Il Problema del Dispiegamento Distribuito di Sensori Mobili ovvero Il Diagramma di Voronoi

Prof. Tiziana Calamoneri
Corso di Algoritmi per le reti
A.A. 2012/13

1

Il Problema

2

Il Problema (1)

- Abbiamo già parlato delle *reti di sensori mobili…*
- … e del *problema del dispiegamento.*
- Una soluzione centralizzata non è sempre auspicabile:
  - E' richiesta la connessione con un server centrale
  - Sono richieste lunghe attese
  - La soluzione non è fault-tolerant
- La possibiltà di movimento facilita i sensori ad auto-dispiegarsi a partire da una configurazione iniziale qualunque fino a sparpagliarsi su tutta l'AoI garantendo la copertura totale.

Il Problema (2)

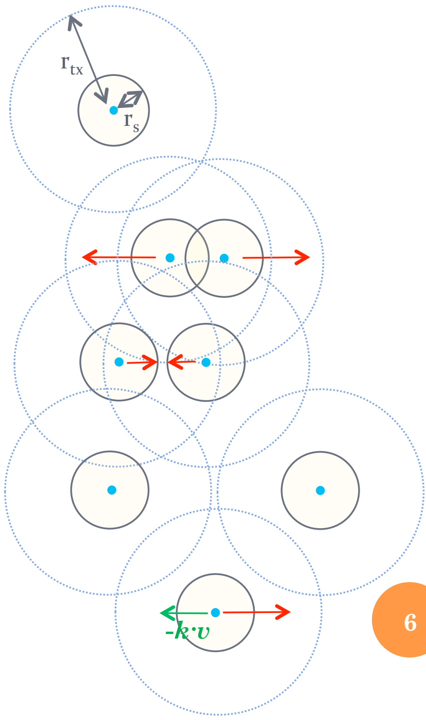
- L'auto-dispiegamento è necessario quando l'ambiente è "ostile"
  - Luoghi contaminati
  - Incendi
  - Campi di guerra…
- I sensori dovrebbero posizionarsi e trasmettere al più presto informazioni sull'ambiente.

IL PROBLEMA (3)

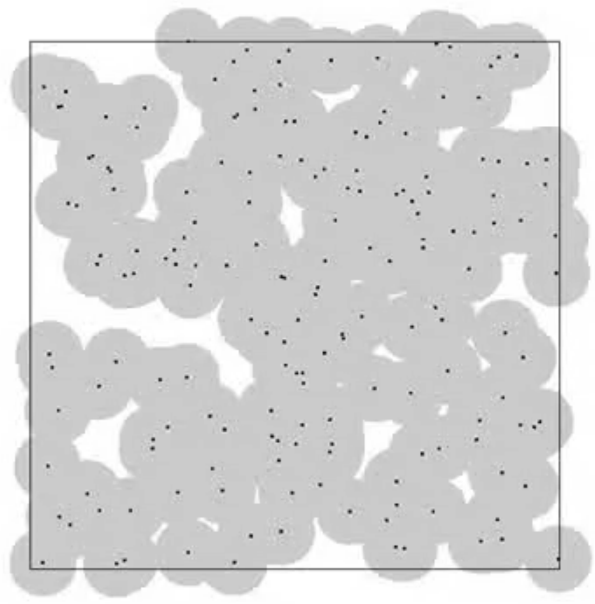
- Oss. Il Problema del dispiegamento è in stretta relazione con il classico problema di geometria computazionale chiamato *art gallery* problem.
- In tale problema si cerca di determinare, in un ambiente poligonale, il min numero di telecamere che si devono installare per garantire che l'intera sala sia sorvegliata.
- Esistono diversi algoritmi per risolvere l' art gallery problem, ma tutti assumono una perfetta conoscenza a priori dell'ambiente.

UN APPROCCIO: FORZE VIRTUALI (1)

- L'idea: i sensori sono delle particelle cariche (forza magnetica) e dotati di massa (forza gravitazionale)
- due sensori si respingono se sono troppo vicini
- due sensori si attraggono se sono lontani ma comunque in comunicazione
- due sensori si ignorano se non sono in comunicazione (troppo lontani)
- attrito per fermare le oscillazioni



UN APPROCCIO: FORZE VIRTUALI (2)



UN APPROCCIO: FORZE VIRTUALI (3)

Punti di debolezza:

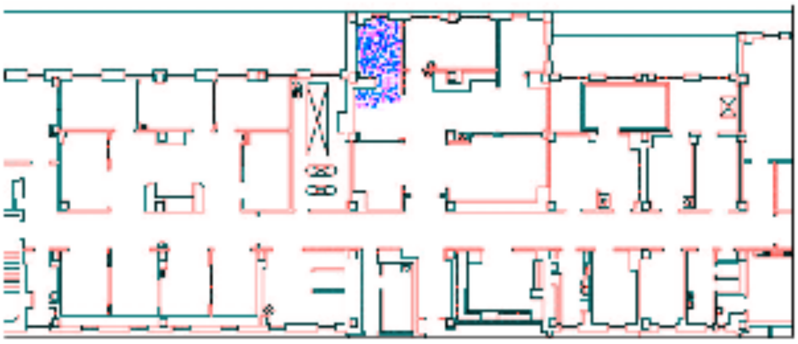
- Necessario settaggio dei parametri in modo arbitrario
- Oscillazione dei sensori
    - Forze di attrito
    - Stopping conditions
- Effetto “carta moschicida” del bordo e degli ostacoli (in alcune versioni, ad es. quando non c’è forza attrattiva)
- …

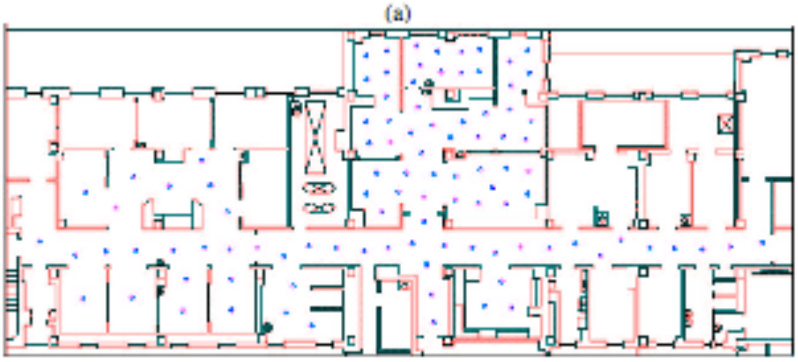
UN APPROCCIO: FORZE VIRTUALI (4)

Punti di debolezza (segue):

- I sensori non passano attraverso le porte e i corridoi

TESINA


(a)

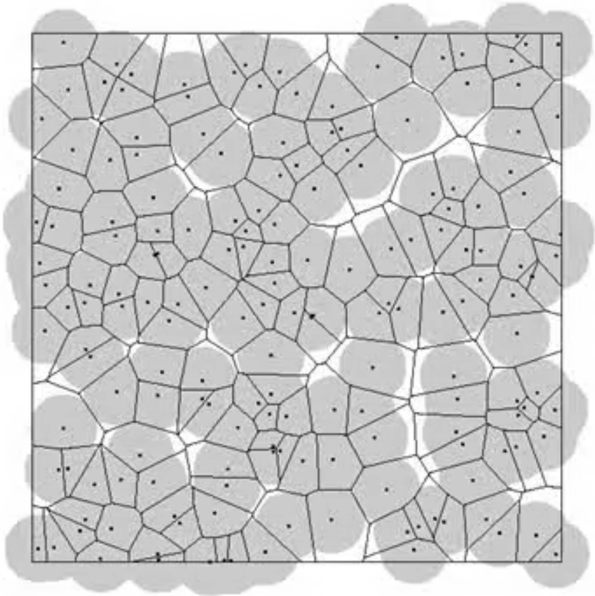

(b)

UN PROTOCOLLO BASATO SUI DIAGRAMMI DI VORONOI

Idea:

- Ogni sensore deve prendersi carico di una porzione di piano e spostarsi per cercare di coprirla al meglio
- Il sensore può considerarsi "soddisfatto" se:
  - Copre completamente la sua porzione oppure
  - Tutto il suo cerchio di sensing è impegnato per coprire la sua porzione
- Se un sensore non è "soddisfatto" si deve spostare per coprire meglio.
- Le porzioni possono essere assegnate seguendo il diagramma di Voronoi.

TESINA



Il Diagramma di Voronoi

Diagramma di Voronoi (1)

- Def. di Diagramma di Voronoi:
  - $\mathscr{P}$: insieme di n punti distinti nel piano
  - VD $(\mathscr{P})$: partizione del piano in n celle V_i tale che:
    - ogni V_i contiene esattamente un punto
    - se un punto del piano Q è in un V_i allora $dist(Q, P_i) < dist(Q, P_j)$ per ogni $P_j \in \mathscr{P}$, $j \neq i$.

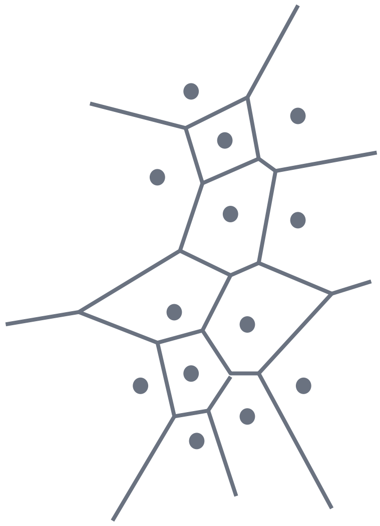


Diagramma di Voronoi (2)

- In altre parole: VD $(\mathscr{P})$ è una partizione del piano in regioni convesse $\{V_1, \ldots, V_n\}$, ogni V_i contiene esattamente un punto $P_i \in \mathscr{P}$ tale che per ogni altro punto in V_i il suo punto più vicino in $\mathscr{P}$ è P_i.

