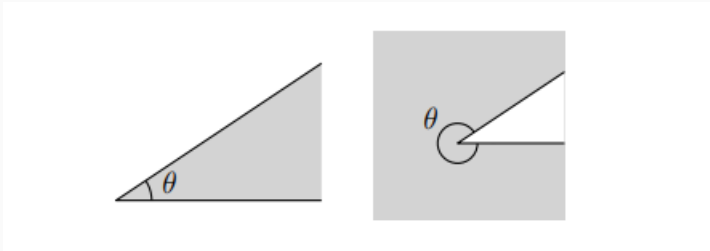


# SEARCHING FOR THE CLOSEST-PAIR IN A QUERY TRANSLATE

Gabriele Rizzitiello, Cristian Traballoni

start





## Abstract



Andremo a considerare una variante del problema Closest-pair nella sua variante range-search, ossia in un contesto di ricerca su intervalli. Sia  $\Gamma$  una forma fissa nel piano (base shape). Siamo interessati a memorizzare un insieme dato di  $N$  punti nel piano in una struttura dati tale che, per ogni traslazione specificata di  $\Gamma$ , la coppia più vicina di punti contenuta nella traslazione (quindi nella figura traslata) possa essere riportata in modo efficiente.



Presentiamo i risultati per questo problema per due casi importanti: quando  **$\Gamma$  è un poligono** e quando  **$\Gamma$  è un corpo convesso generico con bordo smooth.**

Autori: (Jie Xue, Yuan Li, Saladi Rahul Ravi Janardan)



## Abstract

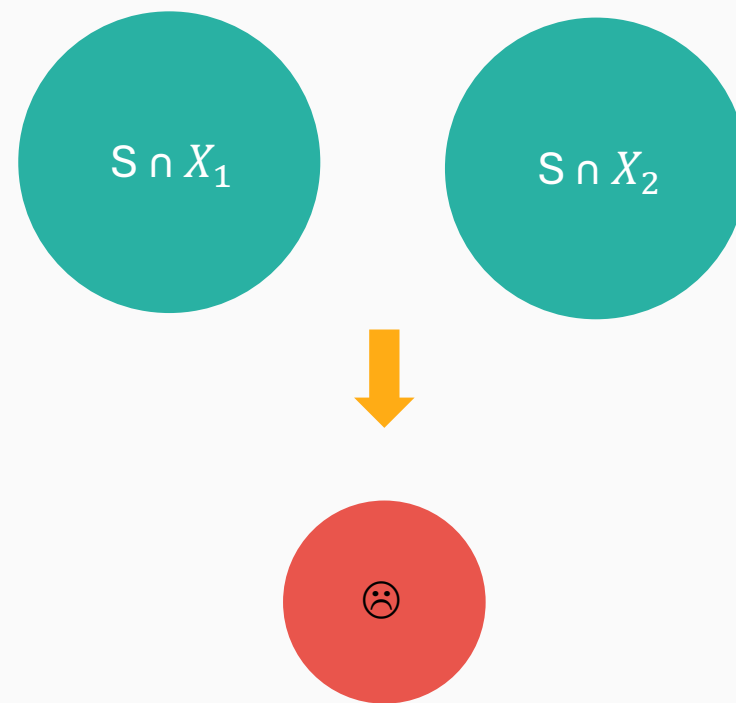


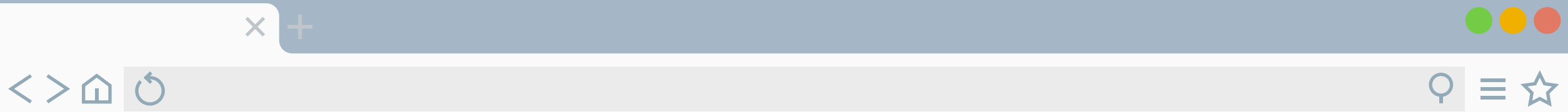
- Quando  $\Gamma$  è un **poligono**, presentiamo una struttura dati che utilizza  $O(n)$  spazio e tempo di query  $O(\log n)$ .
- Quando  $\Gamma$  è un **corpo convesso generico con un bordo smooth**, forniamo una struttura dati quasi ottimale che utilizza  $O(n \log n)$  spazio e tempo di query  $O(\log^2 n)$ .



## Closest-pair nella sua variante range-search >

- Il problema della **coppia più vicina in un intervallo** (Range Closest-Pair, **RCP**), mira a memorizzare un insieme dato  $S$  di  $n$  punti in una struttura dati tale che, per un intervallo di query specificato  $X \in \mathcal{X}$  scelto da uno spazio delle query  $\mathcal{X}$ , la coppia più vicina di punti in  $S \cap X$ , possa essere riportata in modo efficiente
- Il problema **RCP** è **non decomponibile** nel senso che la query  $X$  può essere scritta come  $X = X_1 \cup X_2$ , ma la coppia più vicina in  $S \cap X$  non può essere determinata in modo efficiente conoscendo le coppie più vicine in  $S \cap X_1$  e in  $S \cap X_2$ .





## Problema !

La proprietà di **non decomponibilità** rende il problema piuttosto complesso ma allo stesso tempo interessante, poiché molte tecniche tradizionali di ricerca su intervalli non sono applicabili.

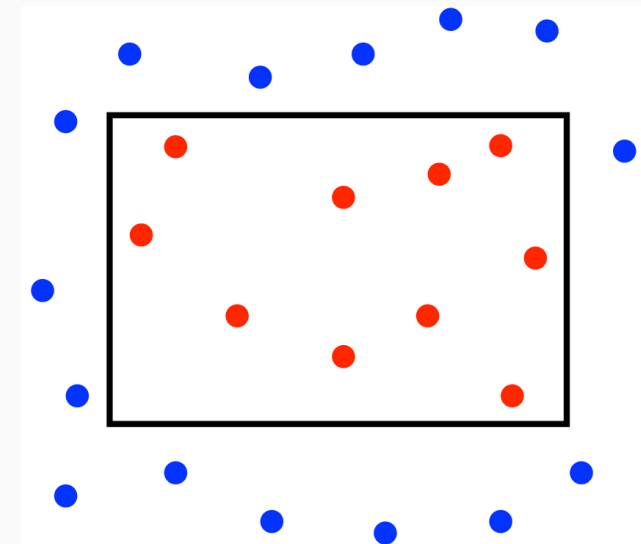
- Nonostante la trattazione preveda un approccio il più generale possibile considerando gli spazi delle query come **chiusi rispetto alla traslazione** (se una certa forma  $\Gamma$  appartiene allo spazio delle query, allora tutte le sue traslazioni appartengono anch'esse allo stesso spazio delle query, tipico dei problemi di range search), rimangono delle sfide importanti.
- ➔ • **NON è possibile utilizzare tecniche classiche di range-searching o un approccio «divide et impera», è necessario sviluppare algoritmi specifici.**

## Poligono - Considerazioni preliminari di lavoro >

Per  $a, b \in R^2$ , usiamo **dist(a, b)** per denotare la distanza euclidea tra  $a$  e  $b$ , e usiamo **[a, b]** per denotare il segmento che collega  $a$  e  $b$ . La lunghezza di una coppia  $\phi = (a, b)$  di punti, denotata da  $|\phi|$ , è la lunghezza del segmento  $[a, b]$ , quindi,  $|\phi| = \text{dist}(a, b)$ .

Per una forma  $\Gamma$  in  $R^2$  e un punto  $p \in R^2$ , denotiamo con  $\Gamma_p$  la **traslazione di  $\Gamma$  in  $p$** , cioè  $p + \Gamma$ , scriviamo dunque  $L_\Gamma = \{\Gamma_p : p \in R^2\}$ , cioè, la collezione di tutte le traslazioni di  $\Gamma$  per qualsiasi punto nello spazio  $R^2$ .

- Sia  $S$  un insieme di punti in  $R^2$  e  $\mathbf{X}$  una collezione di intervalli (o insieme di range query). Una **coppia candidata** in  $S$  rispetto a  $\mathbf{X}$  si riferisce a una coppia di punti in  $S$  che rappresenta la coppia più vicina in  $S \cap X$  per qualche  $X \in \mathbf{X}$ . Indichiamo con  $\Phi(S, \mathbf{X})$  l'insieme delle coppie candidate in  $S$  rispetto a  $\mathbf{X}$ .



## Poligono - Considerazioni preliminari di lavoro >

Un **wedge** è un intervallo in  $R^2$  definito da un angolo  $\theta \in (0, \pi)$  che è l'intersezione di due semipiani (la figura a sinistra nella Figura 1).

Un **co-wedge** è un intervallo in  $R^2$  definito da un angolo  $\theta \in (\pi, 2\pi)$ , che è l'unione di due semipiani (vedi la figura a destra nella Figura 1).

Il bordo di un wedge o co-wedge  $W$  consiste di due semirette che condividono un punto iniziale comune, chiamate i **due rami** di  $W$ . Quando appropriato, è possibile riferirsi collettivamente a wedge e co-wedge come **(co-)wedge**.

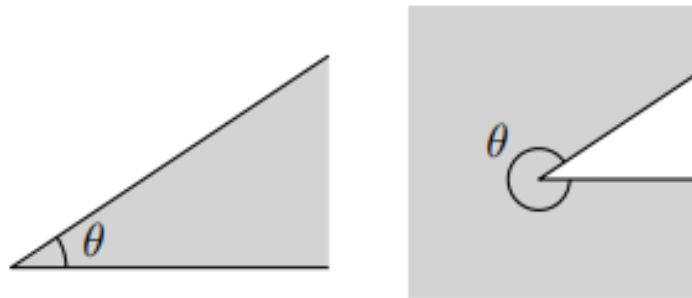
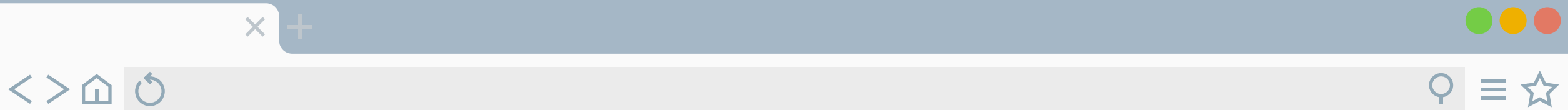


Figure 1: Examples of wedges and co-wedges.



## Lemmi utili

Lemma 1.1 e

Lemma 1.2

- I seguenti lemma sono più volte utilizzati e ritenuti particolarmente utili esplicitamente nel paper per entrambi i casi di  $\Gamma$

**Lemma 1.1:** Sia  $\Gamma$  una forma limitata fissa in  $R^2$ , e  $\mu > 0$  una costante. Inoltre, sia  $S$  un insieme di punti in  $R^2$ . Allora, per ogni punto  $p \in R^2$ , vale che: o la coppia di punti più vicina in  $S \cap \Gamma p$  ha una lunghezza minore di  $\mu$ , oppure  $|S \cap \Gamma p| = O(1)$ , cioè il numero di punti catturati è limitato ad una costante.

