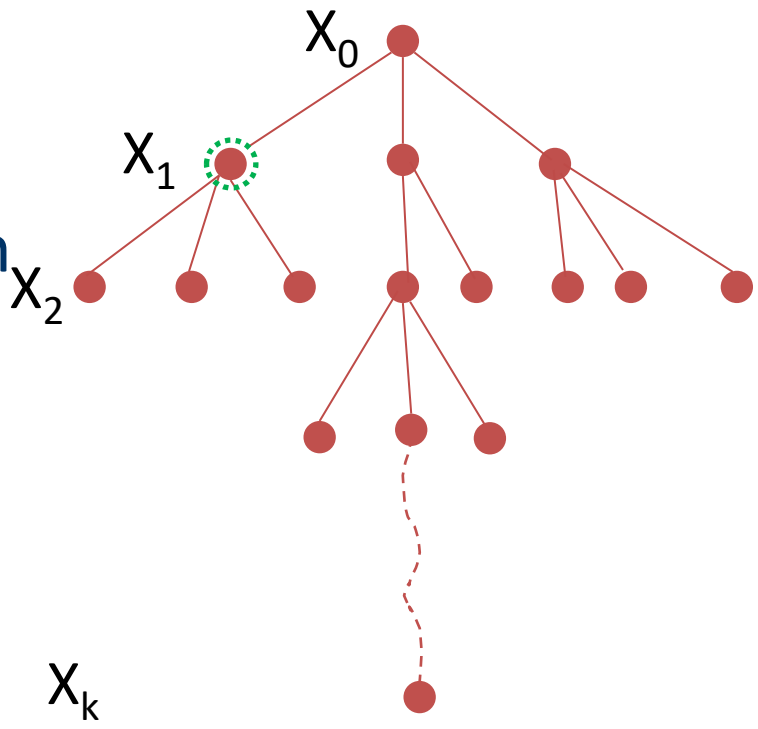
- (3.1.2.1) s_b does not intersect the coupled arc of s_r
  - Observation: This happens if and only if the following two conditions hold
    - the center p of s_r lies in lune(s_b)
    - the endpoint of s_b not in the disk of s_r lies to the left of the right endpoint of s_r
  - Algorithm:
    - Two levels: use hierarchical cuttings on the boundary arcs of the lunes as the first level and do binary search as the second level
    - $O(m^2/\log m + n\log^2 m)$ time

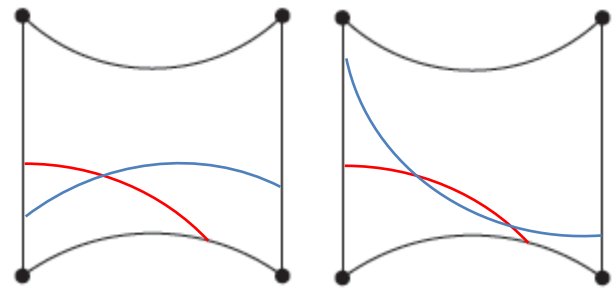




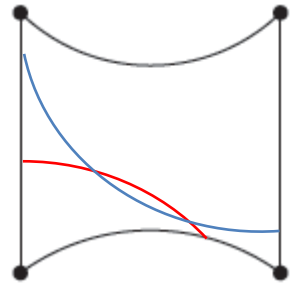
Case (1): long-red arc and long-blue arc intersections

- Cannot consider each individual cutting cell σ as before because each long arc may intersect many cells
- Our approach: counting these intersections using cells in other cuttings in the hierarchy $X_0, X_1,\ldots,X_k$,
- Key observation: a long arc s of a cell σ of X_k must be a short arc of an ancestor cell σ' of X_i for some $i < k$
  - because s must be a short arc of the single cell of X_0
  - σ': the highest ancestor of σ such that s is still a long arc
  - process the intersections of s in σ'
  - Why good?
    - Although s may be a long arc for many cells σ of X_k, s is a long arc for only one such cell σ'