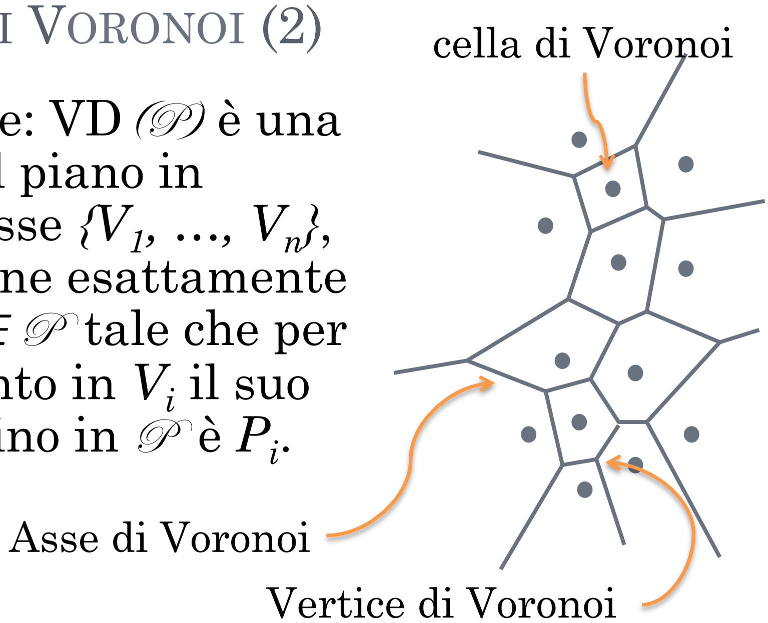


Diagramma di Voronoi (3)

I diagrammi di Voronoi sono stati usati:

- dagli antropologi per descrivere le regioni di influenza di differenti culture;
- dai cristallografi per spiegare la struttura di certi cristalli e metalli;
- dai botanici per studiare la competizione tra le piante;
- dagli economisti per modellare i mercati;
- …

Diagrammi di Voronoi (4)

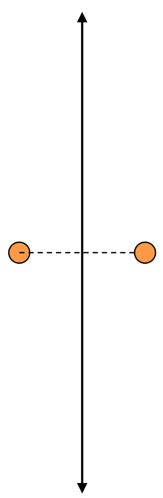
- Un uso informale dei diagr. di Voronoi si può far risalire a Cartesio (1644).
- Dirichlet ha usato diagr. di Voronoi nei suoi studi sulle forme quadratiche nel 1850.
- Il fisico inglese Snow li ha usati nel 1854 per illustrare le dinamiche del contagio del colera.
- I diagr. di Voronoi sono stati chiamati così in onore del matematico russo Georgy F. Voronoi, che li ha definiti e studiati nello spazio *n*-dimensionale nel 1908.
- Essi sono chiamati *poligoni di Thiessen* in meteorologia in onore del meteorologo americano Alfred H. Thiessen; *celle di Wigner-Seitz* in fisica della materia, *domini fondamentali* in teoria dei gruppi e *poligoni fondamentali* in topologia.

DIAGRAMMA DI VORONOI (5)

Diagramma di Voronoi di un punto

DIAGRAMMA DI VORONOI (6)

Diagramma di Voronoi di due punti



Asse che si estende all'infinito in entrambe le direzioni, e genera due semipiani

DIAGRAMMA DI VORONOI (7)

Diagramma di Voronoi di punti collineari

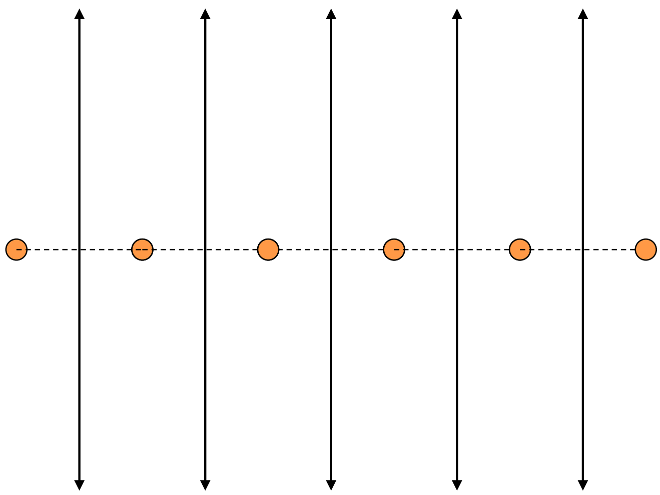
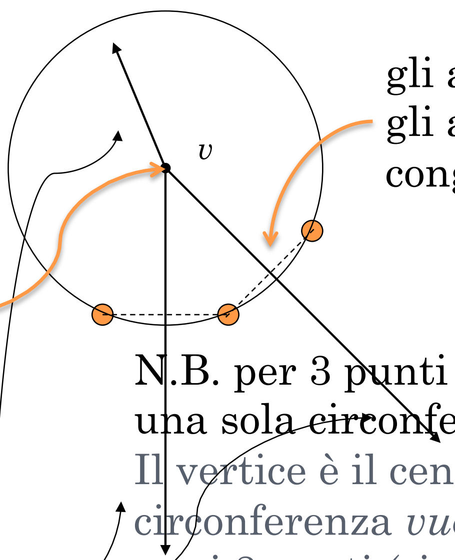


DIAGRAMMA DI VORONOI (8)

Diagramma di Voronoi di 3 punti non collineari



gli assi di Voronoi sono gli assi dei segmenti congiungenti i punti

Il vertice ha grado 3

N.B. per 3 punti passa una ed una sola circonferenza

Il vertice è il centro di una circonferenza *vuota* che passa per i 3 punti (circoscritta al triangolo generato dai 3 punti)

Semirette

DIAGRAMMA DI VORONOI (9)

Diagramma di Voronoi di 4 punti non collineari

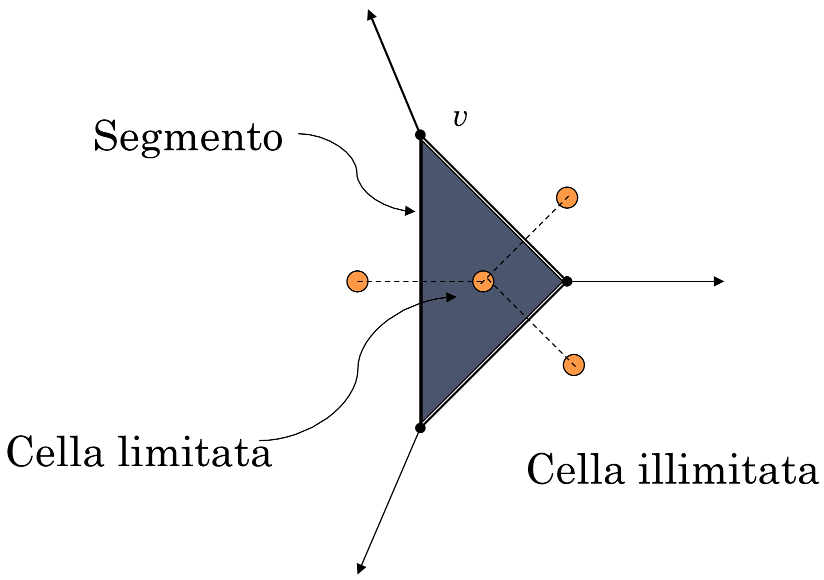
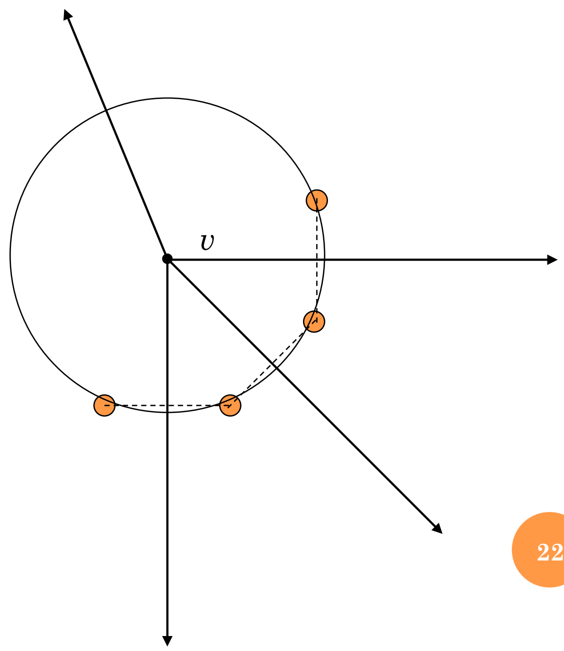


DIAGRAMMA DI VORONOI (10)

Non sempre 4 punti non collineari generano una cella limitata:

Assunzione di Posizione generale: 3 punti non sono collineari e 4 punti non sono cocircolari
Così tutti i vertici hanno grado 3!

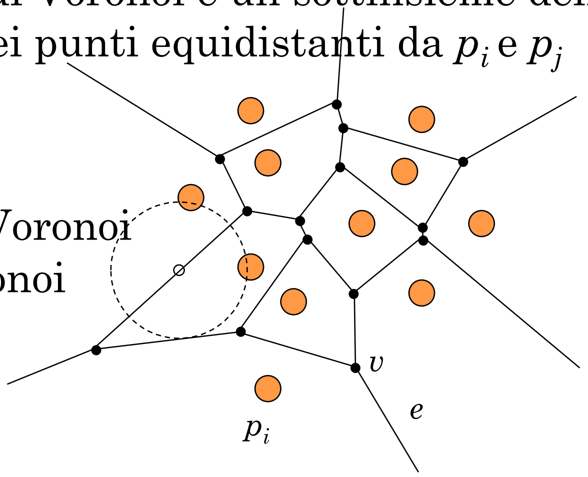


PROPRIETÀ DEL DIAGR. DI VORONOI (1)

Un punto q giace sul segmento di Voronoi tra p_i e p_j *sse* il più grande cerchio vuoto centrato in q tocca solo p_i e p_j