**Lemma 1.2:** Sia  $S$  un insieme di punti in  $R^2$ , e  $X$  una collezione di intervalli (o regioni) in  $R^2$ . Supponiamo che  $(a, b)$  e  $(a', b') \in \Phi(S, X)$  siano due coppie tali che i segmenti  $[a, b]$  e  $[a', b']$  si intersechino. Allora esiste un intervallo  $X \in X$ , tale che:

$$X \cap \{a, b, a', b'\} = \{a, b\} \text{ o } X \cap \{a, b, a', b'\} = \{a', b'\}$$



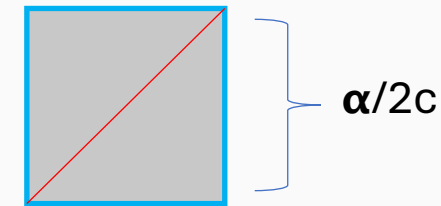
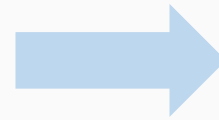
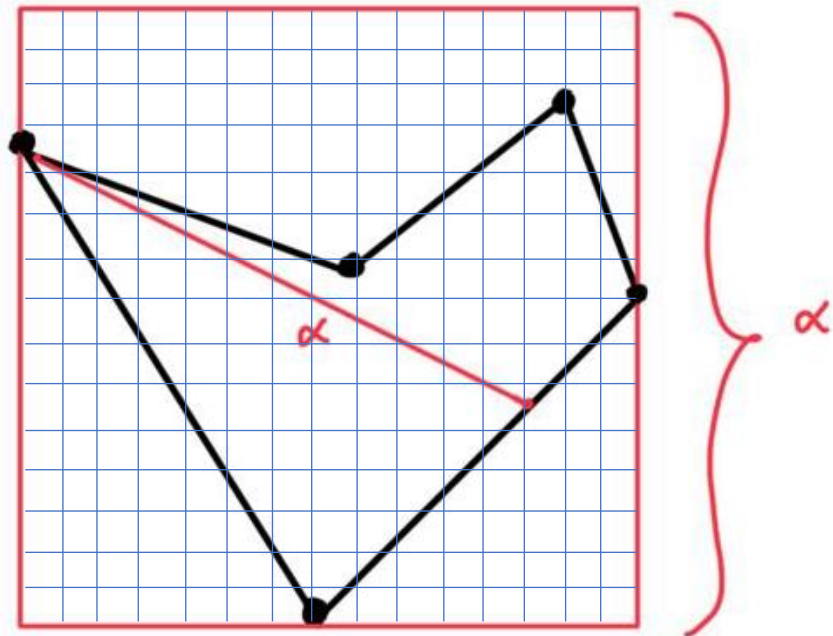
**Proof 1.1:**

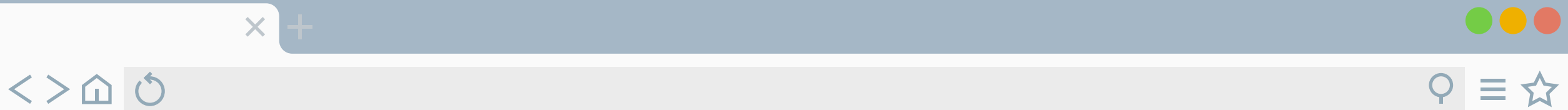
Indichiamo con  $\alpha$  il diametro di  $\Gamma$ , cioè la massima distanza tra 2 punti qualsiasi del poligono. Individuiamo il quadrato  $Z$  di lato  $\alpha$  tale che  $\Gamma \subseteq Z$ . Definiamo  $c = \alpha/\mu$  ( $c$  limitato superiormente).

Supponiamo  $p \in \mathbb{R}^2$  un punto tale che la coppia più vicina in  $S \cap \Gamma_p$  ha lunghezza maggiore o uguale di  $\mu$ .

Notiamo che  $S \cap \Gamma_p \subseteq Z_p$ .

Possiamo decomporre  $Z_p$  in piccoli quadratini di lato  $\alpha/2c \leq \mu/2$ . Siccome la coppia più vicina ha lunghezza maggiore o uguale a  $\mu$ , la distanza fra 2 punti qualsiasi deve essere almeno  $\mu$ . Quindi ogni piccolo quadratino conterrà al massimo un punto in  $S \cap \Gamma_p$ . Per il *Pigeonhole Principle*, abbiamo  $|S \cap \Gamma_p| \leq 4c^2$ .





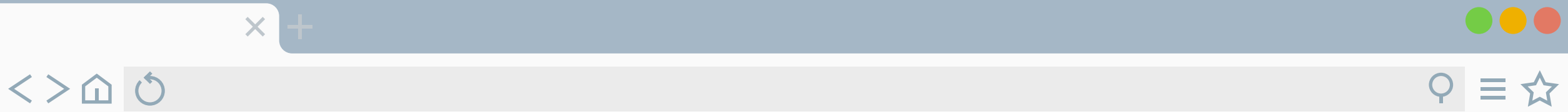
## Lemmi utili

Lemma 1.1 e  
Lemma 1.2

**Lemma 1.1:** Sia  $\Gamma$  una forma limitata fissa in  $R^2$ , e  $\mu > 0$  una costante. Inoltre, sia  $S$  un insieme di punti in  $R^2$ . Allora, per ogni punto  $p \in R^2$ , vale che: o la coppia di punti più vicina in  $S \cap \Gamma_p$  ha una lunghezza minore di  $\mu$ , oppure  $|S \cap \Gamma_p| = O(1)$ , cioè il numero di punti catturati è limitato ad una costante.

**Lemma 1.2:** Sia  $S$  un insieme di punti in  $R^2$ , e  $X$  una collezione di intervalli (o regioni) in  $R^2$ . Supponiamo che  $(a, b)$  e  $(a', b') \in \Phi(S, X)$  siano due coppie tali che i segmenti  $[a, b]$  e  $[a', b']$  si intersechino. Allora esiste un intervallo  $X \in X$ , tale che:

$$X \cap \{a, b, a', b'\} = \{a, b\} \text{ o } X \cap \{a, b, a', b'\} = \{a', b'\}$$



*Proof 1.2:*

Dimostriamo il lemma per *assurdo*:

- Supponiamo che nessun  $X \in \mathcal{X}$  soddisfi la proprietà desiderata. Siano  $X$  e  $X' \in \mathcal{X}$  i due intervalli in cui le coppie di punti più vicini sono  $(a, b)$  e  $(a', b')$ , rispettivamente.
- Allora,  $a, b \in X$  e  $a', b' \in X'$ . Per nostra ipotesi,  $X$  contiene almeno uno tra  $a'$  e  $b'$ , mentre  $X'$  contiene almeno uno tra  $a$  e  $b$ . Senza perdita di generalità, assumiamo che  $a' \in X$  e  $a \in X'$ .
- Sia  $c$  il punto di intersezione dei segmenti  $[a, b]$  e  $[a', b']$ . Poiché  $(a, b)$  è la coppia di punti più vicini in  $X$  e  $a' \in X$ ,  $\text{dist}(a, b) \leq \text{dist}(a', b)$ . Per lo stesso motivo,  $\text{dist}(a', b') \leq \text{dist}(a, b')$ .

Ne segue che  **$\text{dist}(a, b) + \text{dist}(a', b') \leq \text{dist}(a', b) + \text{dist}(a, b')$**

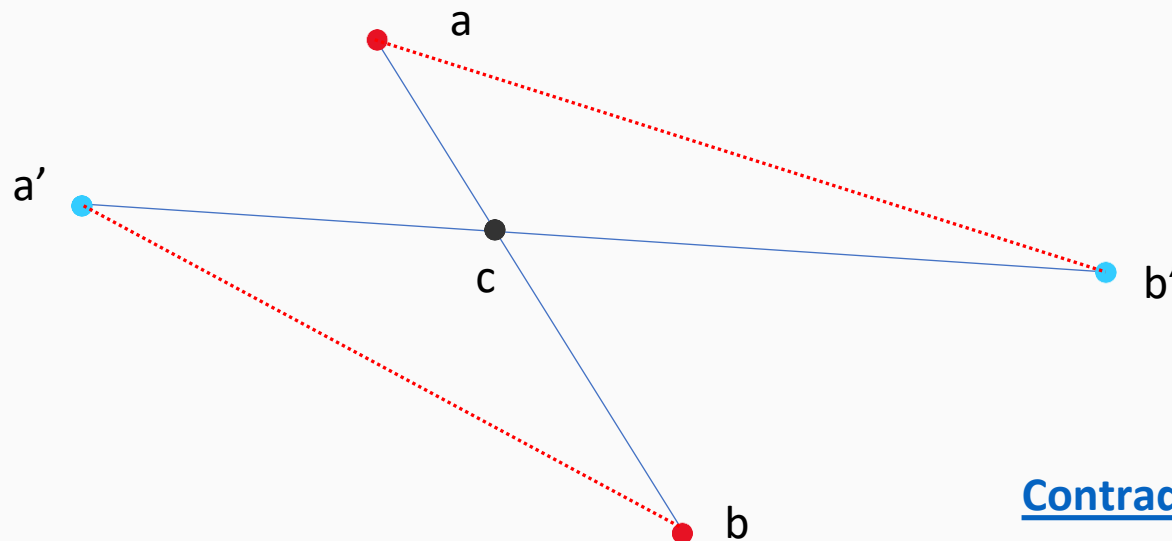


D'altra parte, per la disuguaglianza triangolare:

$$\text{dist}(a, c) + \text{dist}(b', c) > \text{dist}(a, b') \text{ e } \text{dist}(a', c) + \text{dist}(b, c) > \text{dist}(a', b)$$

Questo implica che:

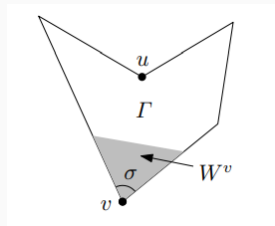
$$\text{dist}(a, b) + \text{dist}(a', b') = \text{dist}(a, c) + \text{dist}(b, c) + \text{dist}(a', c) + \text{dist}(b', c) > \text{dist}(a, b') + \text{dist}(a', b)$$



Contraddizione con la slide precedente!

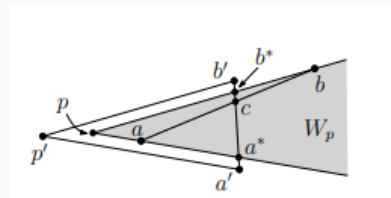


## Pipeline per query di traslazione RCP per poligoni



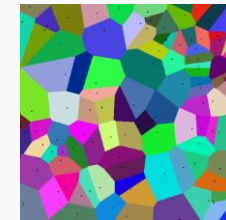
### Problema $L_{\Gamma}$ -RCP

Si utilizza un approccio basato su **griglia** per semplificare il problema e trasformarlo tramite i concetti di **(co-)wedge**



### Problema $L_W$ -RCP

Il problema originario si trasforma diventando un problema **RCP** con query di traslazione **wedge/co-wedge**

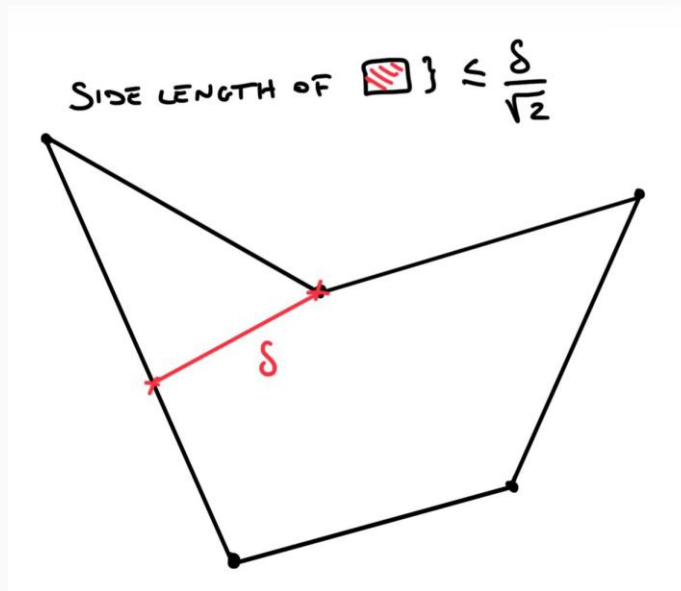


### Problema Range-Reporting

Si affianca al problema  $L_W$ -RCP anche un problema di **Range-Reporting** con query di traslazione  $\Gamma$

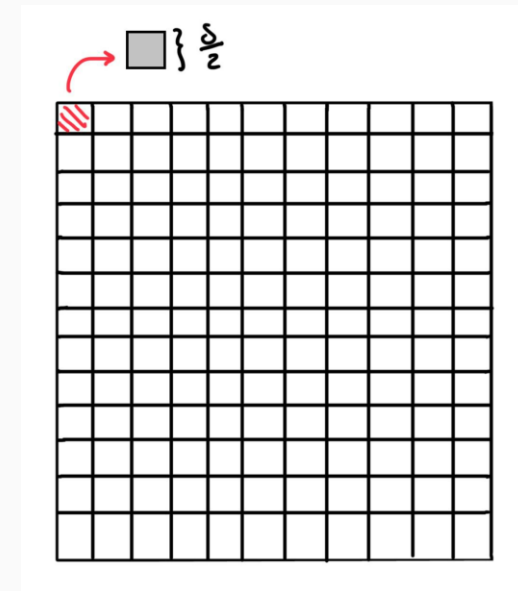
Per un problema di range-search **decomponibile** (ad es. Range Reporting) su un pointset  $S$ , la precedente semplice osservazione ci permette già di ridurre (approssimativamente) una query di traslazione- $\Gamma$  a query di traslazione (co-)wedge nel modo seguente.