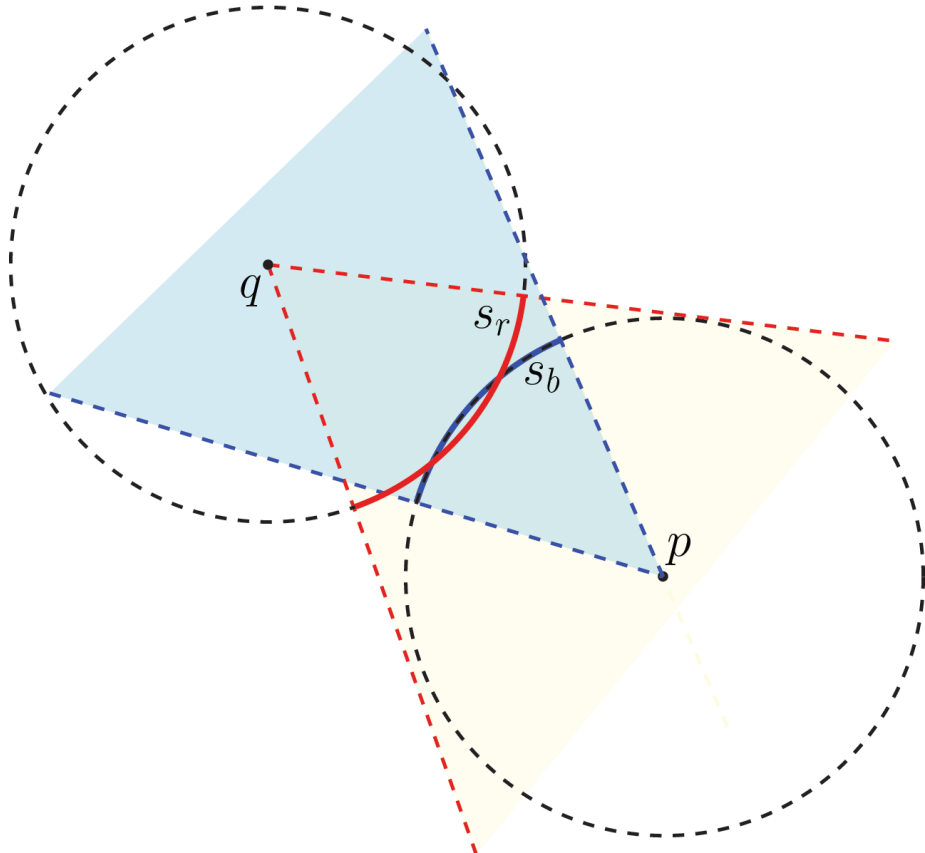




Case (1.2): A long-red arc intersects a long-blue arc twice



- Observation: this case happens if and only if the following five conditions hold:
  - the center q of s_r is in the wedge of s_b
  - the disk of s_r does not contain either endpoint of s_b
  - the center p of s_b is in the wedge of s_r
  - the disk of s_b does not contain either endpoint of s_r
  - the two disks of s_r and s_b intersect
- Algorithm: three levels
- $O(n^{4/3}\log^{16/3}n)$ time, bottleneck of the whole algorithm



Summary

- $O(n^{4/3}\log^{16/3}n)$ time
- $O(n^{1+\varepsilon} + K^{1/3}\, n^{2/3}\, (n^2/(n+K))^{\varepsilon}\, \log^{16/3}n)$, for K as the number of intersections
  - Use a hierarchical cuttings whose size is a function of K
  - Use the above algorithm as a subroutine
  - matches $O(n^{4/3}\log^{16/3}n)$ when $K = \Theta(n^2)$
  - $o(n^{4/3})$ for $K = (n^{2-\varepsilon})$
  - $O(n^{1+\varepsilon})$ for $K = O(n)$

Thank you for your attention!
Questions?