– Un segmento di Voronoi è un sottinsieme dell'asse, cioè del luogo dei punti equidistanti da p_i e p_j

p_i : punti di $\mathscr{P}$
e : segmento di Voronoi
v : verice di Voronoi

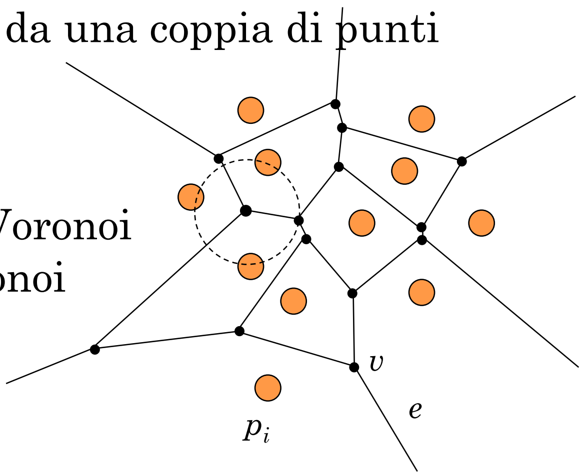


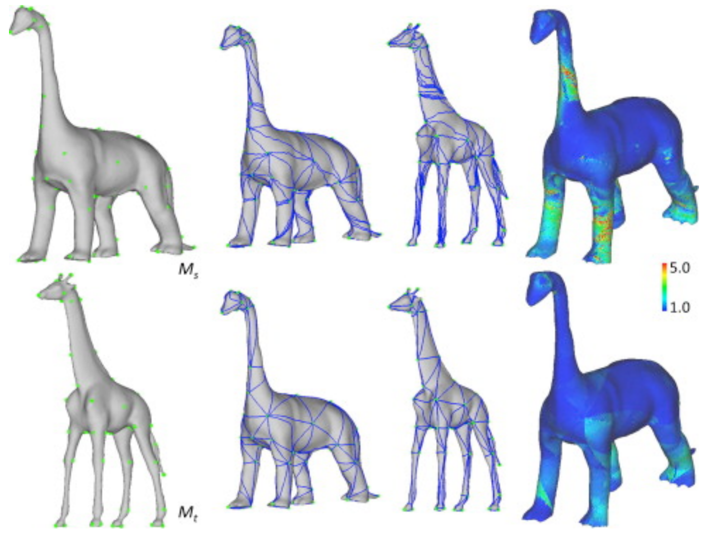
PROPRIETÀ DEL DIAGR. DI VORONOI (2)

Un punto q è un vertice *sse* il più grande cerchio vuoto centrato in q tocca almeno 3 punti di $\mathscr{P}$

Un vertice di Voronoi è l'intersezione di 3 o più assi, ciascuno generato da una coppia di punti

p_i : punti di $\mathscr{P}$
e : segmento di Voronoi
v : verice di Voronoi

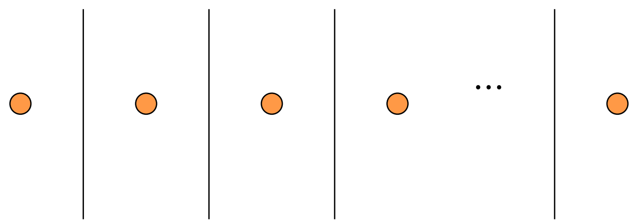




Complessità del Diagramma di Voronoi

Complessità del Diagramma (1)

- Th.: Per $n \geq 3$, $|v| \leq 2n - 5$ e $|e| \leq 3n - 6$
- Dim.: (Caso facile)



punti collineari → $|v| = 0,\ |e| = n - 1$

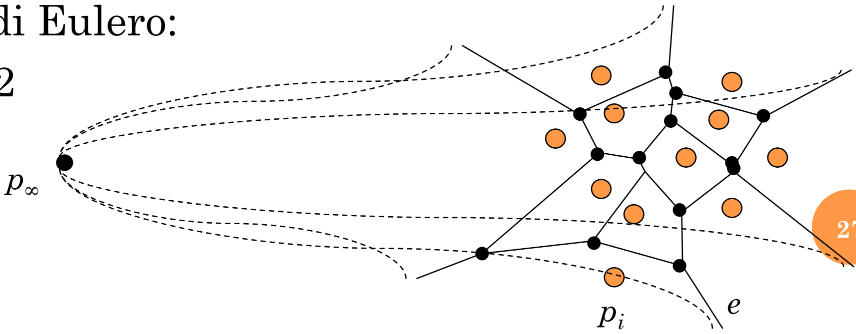
Complessità del Diagramma (2)

Segue dim. del Th.: Per $n \geq 3$, $|v| \leq 2n - 5$ e $|e| \leq 3n - 6$

Dim.: (Caso generale)

- Problema: un diagr. di Voronoi non può essere considerato un grafo perché ha archi e facce di dimensione infinita
- Soluzione: aggiungiamo un punto fittizio
- Ora il diagr. di Voronoi è un grafo planare e connesso → formula di Eulero:

  $|v| - |e| + f = 2$



Complessità del Diagramma (3)

Segue dim. del Th.: Per $n \geq 3$, $|v| \leq 2n - 5$ e $|e| \leq 3n - 6$

$f = n+1$. La formula di Eulero diventa:

$|v| - |e| + n + 1 = 2$ (1)

Inoltre: $\sum_{v \in VD} \deg(v) = 2|e|$

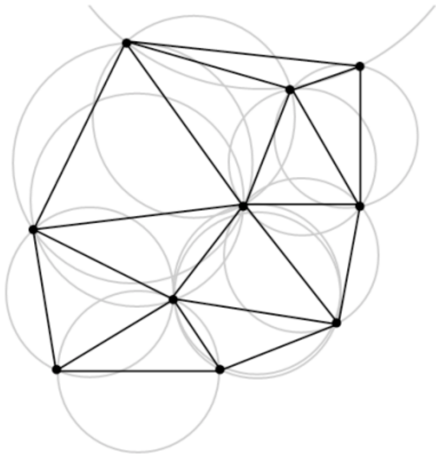
poiché $\deg(v) \geq 3$ → $2|e| \geq 3|v|$ (2)

Unendo (1) e (2):

$|v| \leq 2n-5$

$|e| \leq 3n-6$

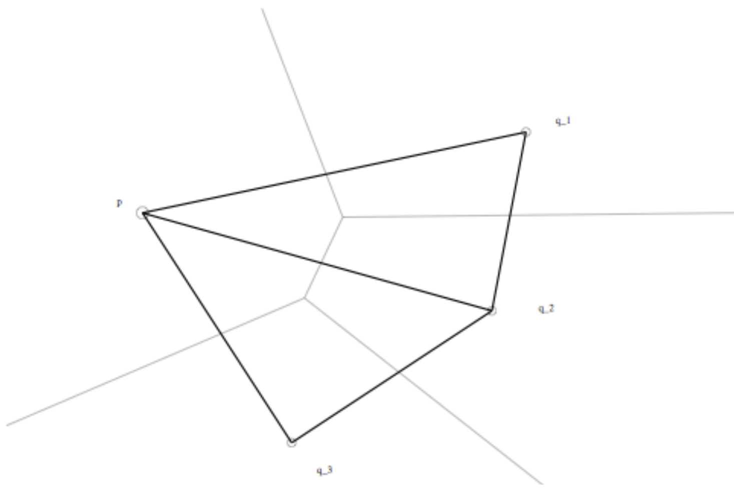
CVD



Il Problema Duale del Diagramma di Voronoi

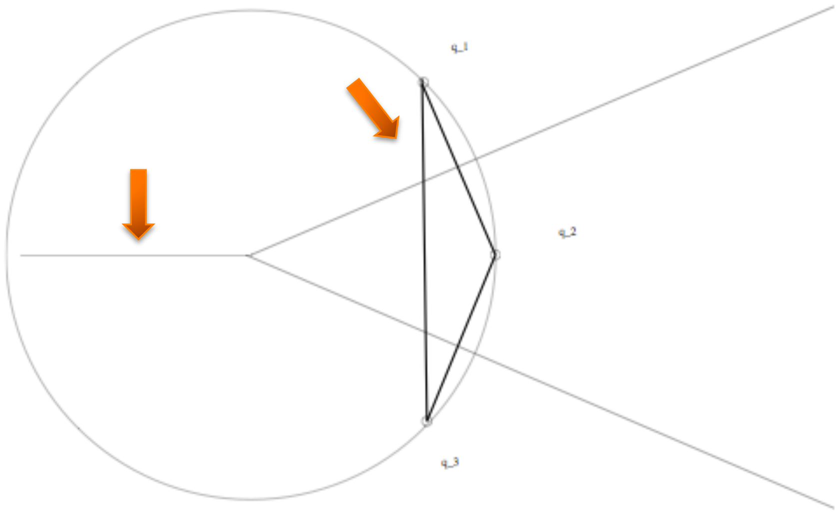
Il Problema duale del Diagr. di Voronoi

- Il problema duale rispetto alla decomposizione del piano in celle di Voronoi è la triangolazione di Delaunay (ottenuta intersecando ogni asse di Voronoi con un segmento che congiunge i punti generatori)



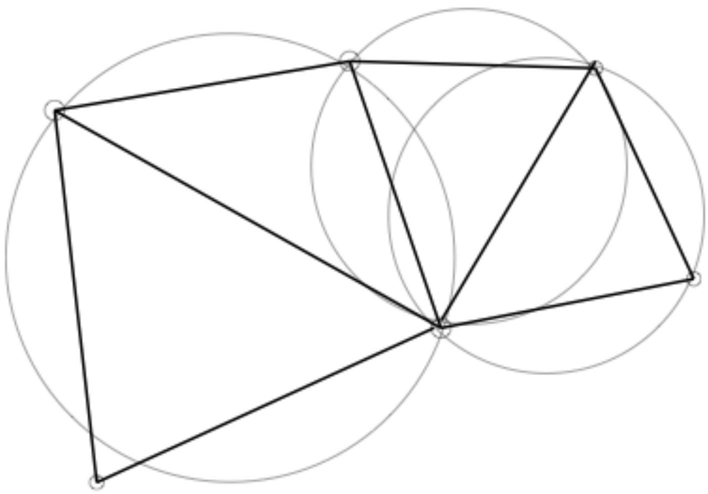
La Triangolazione di Delaunay (1)

- N.B. archi duali non necessariamente si intersecano!