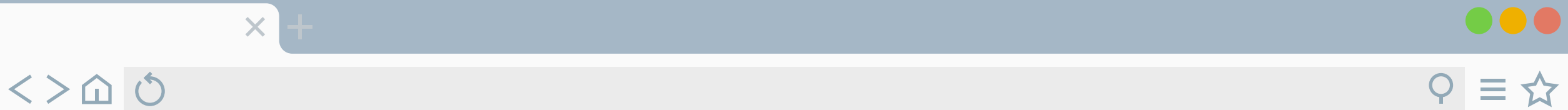
- Stabiliamo quindi l'approccio a griglia basato sul lemma 1.1 e sulla precondizione che  $\partial\Gamma$  sia di almeno 4 lati, così che nessuna terna di spigoli è a 2 a 2 adiacente:



Definiamo  $\delta = \min\{\text{dist}(e, e') : e, e' \text{ sono spigoli non adiacenti di } \Gamma, \text{ dipende solo dal poligono } \Gamma\}$

La griglia  $G$  è realizzata affinché una cella non possa mai intersecarsi con 2 spigoli che non siano adiacenti





Per una cella  $\square$  di  $G$ , definiamo  $S_{\square} = S \cap \square$ . A causa della decomponibilità del problema, **per rispondere ad una query  $\Gamma_q$  su  $S$ , è sufficiente rispondere alla  $\Gamma_q$  su  $S_{\square}$  per tutte le  $\square$  che intersecano  $\Gamma_q$ .**

- Ogni cella di  $G$  è un quadrato di lato  $\delta/2$  (che è minore di  $\frac{\delta}{\sqrt{2}}$ ), abbiamo quindi  $\square \cap \Gamma_q = \square \cap W_q$  per qualche  $W \in W_{\Gamma}$ , e quindi  $S_{\square} \cap \Gamma_q = S_{\square} \cap W_q$ . Applicando questa idea al range-reporting, concludiamo quanto segue.

Lemma 2.2: Esiste una struttura dati per il **Range-Reporting** con traslazioni di  $\Gamma$  che utilizza  **$O(n)$**  spazio e ha un tempo di query pari a  **$O(\log n + k)$** , dove  **$k$**  è il numero di punti restituiti dalla query.



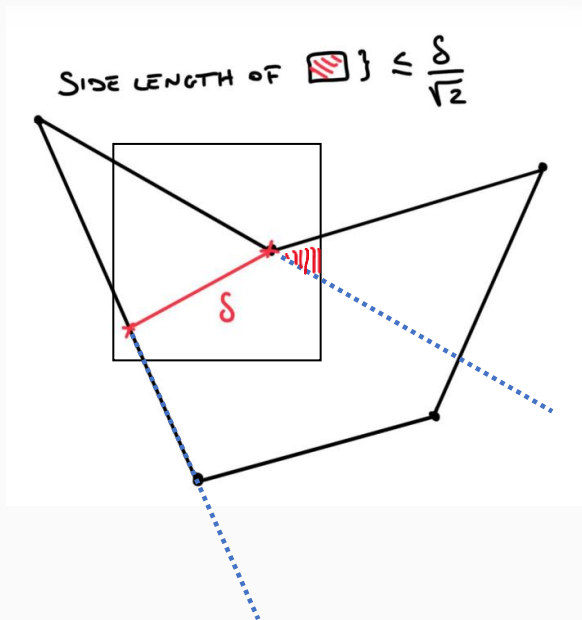
**Sfortunatamente, il nostro problema RCP appartiene alla categoria dei problemi non decomponibili, abbiamo bisogno di nuove idee.**



Perché il lato di  $\square$  deve essere  $\leq \frac{\delta}{\sqrt{2}}$  ?

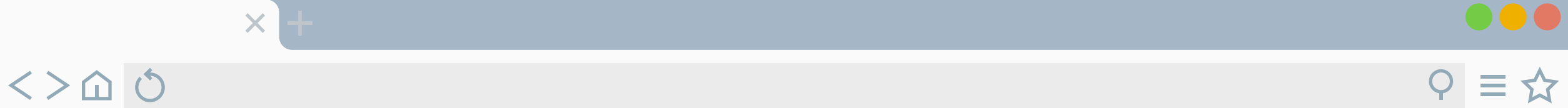


E' possibile notare che se il lato fosse maggiore di  $\frac{\delta}{\sqrt{2}}$  si potrebbe ricadere nel caso di intersezione con 3 spigoli, il che comporta la non validità della condizione:  $\square \cap \Gamma_q = \square \cap W_q$



In figura è mostrato un esempio: la zona in rosso rappresenta la differenza tra l'intersezione del poligono e la cella e l'intersezione del wedge e la stessa cella





Sia  $\mathbf{S}$  il dataset dato in  $\mathbf{R}^2$  di dimensione  $n$ . Supponiamo, per comodità, che le distanze a coppie dei punti in  $S$  siano distinte (così la coppia più vicina in ogni sottoinsieme di  $S$  è unica). Il principale risultato esposto in questa sezione del paper dedicato alla query di traslazione RCP per poligoni è il seguente teorema:

### Teorema 2.1

Sia  $\Gamma$  un poligono fisso in  $\mathbf{R}^2$ . Esiste una struttura dati  $L_\Gamma$ -RCP con spazio  $O(n)$  e tempo di query  $O(\log n)$ .

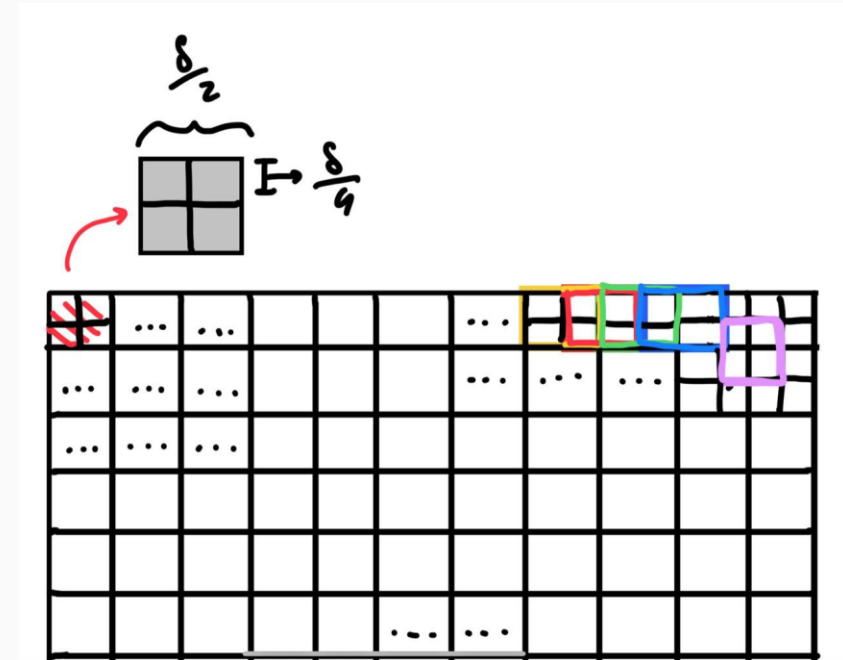
Ma a causa della non decomponibilità del problema RCP al fine di dimostrare il precedente teorema sarà necessario dimostrare un ulteriore teorema basato su range query non correlate direttamente al poligono traslato ma alle regioni (co-)wedge traslate, individuate per un poligono generico. Il nuovo teorema è il seguente:

### Teorema 2.4

Siano  $f, g: \mathbf{N} \rightarrow \mathbf{N}$  funzioni crescenti tali che  $f(a + b) \geq f(a) + f(b)$ . Se per ogni  $W \in \mathcal{W}_\Gamma$  esiste una struttura dati  $L_W$ -RCP con spazio  $O(f(n))$  e tempo di query  $O(g(n))$ , allora esiste una struttura dati  $L_\Gamma$ -RCP con spazio  $O(f(n) + n)$  e tempo di query  $O(g(n) + \log n)$ .

Stabiliamo ancora una **griglia planare G**. Ma questa volta, impostiamo la larghezza di una cella di G a  $\frac{\delta}{4}$ :

- Definiamo ora il concetto di **quad-cella**: una quad-cella  $\oplus$  di G è un quadrato costituito da  $2 \times 2$  celle adiacenti di G. Per una quad-cella  $\oplus$  di G, sia  $S_{\oplus} = S \cap \oplus$ . Notiamo che il lato di una quad-cella di G è  $\delta/2$ , e ogni cella di G è contenuta esattamente in quattro quad-celle di G, così come ogni punto in S. Consideriamo un range di query  $\Gamma_q \in L_{\Gamma}$ . Questa osservazione deriva dal lemma 1.1.



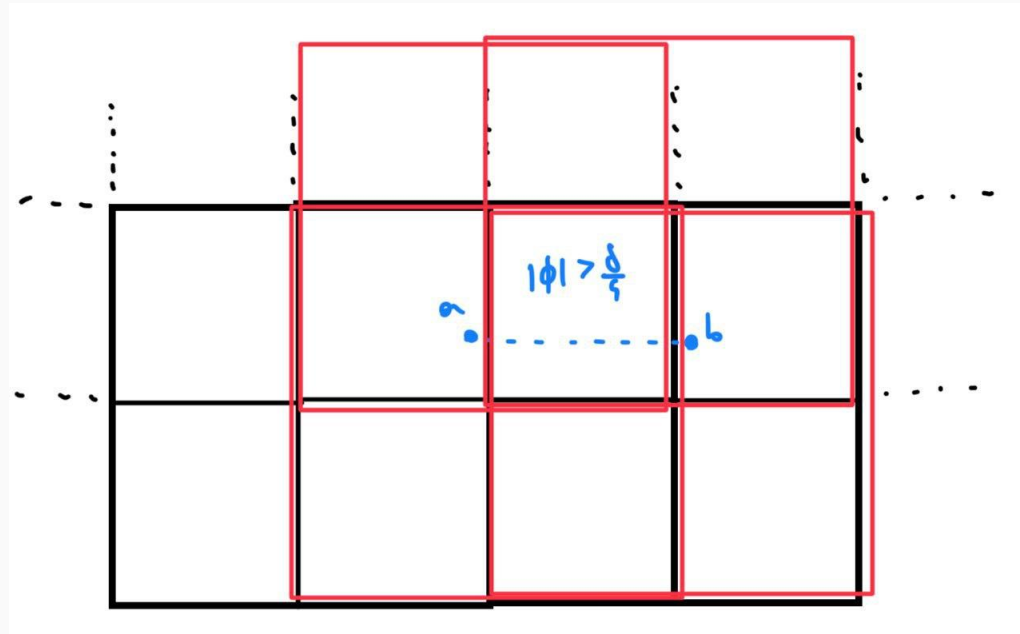
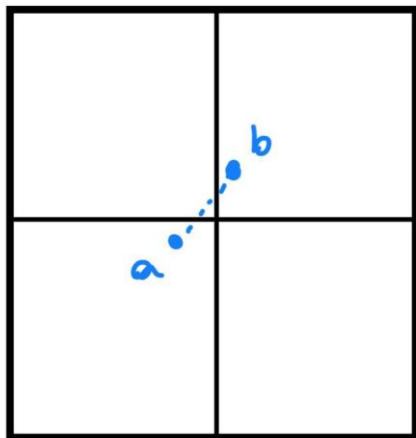
**Lemma 2.3:** Per una quad-cella  $\oplus$  di G tale che  $|S_{\oplus} \cap \Gamma_q| \geq 2$ , sia  $\Phi_{\oplus}$  la coppia più vicina in  $S_{\oplus} \cap \Gamma_q$ . Definiamo  $\phi^*$  come l'elemento coppia più corto tra tutte le coppie più vicine  $\Phi_{\oplus}$ . Se la lunghezza di  $\phi^*$  è al massimo  $\frac{\delta}{4}$ , allora  $\phi^*$  è la coppia più vicina in  $S \cap \Gamma_q$ ; altrimenti  $|S \cap \Gamma_q| = O(1)$ .

