A red-blue intersection problem

Delia Garijo
University of Seville, Spain

Joint work with:
C. Cortés, M.A. Garrido, C. Grima, A. Márquez, A. Moreno, J. Valenzuela, and M.T. Villar
University of Seville, Spain

Midsummer Combinatorial Workshop XIX
The Problem
The Problem
The Problem

Report the set of segments of each colour intersected by segments of the other colour.
The Problem

Report the set of segments of each colour intersected by segments of the other colour

Geometric Intersection Problem
Geometric Intersection Problems

Segment Intersection Problem

Report the intersections of n line segments in the plane

Algorithms for reporting all k intersecting pairs

- Bentley and Ottmann (1979): $O((k+n) \log n)$ time and $O(n)$ space
- Chazelle and Edelsbrunner (1992): $O(k+n \log n)$ time and $O(k+n)$ space
- Balaban (1995): $O(k+n \log n)$ time and $O(n)$ space
Geometric Intersection Problems

Bichromatic Segment Intersection Problem

Report all intersections between \( n_r \) red segments and \( n_b \) blue segments

**Algorithms for reporting bichromatic intersections**

- **Agarwal and Sharir (1988):** \( O((n_r \sqrt{n_b} + n_b \sqrt{n_r}) \log n) \) where \( n = n_r + n_b \)
- **Agarwal (1990):** \( O(k + n^{4/3} \log^{o(1)} n) \) where \( n = n_r + n_b \)
Our Problem
Variation of the bichromatic segment intersection problem

- $R$=set of $n_r$ red points; $B$=set of $n_b$ blue points
  $n=n_r+n_b$
Our Problem
Variation of the bichromatic segment intersection problem

- R=set of $n_r$ red points; B=set of $n_b$ blue points
  $n=n_r+n_b$
- Consider all monochromatic segments
Our Problem
Variation of the bichromatic segment intersection problem

- \( R = \text{set of } n_r \text{ red points}; B = \text{set of } n_b \text{ blue points} \)
- \( n = n_r + n_b \)
- Consider all monochromatic segments
- \( S_b = \text{set of blue segments that intersect at least one red segment}; s_b = |S_b| \)
Our Problem

Variation of the bichromatic segment intersection problem

- $R =$ set of $n_r$ red points; $B =$ set of $n_b$ blue points
  $n = n_r + n_b$
- Consider all monochromatic segments
- $S_b =$ set of blue segments that intersect at least one red segment; $s_b = |S_b|$
- $S_r =$ set of red segments that intersect at least one blue segment; $s_r = |S_r|$
Our Problem
Variation of the bichromatic segment intersection problem

- $R =$ set of $n_r$ red points; $B =$ set of $n_b$ blue points
  $n = n_r + n_b$
- Consider all monochromatic segments
- $S_b =$ set of blue segments that intersect at least one red segment; $s_b = |S_b|$
- $S_r =$ set of red segments that intersect at least one blue segment; $s_r = |S_r|$

Report $S_b$ and $S_r$
The Problem: Report $S_b$ and $S_r$

- We provide an $O(n^2)$ time and space algorithm for reporting $S_b$ and $S_r$
- We prove that the problem is 3-Sum hard
The Problem: Report $S_b$ and $S_r$

- We provide an $O(n^2)$ time and space algorithm for reporting $S_b$ and $S_r$
- We prove that the problem is 3-Sum hard

What is a 3-Sum hard problem?
The Problem: Report $S_b$ and $S_r$

3-Sum Hard Problems

- Class of problems introduced by Gajentaan and Overmars (1995)
- All problems in the class are at least as hard as the base problem:
  
  Given a set $S$ of $n$ integers, are there three elements of $S$ that sum up to zero?
The Problem: Report $S_b$ and $S_r$

3-Sum Hard Problems

- Class of problems introduced by Gajentaan and Overmars (1995)

- All problems in the class are at least as hard as the base problem:
  
  Given a set $S$ of $n$ integers, are there three elements of $S$ that sum up to zero?

There is an $O(n^2)$ algorithm

Conjectured an $\Omega(n^2)$ lower bound
The $O(n^2)$ Algorithm:
Compute $S_b$
The $O(n^2)$ Algorithm:
Compute $S_b$

Be=blue exterior points to CH(R)
Bi=blue interior points to CH(R)
The $O(n^2)$ Algorithm: Compute $S_b$

$B_e$ = blue exterior points to $CH(R)$
$B_i$ = blue interior points to $CH(R)$

$G_e$ = exterior graph

Midsummer Combinatorial Workshop XIX
The $O(n^2)$ Algorithm: Compute $S_b$

- $B_e$ = blue **exterior** points to $\text{CH}(R)$
- $B_i$ = blue **interior** points to $\text{CH}(R)$

$G_e$ = exterior graph

$$S_b = \left\{ E(\overline{G_e}) \right\}$$
The $O(n^2)$ Algorithm: Compute $S_b$