La Triangolazione di Delaunay (2)

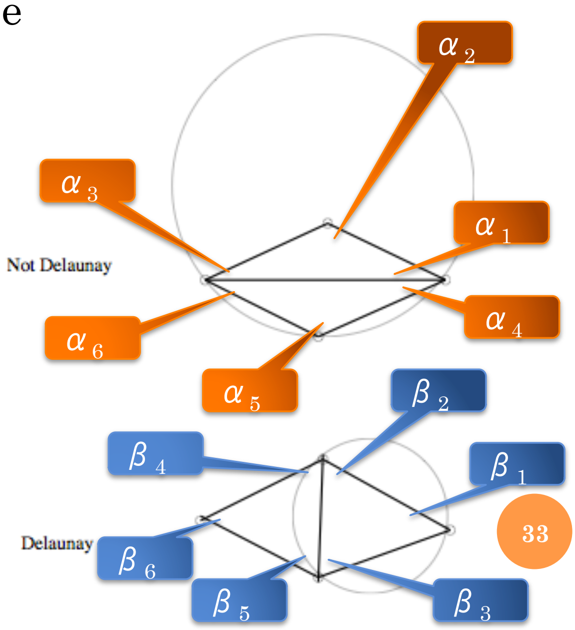
- Proprietà fondamentale: il cerchio circoscritto ad un triangolo non contiene punti dell'insieme



LA TRIANGOLAZIONE DI DELAUNAY (3)

- Proprietà fondamentale: nessun arco può essere illegale
- Un arco è illegale se:

  min α_i<min β_i
- Se *e* è un arco illegale, possibile scambiare triangoli per rendere triangolazione di Delaunay



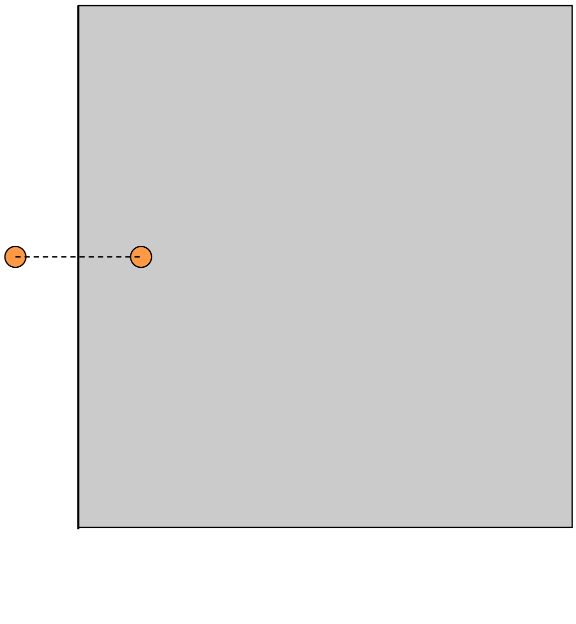
LA TRIANGOLAZIONE DI DELAUNAY (4)

- Ci sono lavori che usano la triangolazione di Delaunay per indirizzare i sensori verso una posizione che consenta la copertura globale
- Esistono diversi algoritmi per calcolare una triangolazione di Delaunay -> TESINA
- E' possibile calcolare il diagramma di Voronoi a partire dalla triangolazione di Delaunay.
- Altrimenti…

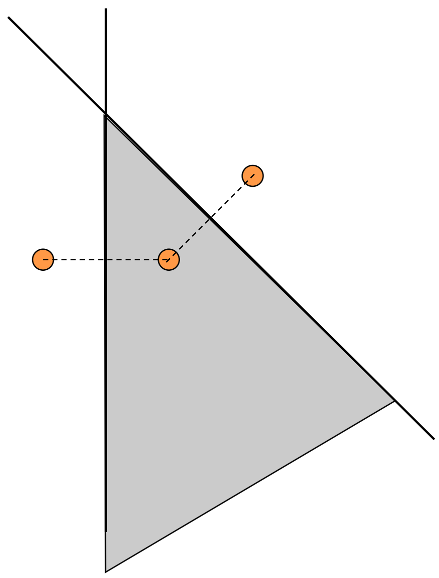
Algoritmi per calcolare il Diagramma di Voronoi

Algoritmo basato sull'Intersezione dei Semipiani (1)

Una cella di Voronoi si ottiene intersecando ripetutamente dei semipiani:

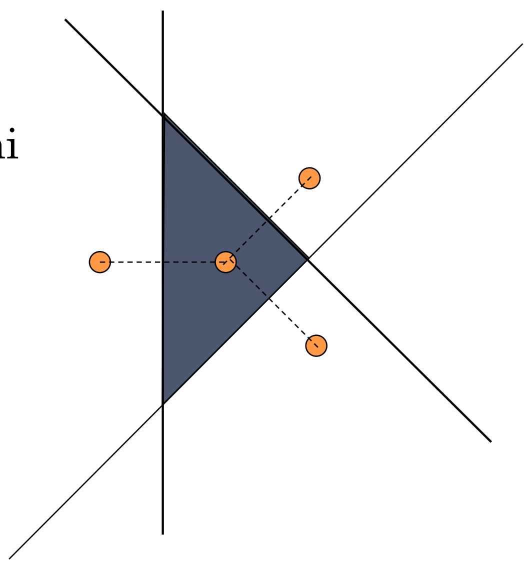


Algoritmo basato sull'Intersezione dei Semipiani (2)



Algoritmo basato sull'Intersezione dei Semipiani (3)

Questa operazione va ripetuta per ogni punto.



ALGORITMO BASATO SULL'INTERSEZIONE DEI SEMIPIANI (4)

Ma quanto costa trovare l'intersezione di un certo numero k di semipiani?
Un algoritmo usa la tecnica del divide-et-impera…

ALGORITMO BASATO SULL'INTERSEZIONE DEI SEMIPIANI (5)

Passo divide:
L'insieme dei k semipiani è ricorsivamente suddiviso fino ad ottenere k semipiani singoli (Nota che questo passo induce una struttura di albero binario).
Passo Impera:
Il semipiano in ogni foglia viene intersecato con un rettangolo R (lo spazio di ricerca). Così in ogni foglia c'è ora un poligono.
Passo Combina:
Calcola ricorsivamente l'intersezione dei due poligoni figli e ponilo nel nodo corrente dell'albero bottom-up.

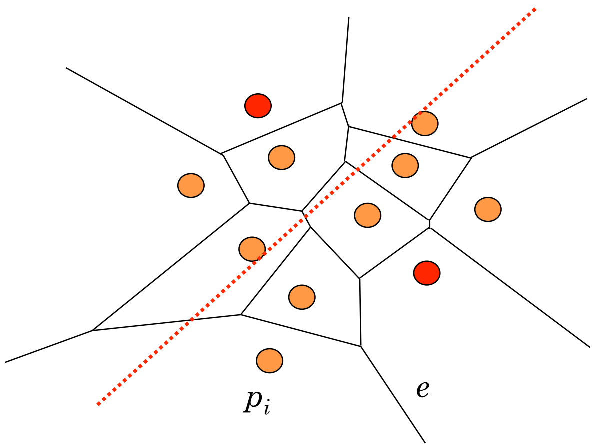
ALGORITMO BASATO SULL'INTERSEZIONE DEI SEMIPIANI (6)

Complessità del Passo Combina:
Siano *p* e *p'* il # di vertici di due generici poligoni che devono essere intersecati a qualche livello dell'albero. Questo passo si può eseguire in tempo $O(p+p')$.

Si può dim. che la complessità dell'intero algoritmo è $O(k \log k)$ e questo è ottimo perché è possibile mostrare che il problema dell'ordinamento per confronti si riduce al problema della ricerca dell'intersezione di semipiani.
Complessità dell'algoritmo per il diagramma di Voronoi: per calcolare una cella devo intersecare $O(n^2)$ semipiani, quindi $O(n^2 \log n^2)$.

INTUIZIONE (1)

Non tutte le coppie formano un asse!



Intuizione (2)

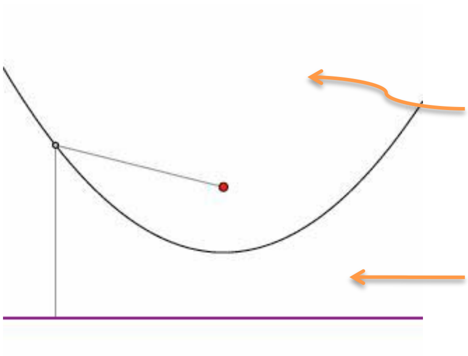
- L'idea è quella di prendere in prestito una tecnica nota della geometria computazionale e sfruttarla.
- Una buona candidata è la sweep line, usata per risolvere problemi geometrici bidimensionali tramite una sequenza di sottoproblemi quasi monodimensionali.
- Esempio classico: [Bentley, Ottmann '79] calcolare le intersezioni di n segmenti facendo spazzare il piano da una linea orizzontale (sweep line).
- Quando essa si muove, incontra degli oggetti e l'algoritmo risolve il problema relativo a questi soli oggetti.

Intuizione (3)

- Questo metodo non funziona così com'è per i diagrammi di Voronoi, poiché sarebbe necessario "predire" la posizione dei siti prima che la sweep line li incontri.
- Fortune [1986] ha ideato un modo per aggirare il problema, basandosi su una linea aggiuntiva, detta beach line.

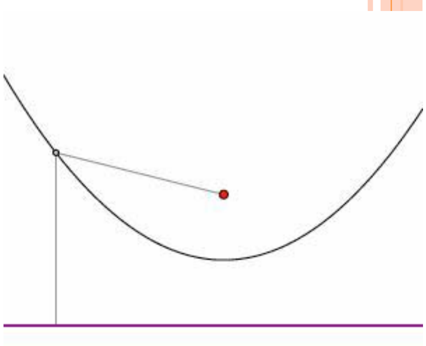
ALGORITMO DI FORTUNE (1)

- Idea: piuttosto che considerare le distanze tra i vari punti, introduciamo una linea (*sweep line*) che spazza il piano e la usiamo per facilitare il confronto tra distanze.
- E' come se questa linea scoprisse il diagr. di Voronoi man mano che spazza il piano.
- N.B. Dato un punto p ed una retta l esterna ad esso, il luogo dei punti equidistanti forma una parabola $P_{p,l}$.