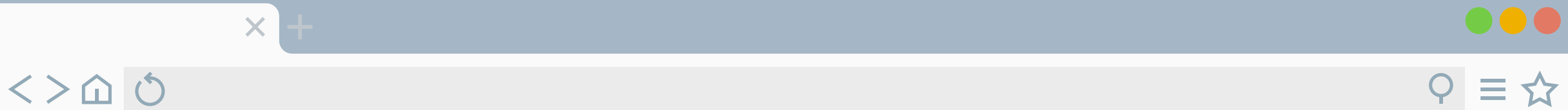
*Proof. 2.3:*

Per prima cosa notiamo che per 2 punti  $a, b \in \mathbb{R}^2$  che hanno  $\text{dist}(a, b) \leq \delta/4$ , il segmento  $[a, b]$  incrocerà al massimo una linea orizzontale e una linea verticale di una  $\oplus$ , in questo modo esisterà sempre una quad-cella di  $G$  tale che  $a, b \in \oplus$ .

Se la lunghezza di  $\phi^*$  è **al massimo  $\delta/4$** , la lunghezza della coppia più vicina in  $\text{Sn}\Gamma_Q$  è anch'essa al massimo  $\delta/4$  **il che implica che la coppia più vicina è contenuta in qualche quad-cell di  $G$** , e quindi deve essere  $\phi^*$ . D'altra parte, se  $\phi^*$  **ha una lunghezza maggiore di  $\delta/4$** , allora o la coppia più vicina in  $\text{Sn}\Gamma_Q$  è  $\phi^*$  oppure non è contenuta in alcuna quad-cell di  $G$ . In entrambi i casi, la lunghezza della coppia più vicina in  $\text{Sn}\Gamma_Q$  è maggiore di  $\delta/4$ , e abbiamo  $|\text{Sn}\Gamma_Q| = O(1)$  per il Lemma 1.1.

$$|\phi| \leq \frac{\delta}{4}$$





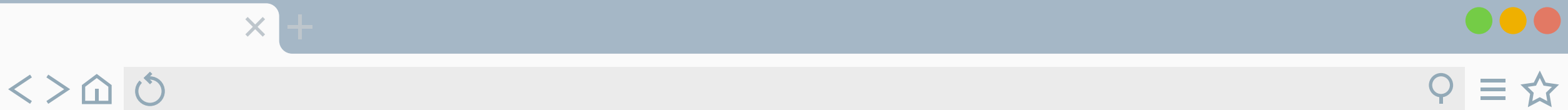
Il teorema 2.4 definisce in particolare che  **$O(f(n))$**  rappresenta il limite superiore per lo spazio di memoria richiesto dalla struttura dati.

**Dimostrazione:** per una quad-cella  $\oplus$  di  $G$ , sia  $m_{\oplus}$  il numero di punti in  $S_{\oplus}$ . Notiamo che ci sono  $O(n)$  quad-celle di  $G$  tali che  $m_{\oplus} > 0$ , poiché ogni punto in  $S$  è contenuto al massimo in quattro quad-celle, queste le chiameremo *quad-celle non vuote*.

- **Punti contenuti in quad-celle:** Ogni punto di  $S$  si trova all'interno di una cella di  $G$ , e ogni cella si trova in 4 quad-celle. Quindi, ogni punto di  $S$  si trova all'interno di al massimo quattro quad-celle.
- **Implicazione:** Di conseguenza, il numero di quad-celle non vuote è al massimo 4 volte il numero di punti, ovvero lineare rispetto ad  $n$ .

Per ogni quad-cella non vuota  $\oplus$  e per ogni  $W \in W_{\Gamma}$ , costruiamo una struttura dati  **$L_W$ -RCP** basata sui punti  $S_{\oplus}$ , per ipotesi, questa struttura dati utilizza uno spazio  $O(f(m_{\oplus}))$ . Osserviamo che  $m_{\oplus} \leq n$  per tutte le  $\oplus$  e che  $\sum m_{\oplus} \leq 4n$ . Dalla condizione  $f(a + b) \geq f(a) + f(b)$ , segue che  $\sum f(m_{\oplus}) = O(f(n))$ , si sta sfruttando la *subadditività* della funzione. Poiché  $|W_{\Gamma}| = O(1)$ , il costo totale dello spazio di queste strutture dati è  $O(f(n))$ .

Oltre a queste strutture dati costruiamo una struttura dati di Range-Reporting su tutto  $S$  che, come stabilito dal Lemma 2.2, usa spazio  $O(n)$ .

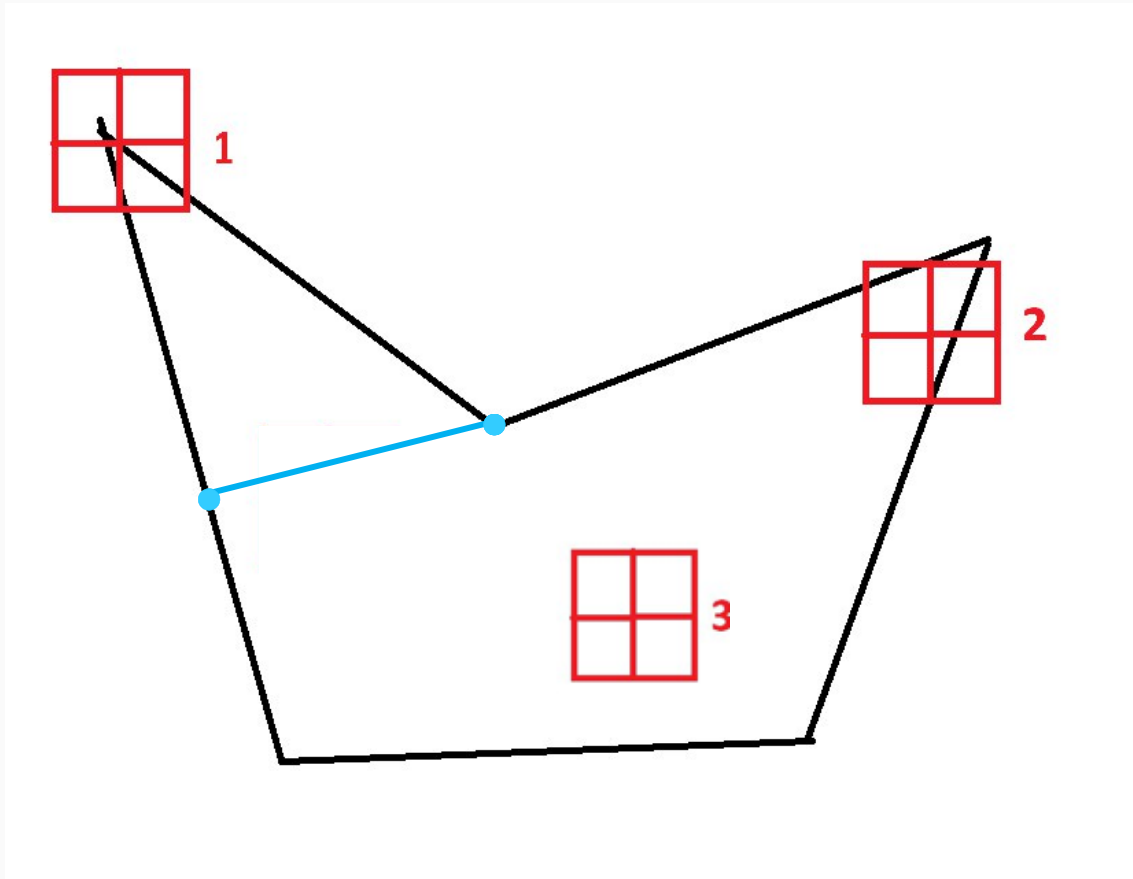


Il teorema 2.4 definisce anche che  $O(g(n))$  rappresenta il limite superiore per il tempo di esecuzione di una query, eseguita sulla struttura dati.

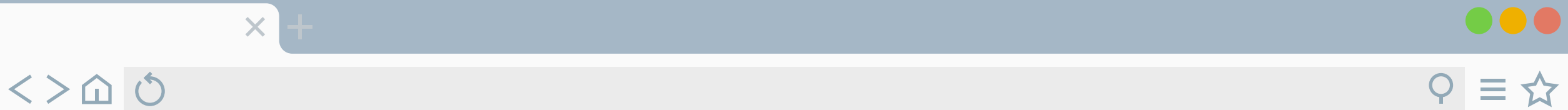
**Dimostrazione:** Per rispondere a una query  $\Gamma_q \in L_\Gamma$ , troviamo tutte le quad-celle non vuote di  $G$  che intersecano  $\Gamma_q$ :

- Le  $\oplus$  non vuote che intersecano  $\Gamma_q$  sono in numero costante  $O(1)$ , poiché è limitato da  $O(\Delta^2/\delta^2)$ , dove  $\Delta$  è il diametro di  $\Gamma$  (ossia la max distanza che si può misurare tra 2 punti qualsiasi del  $\Gamma$ )
- Queste quad-celle possono essere trovate in tempo  $O(\log n)$ .
- Per ogni quad-cella  $\oplus$  di questo tipo, troviamo  $W \in W_\Gamma$  tale che  $\oplus \cap \Gamma_q = \oplus \cap W_q$ , e interroghiamo la struttura dati  $L_W$ -RCP costruita su  $S_\oplus$  per ottenere la coppia più vicina  $\Phi_\oplus$  in  $S_\oplus \cap \Gamma_q$ , questo richiede tempo  $O(g(m_\oplus))$ .

Poiché si considerano solo un numero costante  $O(1)$  di quad-celle, il tempo per questo passo è  $O(g(n))$ .



1. Nel caso in cui la quad-cella intersechi un vertice  $q$ , la struttura dati  $L_W$ -RCP della quad-cella scelta è quella relativa al wedge con origine  $q$  ( $W_q$ ).
2. Nel caso in cui la quad-cella intersechi 2 spigoli, senza intersecare un vertice, è possibile scegliere il wedge delimitato da quei 2 spigoli
3. In tutti gli altri casi, ci saranno più wedge che rispettano la condizione di validità di intersezione rispetto al  $\Gamma q$ ; sarà indifferente a quale wedge, tra questi, riferirsi.



### Coppia lunga e coppia corta

Una volta che queste  $\Phi_{\oplus}$  sono state calcolate, prendiamo l'elemento più corto  $\phi^*$  tra di esse:

- Se la lunghezza di  $\phi^*$  è al massimo  $\delta/4$ , allora  $\phi^*$  è la coppia più vicina in  $S \cap \Gamma_Q$  per il lemma 2.3 e riportiamo semplicemente  $\phi^*$ .
- Altrimenti,  $|S \cap \Gamma_Q| = O(1)$  per il Lemma 2.3. Calcoliamo quindi gli  $O(1)$  punti in  $S \cap \Gamma_Q$  usando la struttura dati di Range-Reporting, e calcoliamo la coppia più vicina in  $S \cap \Gamma_Q$  con approccio brute force (in tempo costante). Poiché il tempo di query della struttura dati di range-reporting è  $O(\log n + k)$  e qui  $k = O(1)$ , il tempo di query complessivo è  $O(g(n) + \log n)$ , come desiderato.

Con il teorema precedente, è sufficiente fornire strutture dati RCP efficienti per query di traslazione wedge e co-wedge. Risolviamo questi problemi nelle slide seguenti.