Be = blue exterior points to $\text{CH}(R)$

Bi = blue interior points to $\text{CH}(R)$

$S_b = \left\{ E(\overline{G_e}) \right\}$

Ge = exterior graph
The $O(n^2)$ Algorithm: Compute $S_b$

$S_b = \begin{cases} 
E(\overline{G_e}) 
\end{cases}$

Be = blue exterior points to $\text{CH}(R)$
Bi = blue interior points to $\text{CH}(R)$

Ge = exterior graph
The $O(n^2)$ Algorithm: Compute $S_b$

$B_e$ = blue $\text{exterior}$ points to $\text{CH}(R)$

$B_i$ = blue $\text{interior}$ points to $\text{CH}(R)$

$G_e$ = exterior graph

$G_i$ = interior graph

$$S_b = \begin{cases} 
E(G_e) 
\end{cases}$$
The $O(n^2)$ Algorithm: Compute $S_b$

$B_e$=blue **exterior** points to CH(R)

$B_i$=blue **interior** points to CH(R)

$G_e$= exterior graph

$G_i$= interior graph

\[
S_b = \begin{cases} 
   E(G_e) \\
   E(G_i) 
\end{cases}
\]
The $O(n^2)$ Algorithm: Compute $S_b$

$S_b = \begin{cases} E(\overline{G_e}) \\ E(\overline{G_i}) \\ \{uv : u \in V(G_i), v \in V(G_e)\} \end{cases}$

Be = blue exterior points to CH(R)
Bi = blue interior points to CH(R)

Gi = interior graph

Ge = exterior graph
The $O(n^2)$ Algorithm: Compute $S_b$

Be=blue exterior points to CH(R)
Bi=blue interior points to CH(R)

Gi= interior graph

Ge= exterior graph

$$S_b = \begin{cases} E(\overline{G_e}) \\ E(\overline{G_i}) \\ \{uv : u \in V(G_i), v \in V(G_e)\} \end{cases}$$

$O(n \log n)$
The $O(n^2)$ Algorithm: Compute $S_b$

Computing $E(\overline{G_e})$
The $O(n^2)$ Algorithm: Compute $S_b$

Computing $E(\overline{G_e})$
The O(n²) Algorithm: Compute $S_b$

Computing $E(G_e)$

Lemma: P is an n-sided convex polygon, Q a set of n exterior points. To decide whether any of the segments with end points in Q intersects P can be done in:

1) $O(n \log n)$ time and $O(n)$ space.
2) $\Theta(n)$ time and space if we know the rotational ordering of the points of Q with respect to P.
The O(n²) Algorithm: Compute S_b

Lemma: P is an n-sided convex polygon, Q a set of n exterior points. To decide whether any of the segments with end points in Q intersects P can be done in:

1) O(n log n) time and O(n) space.
2) Θ(n) time and space if we know the rotational ordering of the points of Q with respect to P.

O(|E(\overline{G}_e)| + n \log n)
which is at most O(n²)
The $O(n^2)$ Algorithm: Compute $S_b$

Computing $E(\overline{G_i})$
The $O(n^2)$ Algorithm:
Compute $S_b$

Computing $E(\overline{G_i})$

**Key tool**: a procedure to partition CH(R) into convex regions, each one is either empty or contains only blue points whose segments are not intersected by red segments.
The O(n²) Algorithm: Compute $S_b$

Equivalence relation
$b_j \sim b_k$ if and only if the segment $b_j b_k$ crosses no red segment

Computing $E(G_i)$
The \( O(n^2) \) Algorithm: Compute \( S_b \)

Computing \( E(\overline{G}_i) \)

Equivalence relation

\( b_j \sim b_k \) if and only if the segment \( b_jb_k \) crosses no red segment
The $O(n^2)$ Algorithm: Compute $S_b$

Computing $E(G_i)$

Equivalence relation

$b_j \sim b_k$ if and only if the segment $b_jb_k$ crosses no red segment
The $O(n^2)$ Algorithm: Compute $S_b$

Computing $E(\overline{G_i})$

Equivalence relation

$b_j \sim b_k$ if and only if the segment $b jb_k$ crosses no red segment
The $O(n^2)$ Algorithm: Compute $S_b$

Computing $E(\overline{G_i})$

Equivalence relation

$b_j \sim b_k$ if and only if the segment $b_jb_k$ crosses no red segment
The $O(n^2)$ Algorithm: Compute $S_b$

Computing $E(\overline{G_i})$

Equivalence relation

$b_j \sim b_k$ if and only if the segment $b_j b_k$ crosses no red segment
The O(n²) Algorithm: Compute $S_b$

Computing $E(\overline{G_i})$

Equivalence relation

$b_j \sim b_k$ if and only if the segment $b_jb_k$ crosses no red segment
The $O(n^2)$ Algorithm: Compute $S_b$

Computing $E(\overline{G_i})$

Equivalence relation

$b_j \sim b_k$ if and only if the segment $b jb k$ crosses no red segment
Computing the planar subdivision formed by the convex regions partitioning $CH(R)$ can be done in $O(n \log n)$ time and $O(n)$ space.