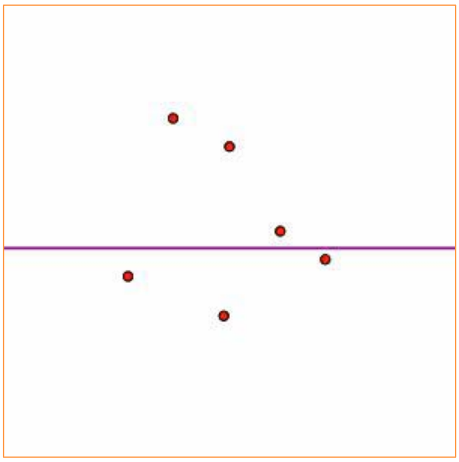
ALGORITMO DI FORTUNE (2)

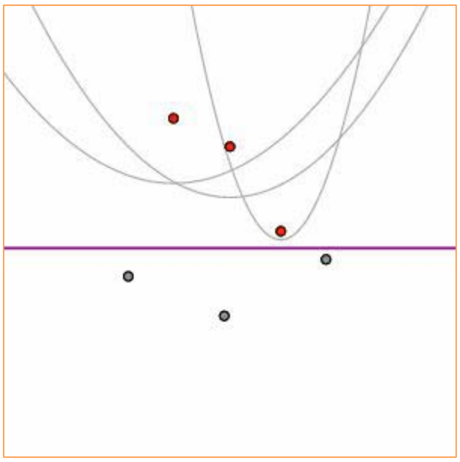
- Si consideri un generico punto $q=(q_x, q_y)$.
- La sweep line sarà orizzontale e la sua coordinata verticale è l_y. Quindi $dist(q,l)=l_y-q_y$.
- La condizione che q sia sulla parabola generata da p ed l è: $dist(q,p)=l_y-q_y$.
- Più in generale:
  - $dist(q,p)< l_y-q_y$ se q giace sopra la parabola
  - $dist(q,p)=l_y-q_y$ se q giace sulla parabola
  - $dist(q,p)>l_y-q_y$ se q giace sotto la parabola



ALGORITMO DI FORTUNE (3)

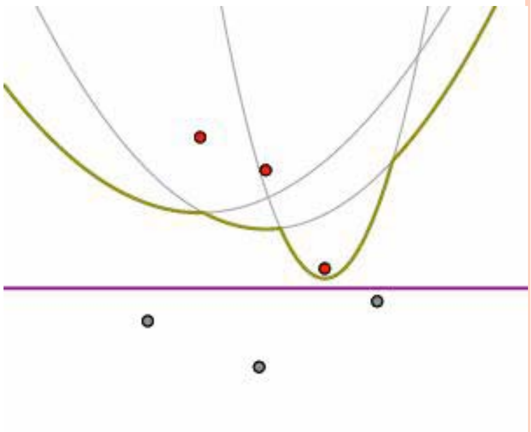
- La sweep line si muove verso il basso.
- In ogni istante, consideriamo i punti al di sopra della sweep line e le parabole che essi definiscono.
- Ad esempio:





ALGORITMO DI FORTUNE (4)

- Definiamo *beach line* la curva formata dai più bassi archi di parabola.
- In altri termini: ogni linea verticale passa per molte parabole; il punto in cui la linea verticale passa per la beach line è il punto più basso.
- N.B. Ogni arco che compone la beach line è associato ad uno dei siti sopra la sweep line.



Per vedere come cambia la beach line: http://merganser.math.gvsu.edu/david/voronoi.08.06/animatedBeachLine.html

Algoritmo di Fortune (5)

- Osserva che se un punto è al di sopra della beach line, sarà più vicino ad uno dei siti sopra la sweep line che alla sweep line stessa.
- Questo significa che questo punto giace nella cella di Voronoi di un punto che la sweep line ha già spazzato.
- Perciò, il diagramma di Voronoi al di sopra della beach line è determinato dai siti al di sopra della sweep line.

Algoritmo di Fortune (6)

- Ora determiniamo la condizione per cui la beach line passi per un punto q.
- Sia q tale che $dist(q,p_1) \leq dist(q, p_i)$ per ogni altro i.
- La condizione che q sia sulla parabola generata da p_1 e da l è: $dist(q, p_1)=l_y-q_y$.
- Unendo le 2: $dist(q, p_i) \geq dist(q,p_1)=l_y-q_y=dist(q,l)$, cioè q non è al di sopra di alcuna parabola, perciò q è sulla beach line. In altre parole:

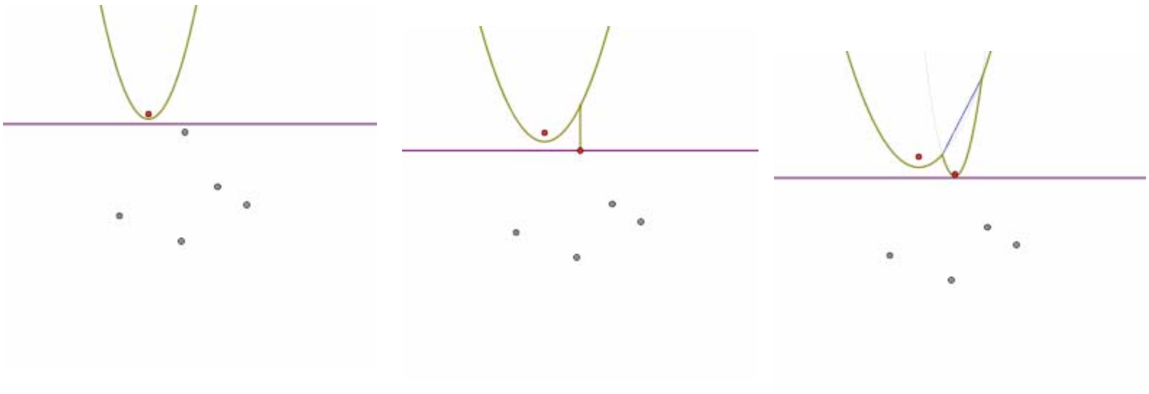
  quando un punto appare sulla beach line, esso è sull'arco di parabola associato al sito a lui più vicino.

Algoritmo di Fortune (7)

- I punti sulla beach line che giacciono all'intersezione di due archi di parabola sono detti *breakpoints*.
- I breakpoints sono contemporaneamente più vicini a due siti. In altre parole,

  i breakpoints giacciono sui segmenti del diagramma di Voronoi.
- Per costruire il diagramma di Voronoi, basta tenere traccia dei breakpoints.

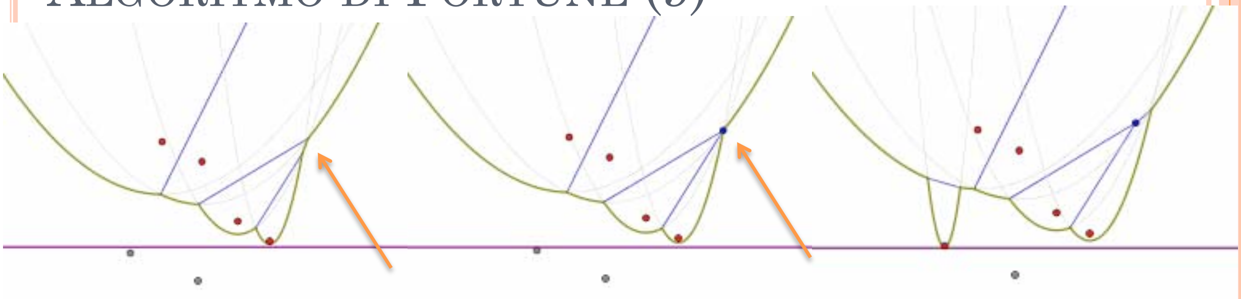
Algoritmo di Fortune (8)



Determinare i segmenti:

- Una coppia di breakpoints che corrisponde ad un segmento nel diagramma di Voronoi appare sulla beach line quando la sweep line attraversa un nuovo sito.
- Chiamiamo questa situazione *site event*.

ALGORITMO DI FORTUNE (9)

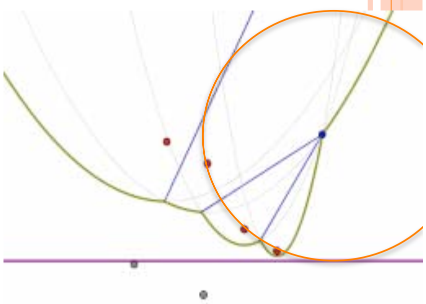


Determinare i vertici:

- Mentre la sweep line si muove, i breakpoints si muovono lungo un segmento fino a raggiungere un vertice, quando uno degli archi di parabola scompare

ALGORITMO DI FORTUNE (10)

- La comparsa di nuovi archi di parabola sulla beach line è facile da determinare: avviene quando la sweep line attraversa un sito.
- Analogamente, la scomparsa di un arco di parabola è facile da determinare: quando questo si riduce ad un punto x, questo punto giace su 3 parabole:
  - quella che contiene l'arco che sta scomparendo
  - l'arco di parabola alla destra di questo
  - l'arco di parabola alla sinistra di questo.



- x è dunque equidistante da 3 siti, corrispondenti ai 3 archi di parabola -> un cerchio centrato in x vi passa.
- Troviamo un vertice quando la sweep line ha finito di spazzare quel cerchio.
- Chiamiamo questa situazione *circle event*.