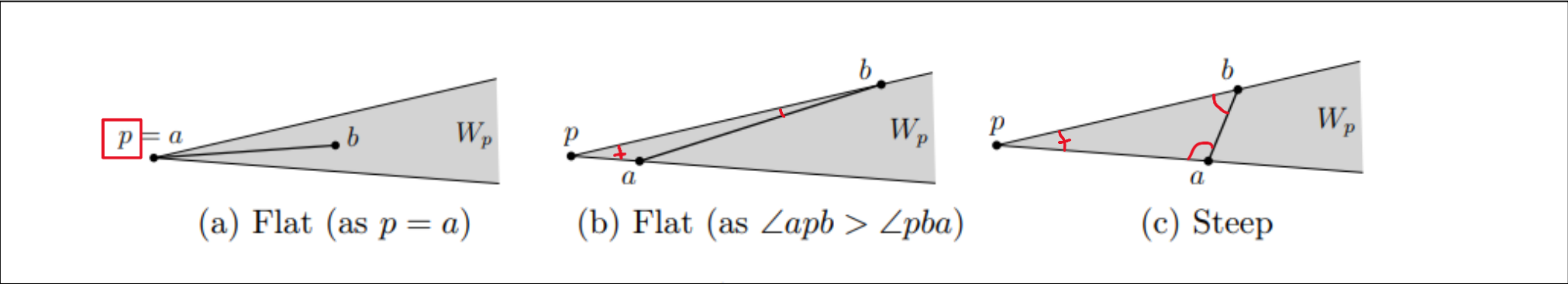
## Gestione di query basate su traslazioni di wedge

In questa parte della presentazione, forniremo una struttura dati  $L_W$ -RCP con spazio  $O(n)$  e tempo di query  $O(\log n)$ . Mostreremo un limite superiore lineare non banale per il numero di coppie candidate in  $S$  rispetto a  $L_W$ . Prima di lavorare sulla dimostrazione, stabiliamo prima un fatto semplice:

**Definizione:** Sia  $A \subseteq \mathbb{R}^2$  un insieme finito. Esiste una (unica)  $W$ -traslata più piccola (sotto l'ordinamento  $\sqsubseteq$ ) che contiene  $A$ . Inoltre, una  $W$ -traslata è la più piccola  $W$ -traslata contenente  $A$  **se e solo se** contiene  $A$  e i suoi due rami intersecano entrambi  $A$ .



**Definizione:** Sia  $\phi = (a, b)$  una coppia di punti in  $\mathbb{R}^2$  e sia  $W_p$  la più piccola regione di  $W$ -traslata contenente  $\{a, b\}$ . Se  $p \notin \{a, b\}$  e l'angolo più piccolo del triangolo  $\triangle pab$  è  $\angle apb$ , allora diciamo che  $\phi$  è *steep*; altrimenti, diciamo che  $\phi$  è *flat*.



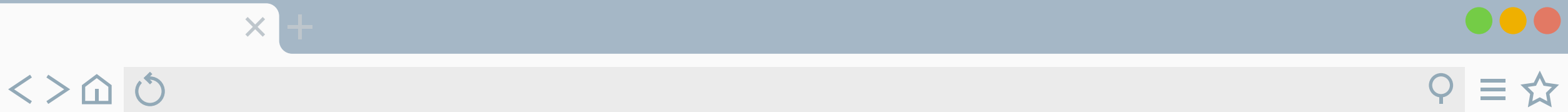
**Lemma 2.7:** Se 2 coppie candidate  $\phi, \phi' \in \Phi(S, L_W)$  si incrociano, allora o  $\phi$  o  $\phi'$  è steep

Il lemma 2.7 implica che le coppie candidate *flat* in  $\Phi(S, L_W)$  non si incrociano tra loro

Le coppie flat non possono incrociarsi a vicenda, quindi è possibile considerarle singolarmente come lati di un grafo planare, i cui vertici sono punti in  $S$ , il che fornisce un limite superiore lineare per il numero di coppie candidate flat.

- Proprietà grafo planare: un grafo è planare se lo si può disegnare nel piano senza che nessuno dei suoi archi si incroci. Questo grafo è *planare* (non ci sono lati che si intersecano tra loro). È un risultato noto nella teoria dei grafi che un grafo planare con  $n$  vertici ha al massimo un numero di lati che è lineare rispetto ad  $n$ .

Ora è sufficiente limitare il numero di coppie candidate *steep* in  $\Phi(S, L_W)$ . Sfortunatamente, due coppie candidate *steep* (o anche una coppia candidate *steep* e una coppia candidate *flat*) possono incrociarsi, facendo fallire la tesi precedente sulla non-intersezione. **Pertanto, abbiamo bisogno di nuove idee.**



**Definizione:** due coppie  $\phi, \phi' \in \Phi(S, L_W)$  sono *adiacenti* se possiamo scrivere  $\phi = (a, b)$  e  $\phi' = (a, b')$ , tale per cui  $b \neq b'$ ; chiamiamo  $\angle bab'$  l'angolo tra  $\phi$  e  $\phi'$ , chiamato  $\text{ang}(\phi, \phi')$

**Lemma 2.9:** Per coppie adiacenti  $\phi$  e  $\phi' \in \Phi(S, L_W)$ , se  $\phi$  e  $\phi'$  sono entrambi steep, allora  $\text{ang}(\phi, \phi') \geq \theta$



## Conseguenze del lemma 2.9

Per un punto  $a \in S$ , consideriamo il subset  $\Psi_a \subseteq \phi(S, L_W)$  formato da tutte le coppie candidate *steep* aventi  $a$  come punto.

Dimostriamo di seguito che  $|\Psi_a| = O(1)$ . Supponiamo sia  $\Psi_a = \{\Psi_1, \dots, \Psi_r\}$  dove  $\Psi_i = (a, b_i)$  e  $b_1, \dots, b_r$  sono ordinati per angolo-polare intorno ad  $a$ .

Per il Lemma 2.9  $\text{ang}(\Psi_i, \Psi_j) \geq \theta$  per ogni  $i, j \in \{1, \dots, r\}$ . Siccome:  $\sum_{i=1}^{r-1} \text{ang}(\Psi_i, \Psi_{i+1}) \leq 2\pi$  abbiamo che  $r \leq \frac{2\pi}{\theta} + 1 = O(1)$ . Quindi  $\sum_{a \in S} |\Psi_a| = O(n)$ , implicando che il numero di coppie candidate *steep* è lineare.

- Siccome sia il numero delle coppie candidate *steep* che *flat* è **lineare**, concludiamo che:

Lemma 2.10:  $|\phi(S, L_W)| = O(n)$ , dove  $n = |S|$

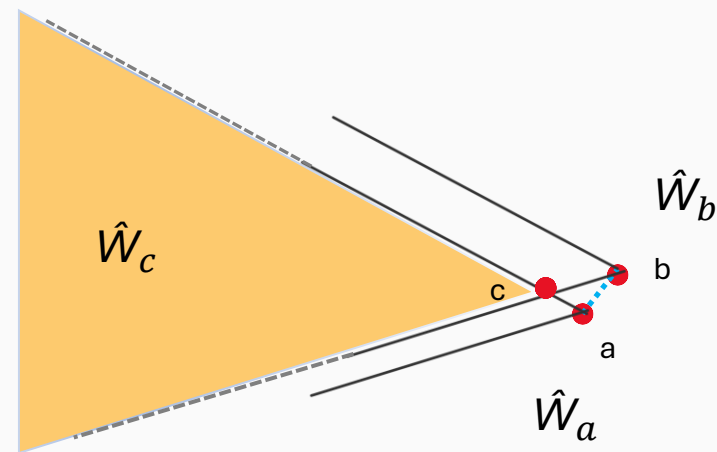
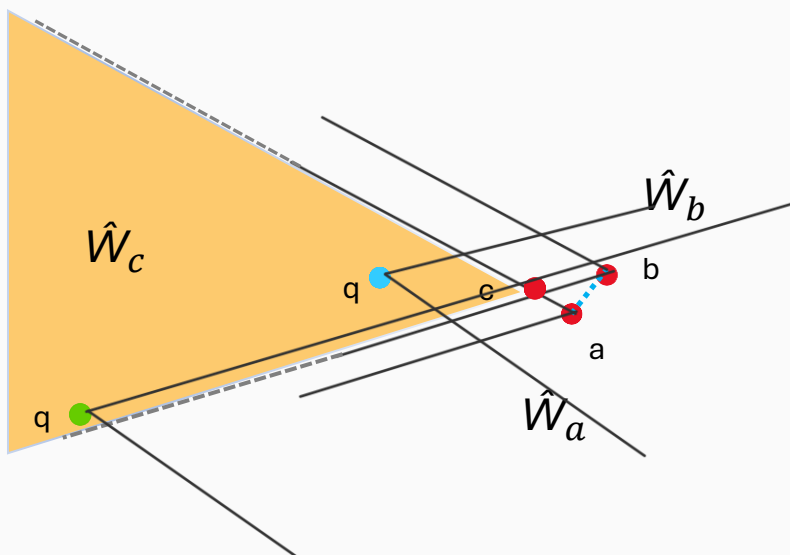
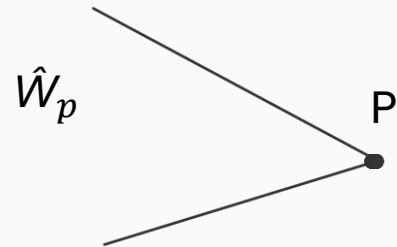
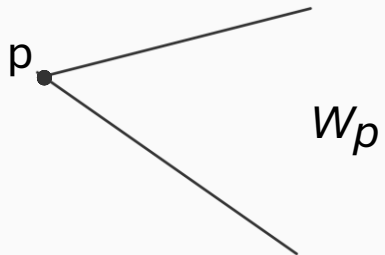


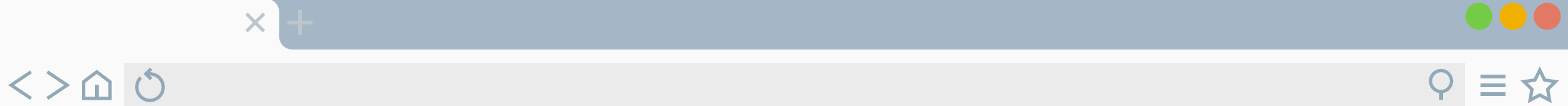
## **$L_W$ -RCP basato su suddivisione planare**

Supponiamo  $\phi(S, L_W) = \{\phi_1, \dots, \phi_m\}$  dove  $\phi_i = (a_i, b_i)$  e  $\phi_1, \dots, \phi_m$  sono in ordine crescente di lunghezza. Abbiamo  $m = O(n)$  per il Lemma 2.10. Ora ci serve una struttura dati che possa riportare, per una query  $W_q \in L_W$ , la più piccola  $i$  tale che  $a_i, b_i \in W_q$  (nota che  $\phi_i$  è la coppia più vicina in  $S$ ).

Progettiamo la struttura dati come segue: sia  $\hat{W} = \{(x,y) : (-x,-y) \in W\}$ , un wedge ottenuto ruotando  $W$  intorno all'origine di  $\pi$ . Per un punto  $p \in R^2$  è chiaro che  $a_i, b_i \in W_p$  se  $p \in \hat{W}_{a_i} \cap \hat{W}_{b_i}$ .

Siccome l'intersezione di molti, però finiti,  $\hat{W}$ -traslati è un  $\hat{W}$ -traslato allora possiamo scrivere che  $\hat{W}_{a_i} \cap \hat{W}_{b_i} = \hat{W}_{c_i}$  per qualche  $c_i \in R^2$ . Segue che  $\phi_i$  è contenuto in  $W_p$  se  $p \in \hat{W}_{c_i}$ .