**Equivalence relation**

$b_j \sim b_k$ if and only if the segment $b_jb_k$ crosses no red segment.

**Theorem:** Computing the planar subdivision formed by the convex regions partitioning $CH(R)$ can be done in $O(n \log n)$ time and $O(n)$ space.
The $O(n^2)$ Algorithm: Compute $S_b$

Lemma: $P$ is an $n$-sided convex polygon, $Q$ a set of $n$ exterior points. To decide whether any of the segments with endpoints in $Q$ intersects $P$ can be done in:

1) $O(n \log n)$ time and $O(n)$ space.

2) $\Theta(n)$ time and space if we know the rotational ordering of the points of $Q$ with respect to $P$. 

Computing $E(G_i)$
The O(n²) Algorithm: Compute Sₚ

Lemma: P is an n-sided convex polygon, Q a set of n exterior points. To decide whether any of the segments with endpoints in Q intersects P can be done in:
1) O(nlogn) time and O(n) space.
2) Θ(n) time and space if we know the rotational ordering of the points of Q with respect to P.

Go from O(n²log n) to O(n²)
- In O(n²), construct the dual arrangement of lines from the points of RUBi
- In O(n) one can read the rotational ordering of the red points with respect to a blue point in the dual.
The Problem: Report $S_b$ and $S_r$

Theorem: The sets $S_b$ and $S_r$ can be computed in $O(n^2)$ time and space.

Is the problem 3-Sum hard?
Hardness

**Theorem:** Computing $s_r$ and $s_b$ is 3-Sum hard

Number of blue segments crossed by at least one red segment
Theorem: Computing $s_r$ and $s_b$ is 3-Sum hard

Corollary: Computing $S_b$ and $S_r$ is 3-Sum hard

Number of blue segments crossed by at least one red segment
**Theorem:** Computing $s_r$ and $s_b$ is 3-Sum hard

**Corollary:** Computing $S_b$ and $S_r$ is 3-Sum hard

**Proof of the theorem**

**3-Sum Hard problem:** Given a set of $n$ points with integer coordinates on three horizontal lines $y=0$, $y=1$ and $y=2$, determine whether there exists a non-horizontal line containing three of the points.

Number of blue segments crossed by at least one red segment
Hardness

n points on each line

D: y=2
C: y=1
A: y=0
Hardness

n points on each line

D: y=2

C: y=1

A: y=0
Midsummer Combinatorial Workshop XIX

**Hardness**

- **D**: \( y = 2 \)
- **C**: \( y = 1 \)
- **A**: \( y = 0 \)

- **n points on each line**

- **Line D**: \( C_{1-\frac{1}{4}} C_1 C_{1+\frac{1}{4}} C_{2-\frac{1}{4}} C_2 C_{2+\frac{1}{4}} C_{3-\frac{1}{4}} C_3 C_{3+\frac{1}{4}} C_{4-\frac{1}{4}} C_4 C_{4+\frac{1}{4}} \)
Hardness

n points on each line

D: y=2

C: y=1

A: y=0
Hardness

n points on each line

\[ r_i = \frac{c_i + c_{i+1}}{2} \]

\( \varepsilon \) such that no blue segment with endpoints \( c - \frac{1}{4}, c + \frac{1}{4} \) is intersected by a red segment with one endpoint of the form \( (r_i, 1 + \varepsilon) \)
$r_i = \frac{c_i + c_{i+1}}{2}$

$\varepsilon$ such that no blue segment with endpoints $c - \frac{1}{4}, c + \frac{1}{4}$ is intersected by a red segment with one endpoint of the form $(r_i, 1 + \varepsilon)$
∃ line through $a \in A$, $c \in C$ and $d \in D \iff s_b > 2n(n-1)$
∃ line through $a \in A$, $c \in C$ and $d \in D \iff s_b > 2n(n−1)$
Hardness

\[ \exists \text{ line through } a \in A, \ c \in C \text{ and } d \in D \iff s_b > 2n(n-1) \]

\[ s_b \geq \frac{2n(2n-1)}{2} - n = 2n(n-1) \]

number of blue segments with endpoints \( c_j \pm \frac{1}{4} \) and \( c_k \pm \frac{1}{4} \) with \( j \neq k \)
Hardness

∃ line through \( a \in A \), \( c \in C \) and \( d \in D \) \( \iff \) \( s_b > 2n(n-1) \)

\[
s_b \geq \frac{2n(2n-1)}{2} - n = 2n(n-1)
\]

number of blue segments with endpoints \( c_j \pm \frac{1}{4} \) and \( c_k \pm \frac{1}{4} \) with \( j \neq k \)
∃ line through \( a \in A, \ c \in C \) and \( d \in D \) ⇔ \( s_b > 2n(n-1) \)

\[
s_b \geq \frac{2n(2n-1)}{2} - n = 2n(n-1)
\]

number of blue segments with endpoints \( c_j \pm \frac{1}{4} \) and \( c_k \pm \frac{1}{4} \) with \( j \neq k \)
Open Problem:
Extend the problem to 3D using monochromatic triangles
Thank you