Algoritmo di Fortune (11)

Algoritmo di Fortune

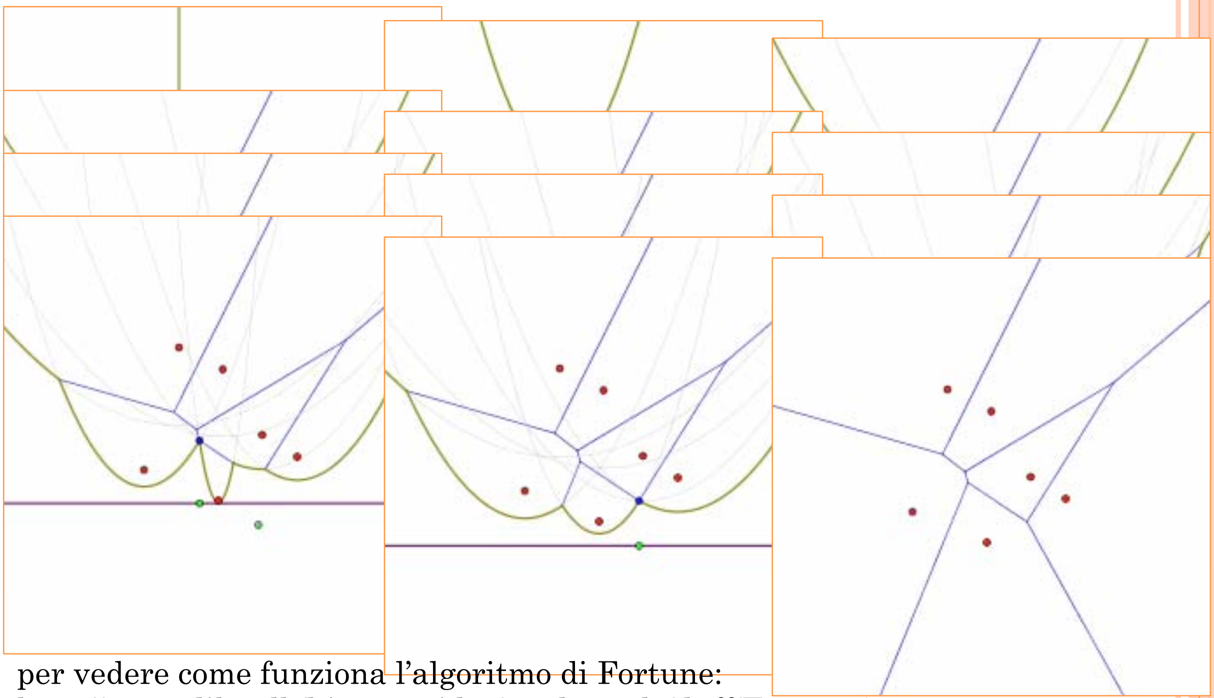
- Per determinare segmenti e vertici del diagramma, dobbiamo determinare la comparsa e scomparsa di archi di parabola sulla beach line.
- Teniamo traccia della beach line immaginando di percorrerla da sx a dx e memorizzando l'ordine dei siti che producono degli archi di parabola.
- Questo ordine non cambia fino a quando non si verifica un site event o un circle event.
- I breakpoints sono implicitamente memorizzati come intersezioni di archi di parabola sulla beach line.

Algoritmo di Fortune (12)

Algoritmo di Fortune (segue)

- Se il prossimo evento incontrato dalla beach line è:
  - un site event, inseriamo il nuovo sito nella lista dei siti nell'ordine in cui compare il suo arco di parabola e memorizziamo il nuovo segmento del diagramma
  - un circle event, memorizziamo il nuovo vertice del diagramma e che esso è un estremo dei segmenti corrispondenti ai due breakpoints che si sono venuti a trovare vicini.
- In entrambi i casi, verifichiamo se si è generata una nuova tripla di siti che produrrà un futuro circle event.
- Il diagramma viene costruito considerando la sequenza (finita) di questi eventi.

ALGORITMO DI FORTUNE (13)



per vedere come funziona l'algoritmo di Fortune:
http://www.diku.dk/hjemmesider/studerende/duff/Fortune/

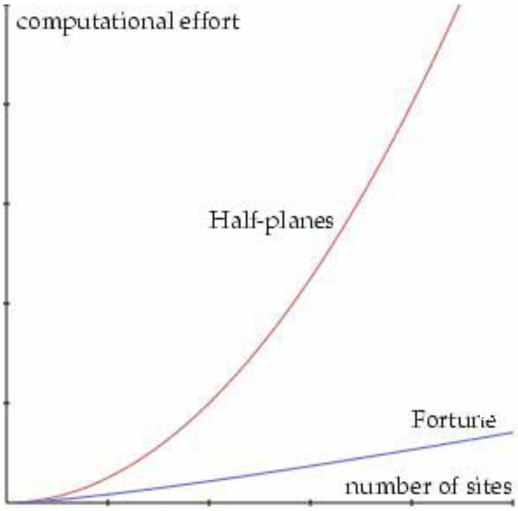
COMPLESSITÀ

- L'analisi della complessità segue l'analisi tipica degli algoritmi basati su sweep line.
- Ogni evento prende tempo *O(1)* più un numero costante di accessi alle strutture dati.
- Ognuno di questi accessi costa *O(logn)* tempo
- Le strutture dati contengono *O(n)* informazioni

- Il tempo totale è *O(n log n)*, e lo spazio è *O(n)*.
- Questa complessità è ottima perché il problema dell'ordinamento per confronti si può ridurre a quello del calcolo del diagramma di Voronoi.

Conclusioni

- L'algoritmo di Fortune è un modo efficiente per calcolare il diagramma di Voronoi.
- Qualunque algoritmo si usi, è ragionevole pensare che la complessità cresca al crescere del # dei siti.
- L'algoritmo basato sull'intersezione dei semipiani impiega tempo $O(n^2 \log n)$ se n sono i siti.
- La complessità dell'algoritmo di Fortune è $O(n \log n)$.



Verso i Sensori Eterogenei

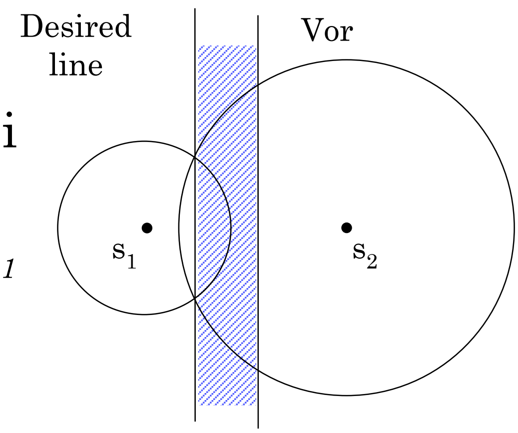
SENSORI ETEROGENEI

- I sensori non devono necessariamente essere tutti uguali. Abbiamo una rete di sensori eterogenei in caso di:
  - dispositivi diversi
  - capacità del dispositivo dipendente dalla posizione (terreno non perfettamente liscio, ostacoli, …)

- Gli approcci descritti (forze virtuali e Voronoi) non funzionano con sensori eterogenei:
  - Forze virtuali: le forze dipendono solo dalla distanza
  - Voronoi: le celle non tengono conto della capacità di copertura

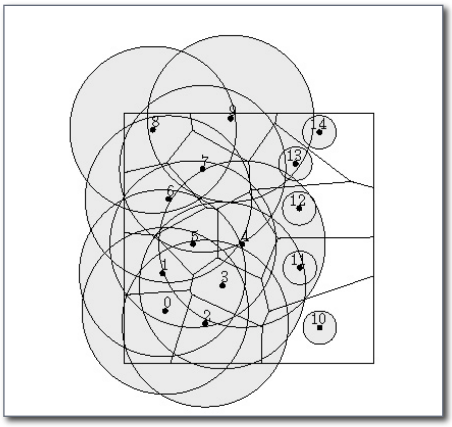
LIMITAZIONI DI VORONOI (1)

- L'algoritmo di Voronoi assegnerebbe:
  - il semipiano di sinistra ad s_1 (anche la zona ombreggiata)
  - il semipiano di destra ad s_2

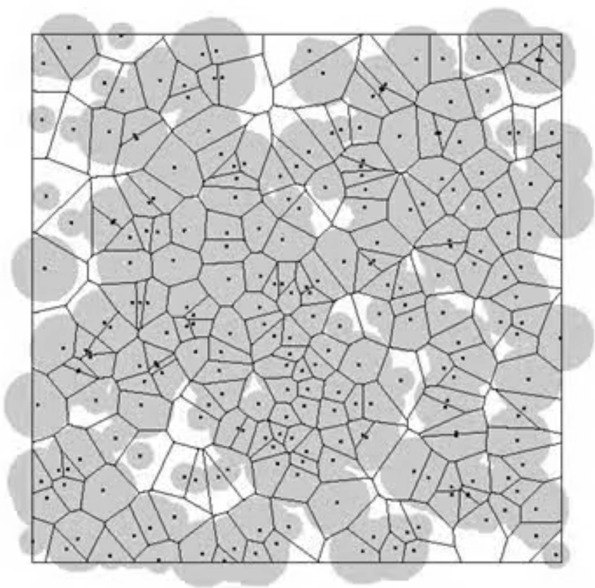


LIMITAZIONI DI VORONOI (2)

- Situazione di stallo:
  - i sensori di sinistra (cerchi grandi) non si muovono perché non vedono buchi di copertura, coprendo il loro poligono interamente
  - i sensori a destra (cerchi piccoli) non si muovono perché stanno già sfruttando al meglio le loro capacità di copertura

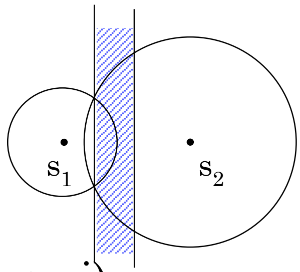


LIMITAZIONI DI VORONOI (3)



UNA NUOVA NOZIONE DI DISTANZA

- la diversità dei dispositivi NON viene considerata
- Introduciamo una nuova nozione di distanza che tiene conto della:
    - distanza euclidea
    - diversità dei dispositivi



- Molte possibilità, ma vogliamo:
    - diagrammi con linee rette (poligoni convessi)
    - una distanza per cui il luogo dei punti equidistanti da due sensori eterogenei passi per l'intersezione dei cerchi di sensing

DISTANZA DI LAGUERRE (1)

W. Blaschke. Vorlesungen uber Differentialgeometrie III. Springer Berlin. 1929.

- Definita in $\mathscr{R}^3$
- Dati due punti $P=(x,y,z)$ e Q=(x',y',z') la loro distanza di Laguerre è:
    - $d_L^2(P,Q)=(x-x')^2+(y-y')^2-(z-z')^2$
- P cerchio (orientato) di centro (x,y) e raggio $|z|$