Sovrapponendo  $\hat{W}_{c_1}, \dots, \hat{W}_{c_m}$  otteniamo una suddivisione planare le cui celle sono  $\Sigma_1, \dots, \Sigma_m$  dove

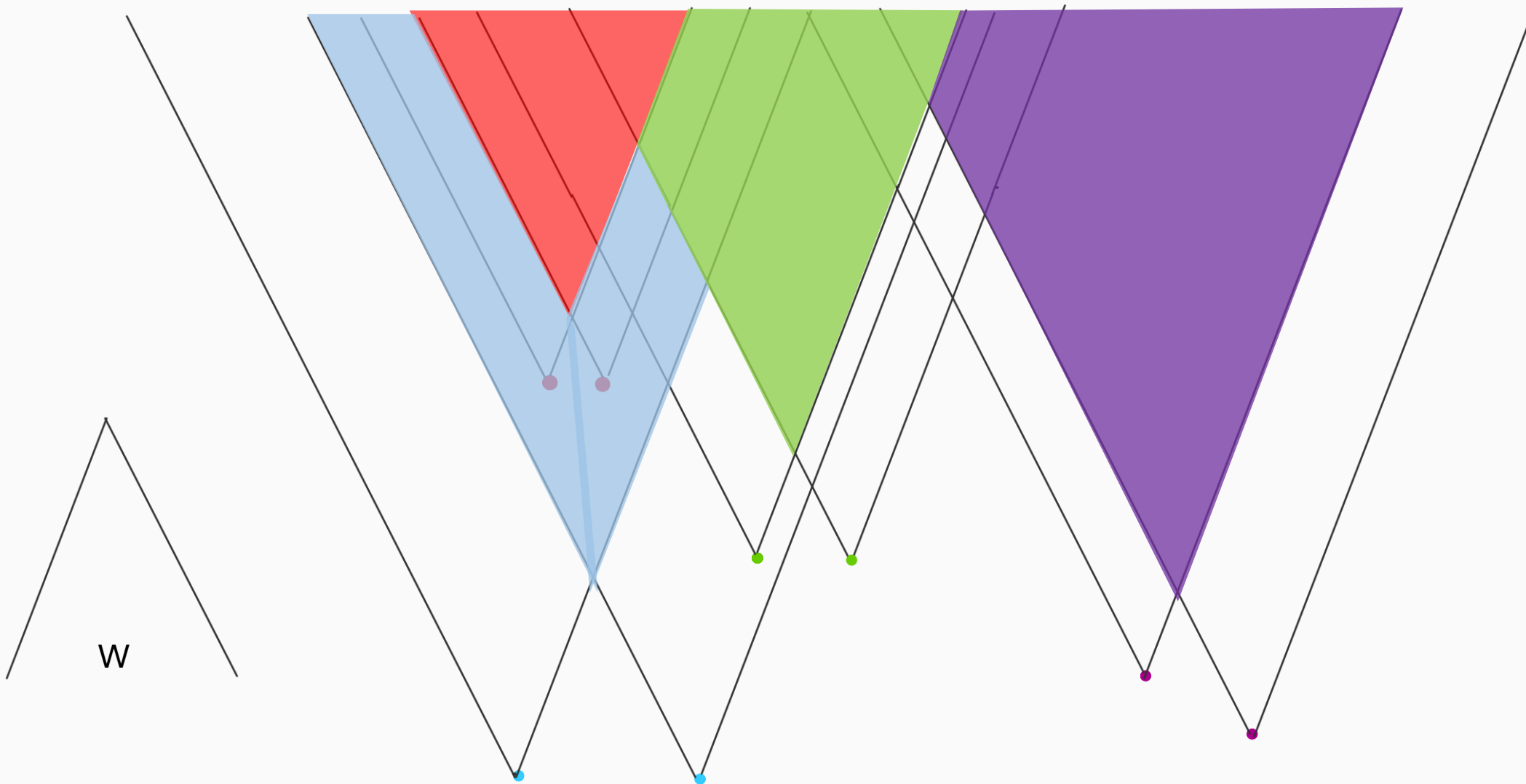
$$\Sigma_i = \hat{W}_{c_i} \setminus \bigcup_{j=1}^{i-1} \hat{W}_{c_j}.$$

Questa suddivisione ha complessità  $O(m)$  poiché sovrapporre un nuovo  $\hat{W}$ -trasalto può creare al massimo 2 nuovi vertici. **La risposta per una query  $W_q$  è i se  $q \in \Sigma_i$ .** Quindi, il problema può essere risolto costruendo su questa suddivisione una *struttura dati di point-location* con **spazio  $O(m)$**  e **tempo di query  $O(\log m)$** .

Dato che  **$m = O(n)$** , concludiamo:

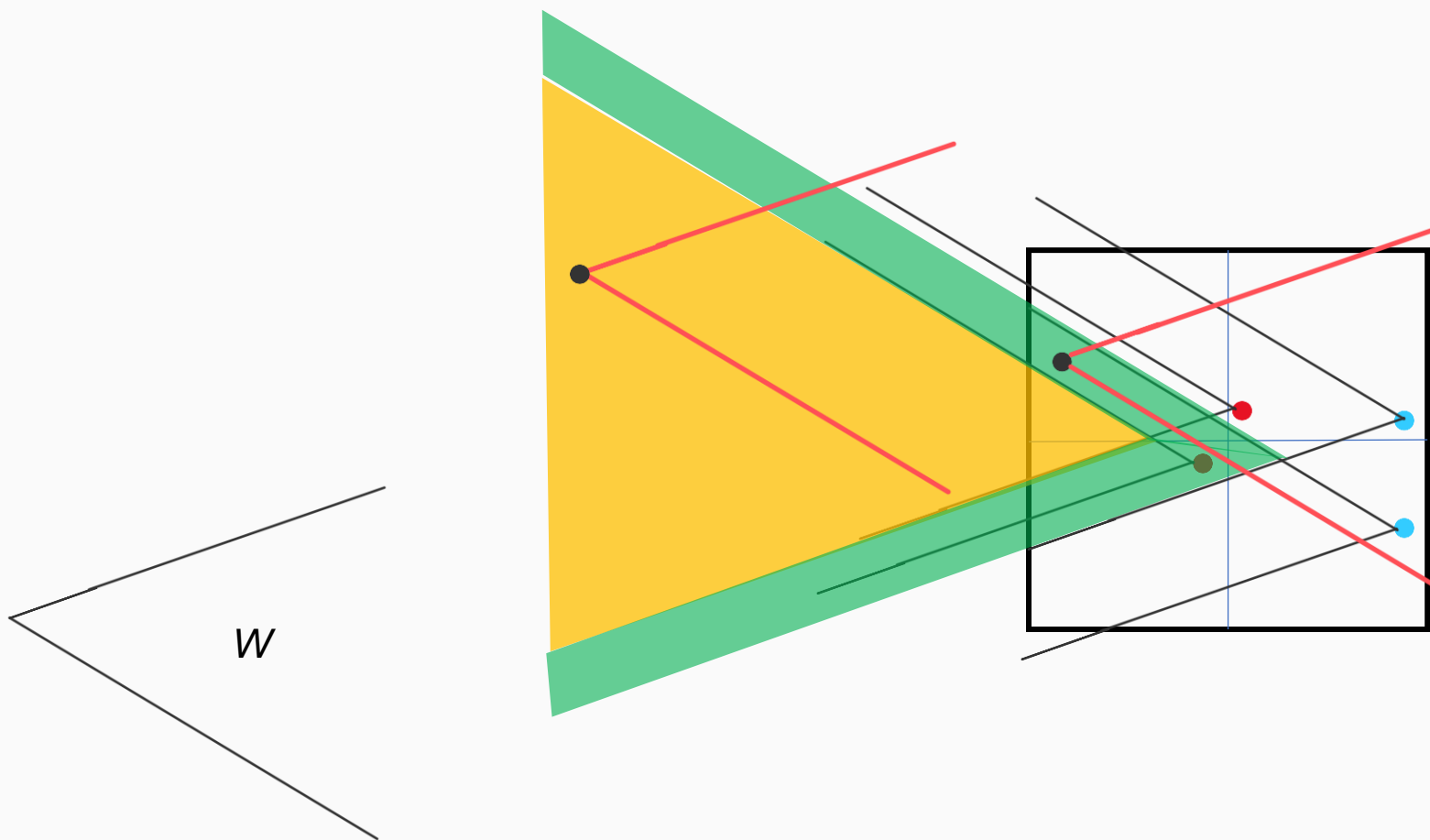
**Teorema 2.11:**

Esiste una struttura dati  $L_W$ -RCP che usa spazio  $O(n)$  e ha tempo di query  $O(\log n)$





$$\phi(S_{\oplus}, L_W) = \{\phi_1, \phi_2, \dots, \phi_{m_{\oplus}}\}$$





## Corpo convesso (smooth boundary) - Considerazioni preliminari di lavoro >

Un **corpo convesso** in  $R^2$  si riferisce a una **forma convessa compatta** con un interno non vuoto.

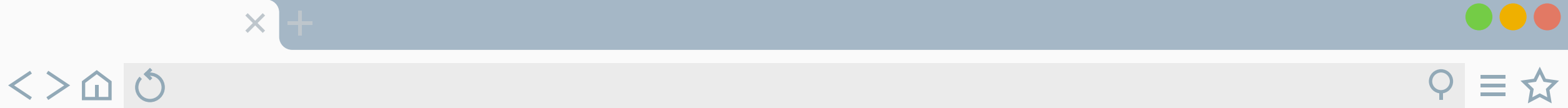
Se  $C$  è un corpo convesso in  $R^2$ , denotiamo con  $\partial C$  il bordo di  $C$ , che è un ciclo semplice, e con  $C^\circ$  l'interno di  $C$ , ovvero:

$$C^\circ = \frac{C}{\partial C}$$

L'operazione descritta è una differenza insiemistica:  $A \setminus B = \{x \in A : x \notin B\}$ , quindi, si prende tutto ciò che appartiene a  $A$ , e che non appartiene a  $B$ .

### Convessità

Un insieme  $C \subseteq R^2$  è convesso se, per ogni coppia di punti  $a$  e  $b$  contenuti in  $C$ , il segmento che li unisce è interamente contenuto in  $C$ . Matematicamente si ha che: se  $a, b \in C$ , allora  $[a, b] \subseteq C$

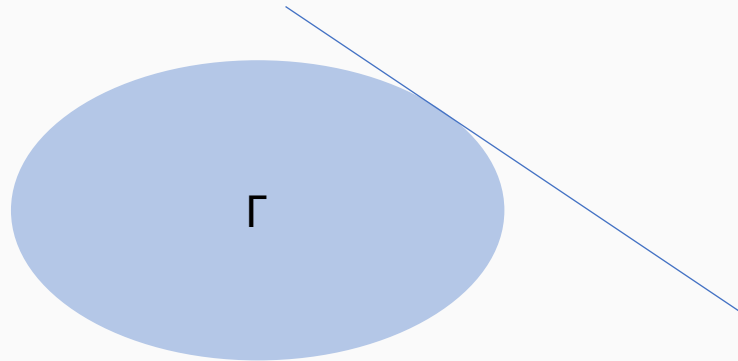


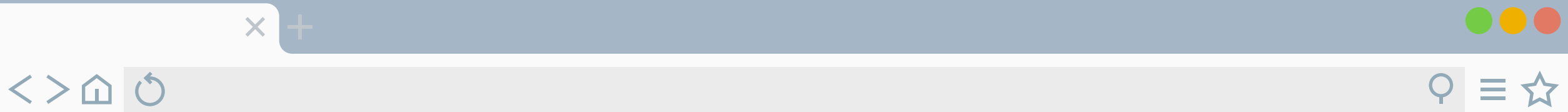
## Corpo convesso (smooth boundary) >

Sia  $\Gamma$  un corpo convesso fissato il cui bordo è *smooth*, ovvero, per ogni punto del bordo esiste un'unica linea tangente  $l$  a  $\Gamma$ . Assumiamo di poter computare in tempo costante il segmento  $\Gamma \cap l$  per qualsiasi  $l \in R^2$ . Il risultato principale è il seguente:

### Teorema 3.1:

Sia  $\Gamma$  un corpo convesso fissato *smooth* in  $R^2$ , allora esiste una struttura dati che utilizza spazio  $O(n \log n)$  e tempo di esecuzione di una query in  $O(\log^2 n)$





Sia  $S$  un dataset in  $R^2$  di dimensione  $n$ . Supponiamo che non esistano terne di punti *colineari* (non possono giacere sulla medesima linea). Esponiamo il seguente teorema:

### Teorema 3.2:

Sia  $\Gamma$  un corpo convesso *smooth*, allora esiste una costante  $\tau > 0$  (dipendente solo da  $\Gamma$ ) tale che se  $(a, b), (c, d) \in \phi(S, L_\Gamma)$  siano due coppie la cui lunghezza è al massimo  $\tau$ , allora i segmenti  $[a, b], [c, d]$  non si incrociano

**Definizione:** siano  $C$  e  $D$  due corpi convessi in  $R^2$  tali che  $C \cap D \neq \emptyset$ . Diciamo che  $C$  e  $D$  si *intersecano chiaramente* se  $\partial C \cap D$  e  $\partial D \cap C$  sono entrambi connessi.