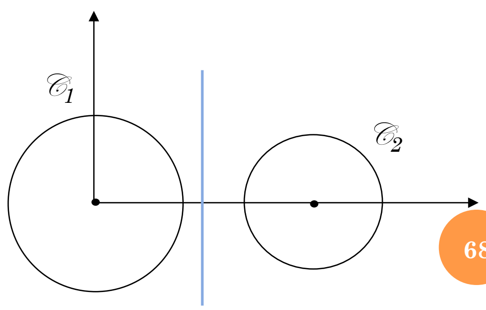
DISTANZA DI LAGUERRE (2)

- Dati due cerchi $\mathscr{C}_1$ e $\mathscr{C}_2$, di centri C_1 e C_2 e raggi r_1 ed r_2, la distanza di Laguerre tra i due cerchi è:
  - $d_L^2(\mathscr{C}_1, \mathscr{C}_2) = d_E^2(C_1, C_2) - (r_1 - r_2)^2$
- La distanza di Laguerre tra un punto $P=(x,y)$ e un cerchio $\mathscr{C}=(x',y',r)$ è:
  - $d_L^2(P,C)=(x-x')^2+(y-y')^2-r^2$

DISTANZA DI LAGUERRE (3)

- Lemma. Dati due cerchi $\mathscr{C}_1$ e $\mathscr{C}_2$ con centri distinti C_1 e C_2 e raggi r_1 ed r_2, il luogo dei punti equidistanti da $\mathscr{C}_1$ e $\mathscr{C}_2$ è una retta (detta asse radicale) perpendicolare al segmento che connette C_1 e C_2 posto a distanza k da C_1, con

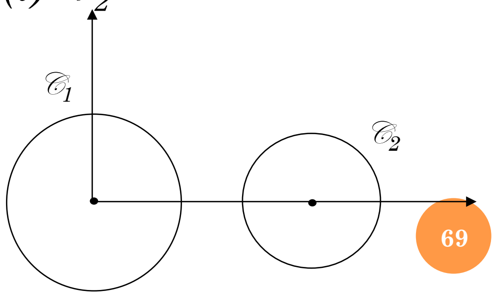
$$k = \frac{d_E(C_1,C_2)}{2} + \frac{r_1^2 - r_2^2}{2d_E(C_1,C_2)}$$



DISTANZA DI LAGUERRE (4)

Dim. Luogo dei punti $P(t)=(x(t), y(t))$ equidistanti, cioè tale che $d_L(P(t), \mathscr{C}_1)=d_L(P(t), \mathscr{C}_2)$.

- Se $C_1=C_2$ ed $r_1=r_2 \Rightarrow$ intero piano
- Se $C_1=C_2$ ed $r_1\neq r_2 \Rightarrow$ insieme vuoto
- Se $C_1\neq C_2$:
  $x(t)^2+y(t)^2-r_1^2=(d_E(C_1,C_2)-x(t))^2+y(t)^2-r_2^2$



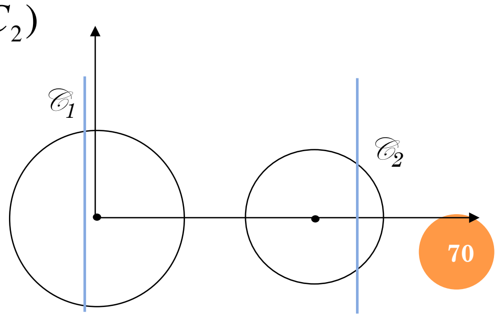
$d_L^2(P,C)=(x-x')^2+(y-y')^2-r^2$

DISTANZA DI LAGUERRE (5)

- Lemma. Dati due cerchi $\mathscr{C}_1$ e $\mathscr{C}_2$ con centri C_1 e C_2 distinti e raggi r_1 ed r_2, i loro centri giacciono dalla stessa parte dell'asse radicale se e solo se
  $$d_E^2(C_1,C_2) < |r_1^2-r_2^2|.$$

Dim. L'asse può giacere a destra o a sinistra.

- Destra:
  $$k=\frac{d_E(C_1,C_2)}{2}+\frac{r_1^2-r_2^2}{2d_E(C_1,C_2)}\geq d_E(C_1,C_2)$$
  $$d_E^2(C_1,C_2) \leq r_1^2-r_2^2 \Rightarrow r_1\geq r_2$$
- Sinistra:
  $$\frac{d_E(C_1,C_2)}{2}+\frac{r_1^2-r_2^2}{d_E(C_1,C_2)}\leq 0$$
  $$d_E^2(C_1,C_2) \leq r_2^2-r_1^2 \Rightarrow r_2\geq r_1$$



DISTANZA DI LAGUERRE (6)

- Collocazioni possibili dell'asse radicale di due cerchi $\mathscr{C}_1$ e $\mathscr{C}_2$

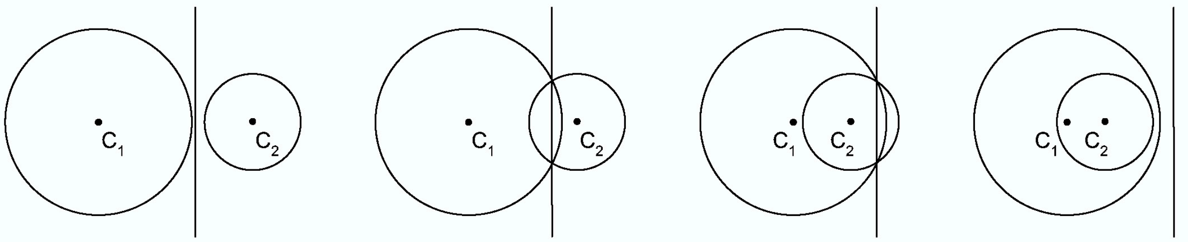
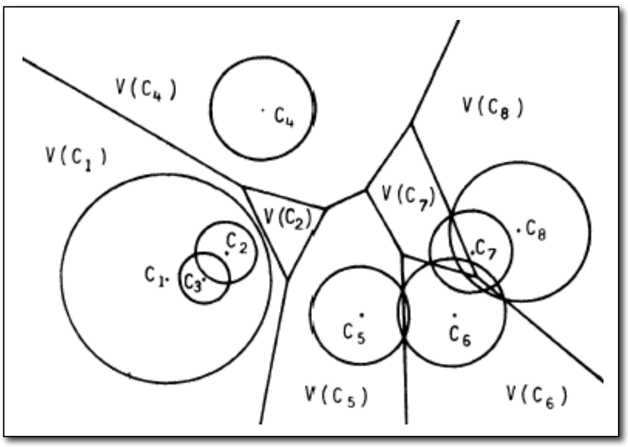


DIAGRAMMA DI VORONOI-LAGUERRE (1)

Diagramma di Voronoi-Laguerre di $\mathscr{C}_1, \ldots, \mathscr{C}_n$:

- $V_i = \cap \{p \in \mathscr{R}^2 \mid d_L{}^2(\mathscr{C}_i, P) \leq d_L{}^2(\mathscr{C}_j, P)\}$

H. Imai, M. Iri, K. Murota. "Voronoi Diagram in the Laguerre Geometry and its Applications". SIAM J. Comput. 14(1), 93-105. 1985.



somiglianze e differenze con i diagrammi di Voronoi classici…

DIAGRAMMA DI VORONOI-LAGUERRE (2)

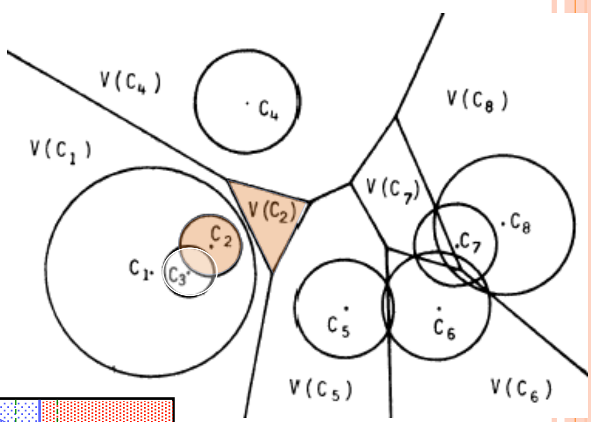
Somiglianze:

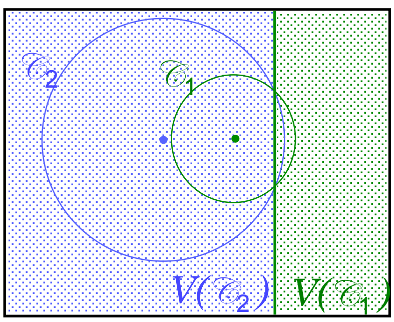
- I poligoni di Voronoi-Laguerre partizionano il piano
- V_i è sempre convesso perché intersezione di semipiani
- se $r_i=0$ per ogni $i=1, \ldots, n$, il diagramma di Voronoi-Laguerre si riduce al diagramma di Voronoi classico.

DIAGRAMMA DI VORONOI-LAGUERRE (3)

Differenze:

- $\mathscr{C}_i$ può essere esterno a V_i (vedi $\mathscr{C}_2$)
- V_i può essere vuoto (ad es. se $\mathscr{C}_i$ è nell'unione di altri cerchi - vedi $\mathscr{C}_3$)





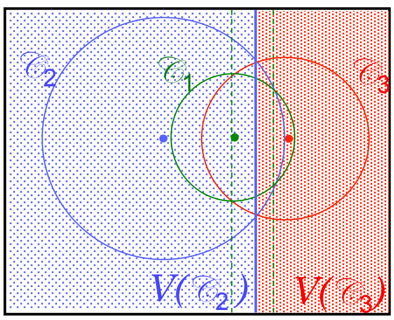


Diagramma di Voronoi-Laguerre (4)

- Teorema. Dati n cerchi $\mathscr{C}_i$ con centri $C_i=(x_i,y_i)$ e raggi r_i, $i=1, \ldots, n$, siano V_i i loro poligoni di Voronoi-Laguerre.

  Per ogni i e j, $V_i \cap \mathscr{C}_j \subseteq \mathscr{C}_i$.