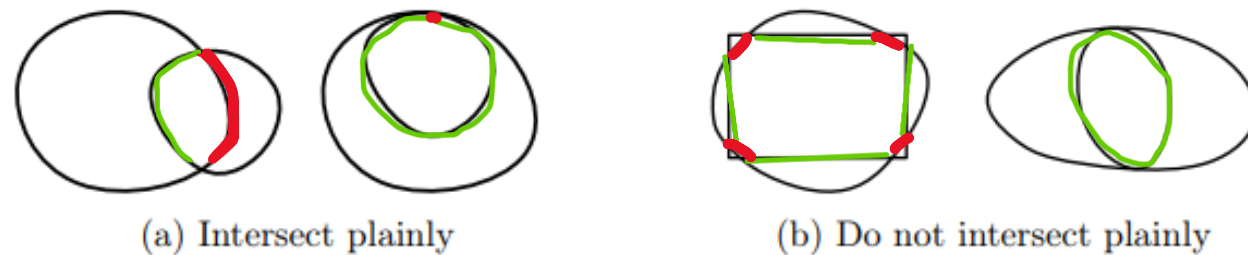


Figure 7: An illustration the concept of “intersect plainly”

### Teorema 3.3:

Sia  $C$  un corpo convesso in  $R^2$ , e  $p_1, p_2, p'_1, p'_2$  siano 4 punti (non necessariamente distinti), tali che  $I \neq \emptyset$  e  $I' \neq \emptyset$ , dove  $I = C_{p_1} \cap C_{p_2}$  e  $I' = C_{p'_1} \cap C_{p'_2}$ . Supponiamo che qualsiasi terna tra  $p_1, p_2, p'_1, p'_2$ , non sia colineare, al meno che 2 punti non coincidano. Se i segmenti  $[p_1, p_2], [p'_1, p'_2]$  non si incrociano e  $I \cap I' \neq \emptyset$ , allora  $I$  e  $I'$  si *intersecano chiaramente*

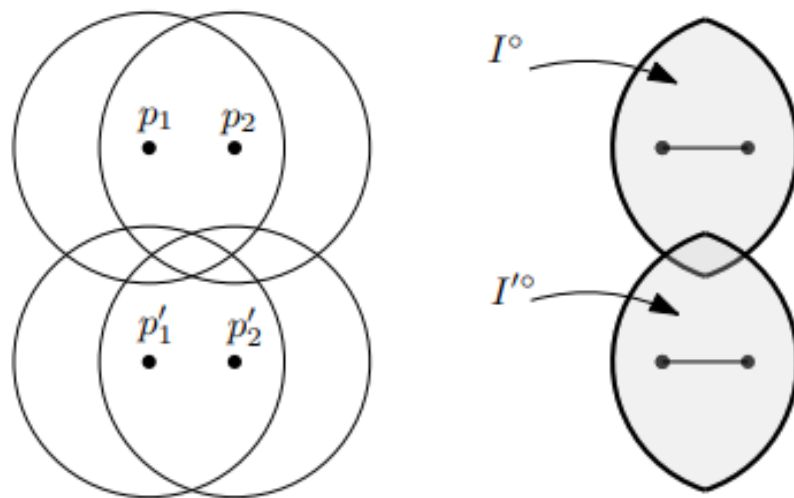
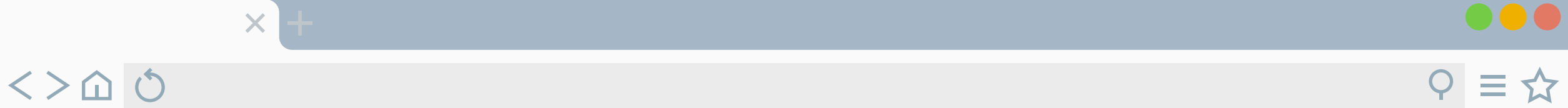


Figure 8: Illustration of Theorem 3.3 when  $C$  is a disc



## Gestione delle query >

La struttura dati proposta consiste di 2 parti: **D1 e D2**.

- D1 gestisce le query per cui la lunghezza della risposta (coppia più vicina) è *corta*
- D2 gestisce le query per cui la lunghezza della risposta è *lunga*

## Gestione delle query – Caso risposta *corta*

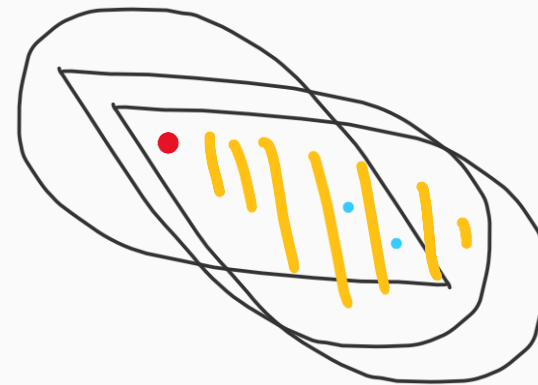
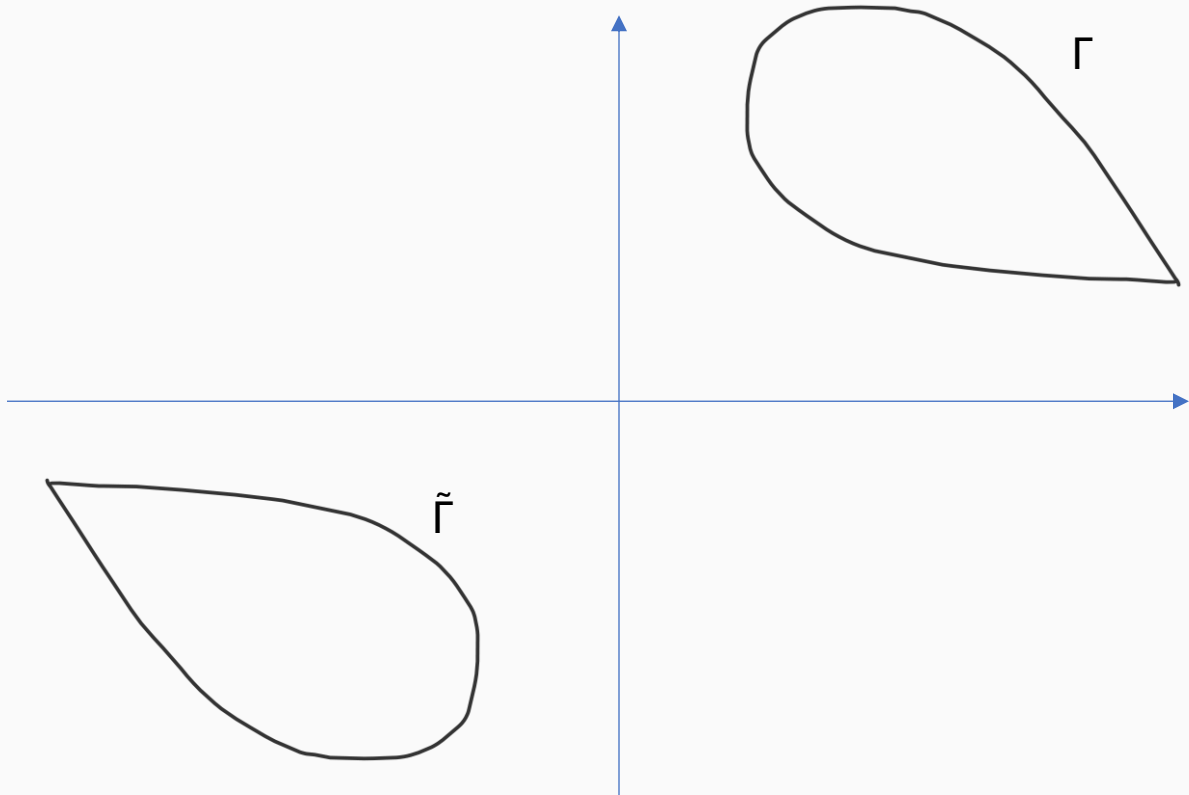
Sia  $\tau > 0$  la costante tale per cui, 2 coppie candidate qualsiasi non si incrociano. Per una query  $\Gamma_q \in L_\Gamma$ , D1 riporta la coppia più vicina in  $S \cap \Gamma_q$  se la sua lunghezza è al massimo  $\tau$ , riporta niente altrimenti.

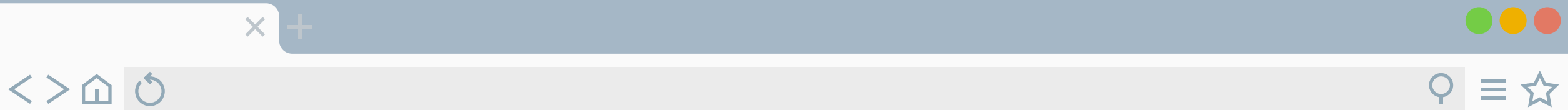
Sia  $\phi_{\leq \tau} \subseteq \phi(S, L_\Gamma)$  la sottocollezione composta dalle coppie candidate di lunghezza al massimo  $\tau$ , e supponiamo  $m = |\phi_{\leq \tau}|$ .

Si ha  $m = O(n)$ , poiché il grafo  $G = (S, \phi_{\leq \tau})$  è *planare* (per il Teorema 3.2).

**Definizione:** sia  $\tilde{\Gamma} = \{(x,y) \in R^2 : (-x, -y) \in \Gamma\}$ .

Per una coppia  $\theta = (a, b) \in \Phi_{\leq \tau}$ , scriviamo  $I^\theta = \tilde{\Gamma}_a \cap \tilde{\Gamma}_b$ . Allora  $\theta$  è contenuto nella range query  $\Gamma_q$  se  $q \in I^\theta$





## MDS

Sia  $\Psi = (\theta_1, \dots, \theta_r) \subseteq \phi_{\leq \tau}$  e  $U = \bigcup_{\theta \in \Psi} I^\theta$ . Una MDS è una struttura dati di *membership* che può decidere per una data query  $\Gamma_q$  se quest'ultima contiene una coppia in  $\Psi$  oppure no.

Dal Teorema 3.2, per qualsiasi  $\theta_i, \theta_j$  queste non si incrociano. Inoltre non esistono 3 punti in  $S$  colineari per assunzione.

Per il Teorema 3.3,  $I^{\theta_i}$  e  $I^{\theta_j}$  se si intersecano, si intersecano *chiaramente*. Segue che  $\{I^{\theta_i}, \dots, I^{\theta_r}\}$  è una famiglia di pseudodischi e quindi la complessità della loro unione è  $O(r)$ .

- In questo modo è possibile utilizzare una struttura dati ottimale di point-location per decidere se un punto è contenuto in  $U$  in tempo  $O(\log r)$  e con spazio  $O(r)$ .
- Utilizzeremo una MDS per costruire la struttura dati voluta D1.