  Cioè, l'intersezione di V_i con un cerchio $\mathscr{C}_j$ è contenuta in $\mathscr{C}_i$.

Diagramma di Voronoi-Laguerre (5)

Dim. Per assurdo, esista un punto $P \subseteq V_i$ in $\mathscr{C}_j$ ma non in $\mathscr{C}_i$, per qualche $j \neq i$.

- Poiché $P \subseteq V_i$ $d_L(P, \mathscr{C}_i) < d_L(P, \mathscr{C}_j)$ per ogni $j \neq i$, cioè

$$d_E^{\,2}(P, \mathscr{C}_i) - r_i^{\,2} < d_E^{\,2}(c_j, P) - r_j^{\,2}$$

- Poiché P è in $\mathscr{C}_j$ ma non in $\mathscr{C}_i$,

$$d_E^{\,2}(P, \mathscr{C}_j) \leq r_j^{\,2} \text{ e } d_E^{\,2}(P, \mathscr{C}_i) \geq r_i^{\,2}$$

- Sostituendo: $0<0$ Assurdo.

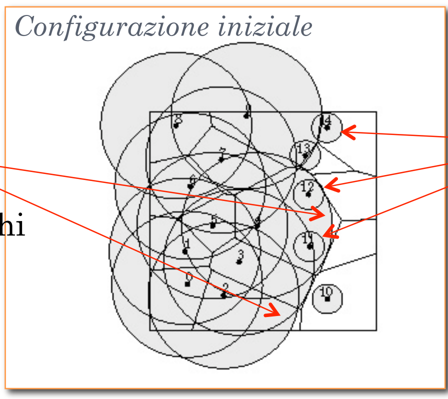
ALGORITMO BASATO SU VORONOI-LAGUERRE (1)

Algoritmo eseguito da ogni sensore s_i:

- calcola V_i
- se s_i è interno a V_i si sposta verso il minimax (o al massimo di $d_i^{max}=r_{tx}/2\text{-}r_i$ dove $r_{tx}=min_i\, r_i^{tx}$) se accresce la copertura di V_i
- se s_i è esterno a V_i si sposta verso il minimax (o al massimo di $d_i^{max}=r_{tx}/2\text{-}r_i$)
- se V_i è nullo, non fa niente.

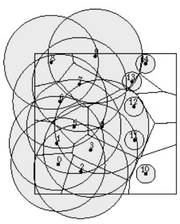
ALGORITMO BASATO SU VORONOI-LAGUERRE (2)

Cerchi grandi
Alcuni si spostano
per coprire meglio i buchi
nel loro poligono

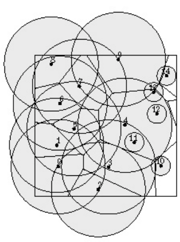


Cerchi piccoli
Molti si spostano
perché sono esterni
al loro poligono

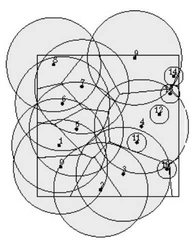
Configurazione iniziale



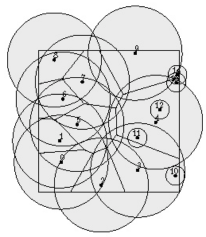
Round 6



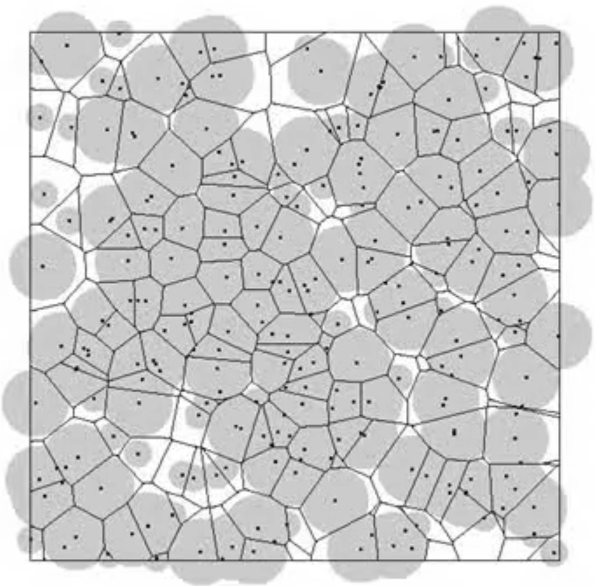
Round 9



Round 12:
Stallo risolto!

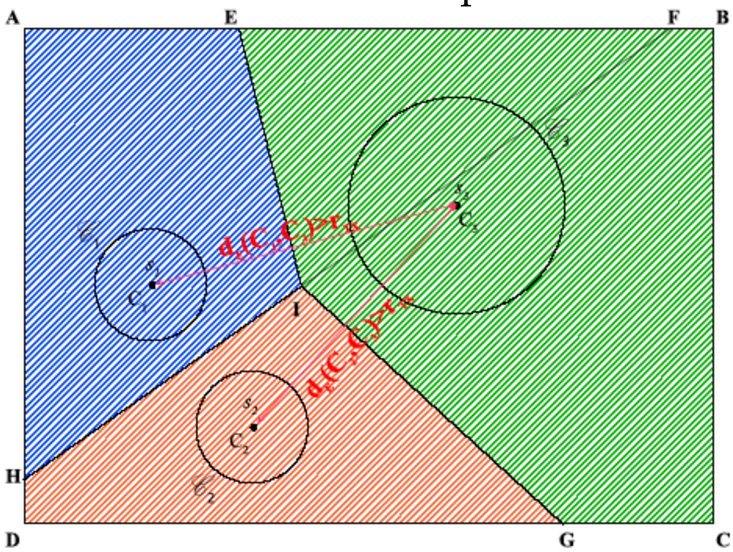


ALGORITMO BASATO SU VORONOI-LAGUERRE (3)



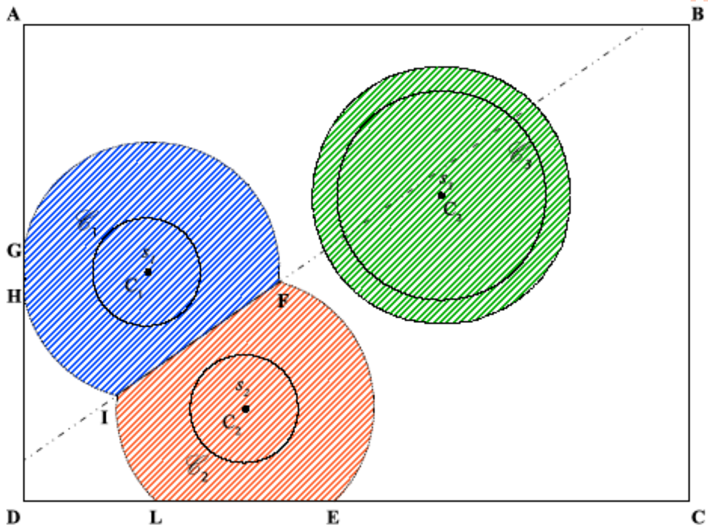
PROPRIETÀ DELL'ALGORITMO (1)

- Osserva:
  - Poligono "locale"≠poligono "globale" e l'insieme dei poligoni locali non forma una partizione!



PROPRIETÀ DELL'ALGORITMO (2)

- Definiamo il poligono curvo V'_i generato dall'intersezione del poligono "locale" con il cerchio di raggio $d_i^{max}+r_i=r_{tx}/2$.



PROPRIETÀ DELL'ALGORITMO (3)

- Lemma. $V'_i \cap V'_j = \phi \ \forall \, i \neq j$
- Lemma. $\forall \, i \neq j,\ V'_i \cap \mathscr{C}_j \subseteq \mathscr{C}_i$.

  In altre parole, ogni poligono curvo può essere coperto dal sensore che lo genera meglio che da qualunque altro sensore.

PROPRIETÀ DELL'ALGORITMO (4)

Th. L'algoritmo converge.

Dim. Sia $V'_i{}^{(k)}$ il poligono curvo di s_i al round k.

- $A_i^{(k)}$ ed $A_i^{(k)}(s_i)$ è l'area coperta in $V'_i{}^{(k)}$ da tutti i sensori e dal solo s_i al round k. $A_i'^{(k)}$ è l'area coperta considerando le posizioni dei sensori al round $k+1$.
- N.B. $A_i'^{(k)} \neq A_i^{(k+1)}$
- $A^{(k)}_{total}$ è l'area coperta dell'AoI da tutti i sensori.
- Dobbiamo dim. che $A^{(k)}_{total} < A^{(k+1)}_{total}$

PROPRIETÀ DELL'ALGORITMO (5)

Dim. (segue)

- $\mathscr{P}^{(k)}=\{V'_1{}^{(k)}, V'_2{}^{(k)}, \ldots, AoI \setminus \cup_i V'_i{}^{(k)}\}$ è una partizione dell'AoI.
- $AoI \setminus \cup_i V'_i{}^{(k)}$ è costituito da punti scoperti che non possono essere coperti in un round e non contribuisce ad $A^{(k)}_{total}$.
- $A^{(k)}_{total} = \Sigma_i A_i^{(k)}$
- $A_i^{(k)} = A_i^{(k)}(s_i)$ (lemma precedente)
- $A_i^{(k)}(s_i) < A_i'^{(k)}(s_i)$ (per l'algoritmo)
- $A_i'^{(k)}(s_i) \leq A_i'^{(k)}$
- Quindi: $A^{(k)}_{total} = \Sigma_i A_i^{(k)} < \Sigma_i A_i'^{(k)}$
- Poiché la copertura al round $k+1$ non dipende dalla partizione:
  $\Sigma_i A_i'^{(k)} = A^{(k+1)}_{total}$

Proprietà dell'Algoritmo (6)

- La convergenza non implica la terminazione.
- Per garantire la terminazione, introduciamo una soglia di movimento minimo ε, in modo che i sensori non si muovono se devono farlo per meno di ε.
- Corollario. L'algoritmo con l'aggiunta della soglia di movimento minimo termina.

Problemi Aperti

- Ostacoli e asperità del terreno
  - anisotropia
  - impedimento al movimento
- Aree di forma complessa
  - regioni concave e corridoi
- …