# D1

Assumiamo  $\Psi$  una sottocollezione di  $\phi_{\leq \tau}$  ordinata per lunghezza crescente delle coppie. Sia  $D1(\Psi)$  strutturata come segue. Se  $r = 1$  allora  $D1(\Psi)$  memorizza soltanto l'unica coppia  $\theta_1$ ; se invece  $r > 1$ , siano  $\Psi_1 = (\theta_1, \dots, \theta_{r/2})$  e  $\Psi_2 = (\theta_{(r/2)+1}, \dots, \theta_r)$ . Allora  $D1(\Psi)$  consiste di 3 parti:

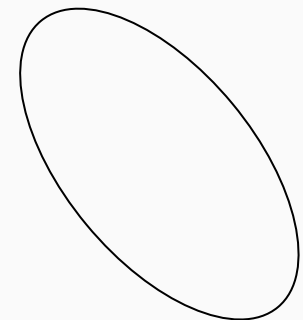
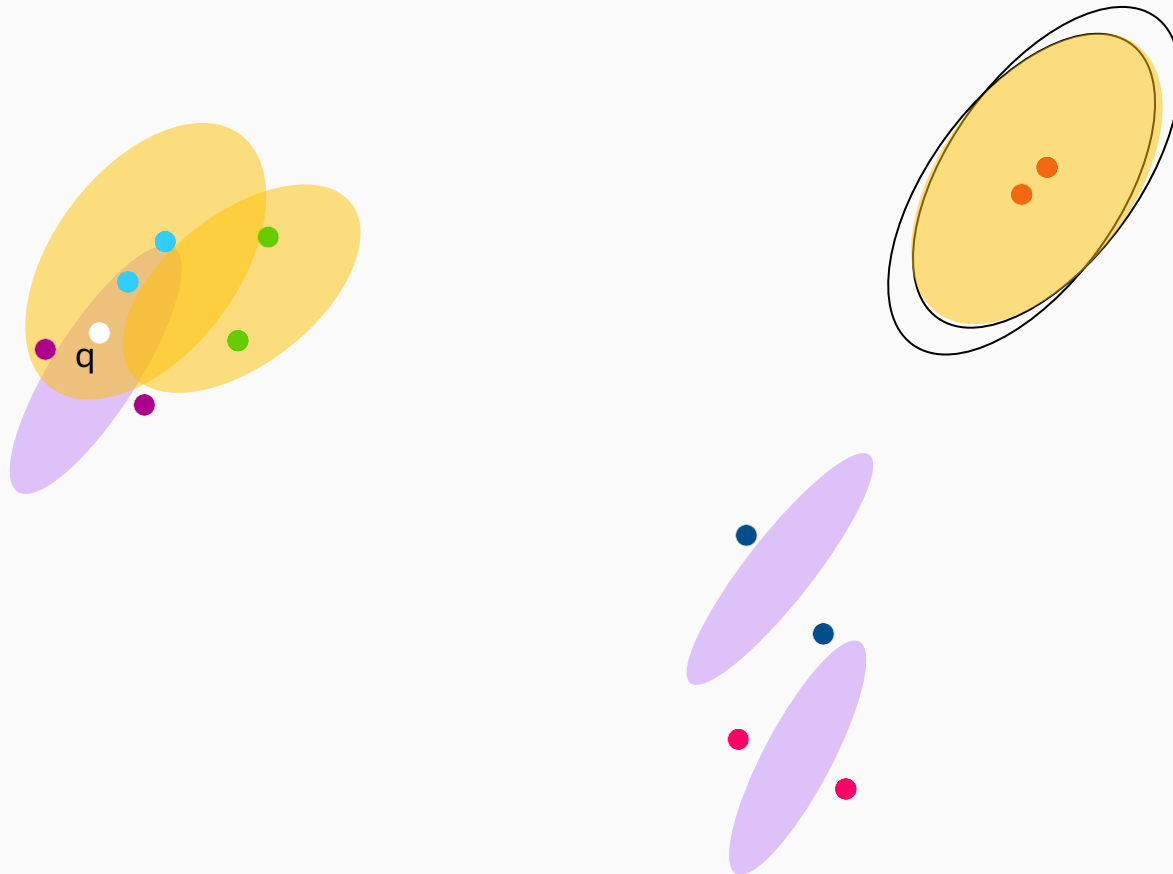
1.  $D1(\Psi_1)$
  2.  $D1(\Psi_2)$
  3.  $M(\Psi_1)$
- } Definite ricorsivamente

- Si esegue la query su  $M$ , per verificare se  $q \in \bigcup_{\theta \in \Psi_1} I^\theta$ 
  - I. Se l'esisto è positivo allora  $\theta^*$  deve stare in  $\Psi_1$ , quindi, ricorsivamente si esegue una query su  $D1(\Psi_1)$
  - II. Altrimenti, ricorsivamente si esegue una query su  $D1(\Psi_2)$

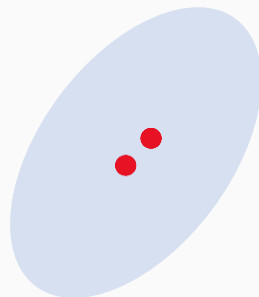
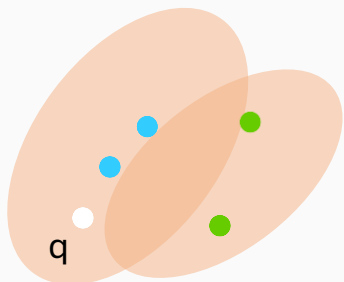
**Definiamo quindi  $D1 = D1(\phi_{\leq \tau})$**

$$\Psi = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) \longrightarrow \Psi_1 = (\theta_1, \theta_2, \theta_3), \Psi_2 = (\theta_4, \theta_5, \theta_6)$$

$$D1(\Psi) = \{D1(\Psi_1), D1(\Psi_2), M(\Psi_1)\}$$



$$\Psi_1 = (\theta_1, \theta_2, \theta_3) \longrightarrow \Psi_{1.1} = (\theta_1), \Psi_{2.3} = (\theta_2, \theta_3)$$

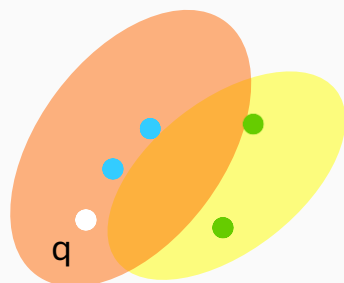


$$D1(\Psi) = \{D1(\Psi_1), D1(\Psi_2), M(\Psi_1)\}$$

$$D1(\Psi_1) = \{D1(\Psi_{1.1}), D1(\Psi_{2.3}), M(\Psi_{1.1})\}$$



$$\Psi_{2.3} = (\theta_2, \theta_3) \longrightarrow \Psi_{\theta_2} = (\theta_2), \Psi_{\theta_3} = (\theta_3)$$



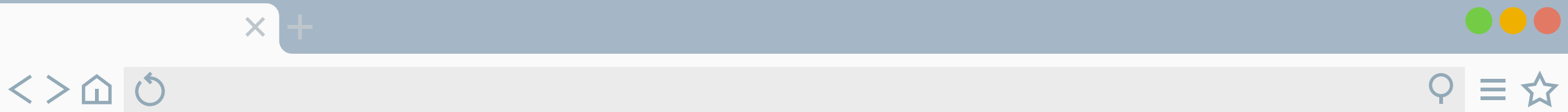
$$D1(\Psi) = \{D1(\Psi_1), D1(\Psi_2), M(\Psi_1)\}$$

$$D1(\Psi_1) = \{D1(\Psi_{1.1}), D1(\Psi_{2.3}), M(\Psi_{1.1})\}$$

$$D1(\Psi_{2.3}) = \{D1(\Psi_{\theta_2}), D1(\Psi_{\theta_3}), M(\Psi_{\theta_2})\}$$



$$D1(\Psi_{\theta_2}) = \{\theta_2\}$$



Definiamo **sp(r)** e **qt(r)** come lo spazio massimo e il tempo di query massimo di D1 per qualsiasi sottocollezione  $\Psi \subseteq \phi_{\leq \tau}$  con  $|\Psi| = r$ .

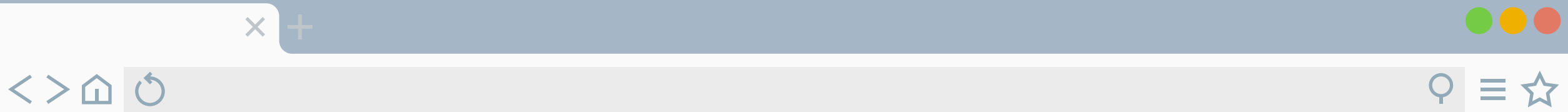
Assumiamo le coppie che compongono  $\Psi$  ordinate per lunghezza crescente. Scriviamo  $\Psi_1 = (\theta_1, \dots, \theta_{r/2})$  e  $\Psi_2 = (\theta_{(r/2)+1}, \dots, \theta_r)$ . Sappiamo che lo spazio utilizzato da  $M(\Psi_1)$  è  $O(r)$  il che implica la seguente ricorrenza:

$$\mathbf{sp(r) \leq 2 \, sp(r/2) + O(r)}$$

La quale si risolve in  $sp(r) = O(r \log r)$ . Per quanto riguarda il tempo di query sappiamo che per  $M(\Psi_1)$  impiega  $O(\log r)$ ; abbiamo quindi:

$$\mathbf{qt(r) \leq qt(r/2) + O(\log r)}$$

Che si risolve in  $qt(r) = O(\log^2 r)$ . Siccome  $D1 = D1(\phi_{\leq \tau})$  concludiamo che lo spazio utilizzato da D1 è  $O(m \log m)$  e il tempo di query è  $O(\log^2 m)$ , dove  $m = |\phi_{\leq \tau}|$



## Gestione delle query – Caso risposta *lunga*

Se D1 non riesce a riportare una risposta valida alla query, allora la lunghezza della coppia più vicina in  $S \cap \Gamma_q$  è maggiore di  $\tau$ , quindi  $|S \cap \Gamma_q| = O(1)$  per il lemma 1.1.

In questo caso viene usata D2, che semplicemente riporta tutti i punti in  $S \cap \Gamma_q$  computando la coppia più vicina con un approccio brute-force che richiede  $O(1)$  tempo (essendo il numero di coppie costante).

Riutilizzando  $\tilde{\Gamma}$  sappiamo che, per qualsiasi  $a \in R^2$ ,  $a \in \Gamma_q$  se  $q \in \tilde{\Gamma}_a$ . Quindi il problema è equivalente a riportare le regioni  $\{\tilde{\Gamma}_a : a \in S\}$  che contengono  $q$ .

Definiamo la *depth* di un punto  $p$ ,  $\text{dep}(p)$ , come il numero di regioni  $\tilde{\Gamma}_a$  che contengono  $p$ . Sia  $A$  la disposizione delle regioni  $\tilde{\Gamma}_a$  e  $k$  una costante sufficientemente grande. I livelli minori uguali a  $k$  di  $A$ , chiamati  $A_{\leq k}$ , sono la sottodisposizione di  $A$  contenente le regioni  $R_{\leq k} = \{p \in R^2 : \text{dep}(p) \leq k\}$ .

Per il teorema 3.3 se due regioni si intersecano allora si intersecano chiaramente, il che implica che  $A$  è una famiglia di pseudodischi. Concludiamo con il seguente lemma:

Lemma 3.4: La complessità di  $A_{\leq k}$  è  $O(n)$  per una costante  $k$



## D2

Per il lemma 3.4 possiamo costruire una struttura dati di point location su  $A_{\leq k}$  con spazio  $O(n)$  e tempo di query  $O(\log n)$ .

Associamo a ogni cella  $\Delta$  di  $A_{\leq k}$  le (massimo  $k$ ) regioni contenenti  $\Delta$ ; dove una cella  $\Delta$  è costruita in modo tale che tutti i punti al suo interno siano contenuti dallo stesso numero di regioni di  $A_{\leq k}$ .

Possiamo quindi usare la struttura dati di point location, a partire da una query  $\Gamma_q$ . Siccome  $|S \cap \Gamma_q| = O(1)$  e  $k$  è abbastanza grande, avremo che  $|S \cap \Gamma_q| \leq k$  e quindi  $q$  è in  $A_{\leq k}$ .

Troviamo quindi in  $O(\log n)$  le celle  $\Delta$  contenenti  $q$ . Allora le regioni associate alle celle  $\Delta$  sono esattamente quelle contenenti  $q$ .

- Otteniamo così la struttura dati desiderata D2 con spazio  $O(n)$  e tempo di query  $O(\log n)$ .
- **Combinando D2 con D1 il teorema 3.1 è dimostrato.**



**Grazie dell'ascolto !**