Géométrie Algorithmique

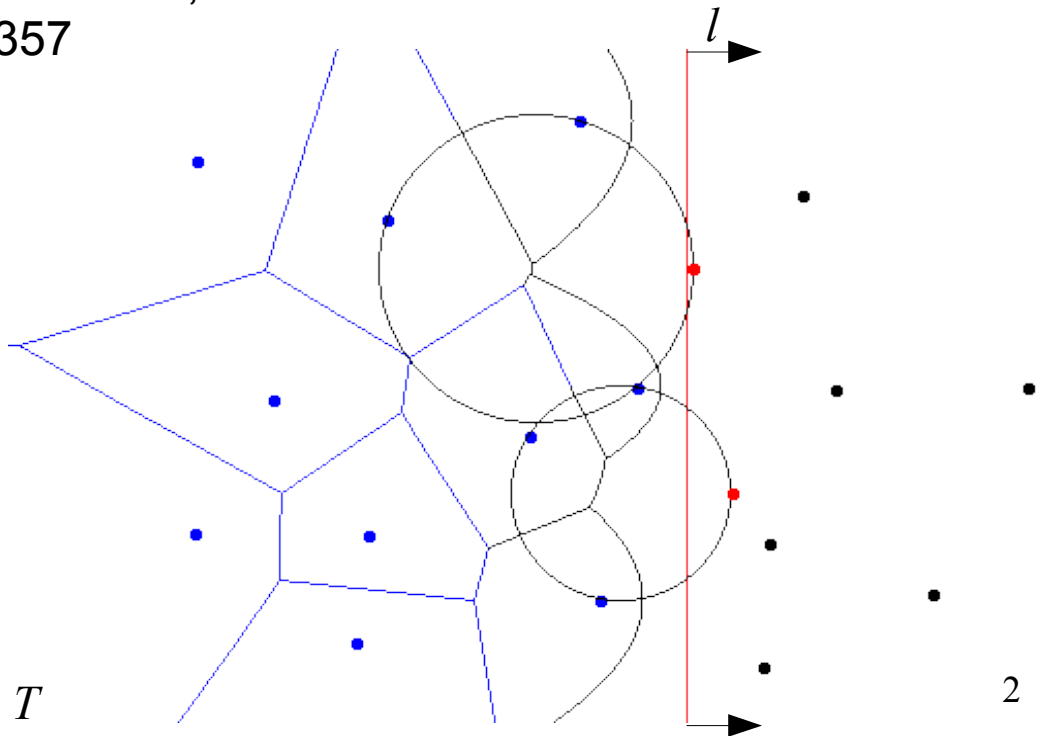
Plan du cours

- Introduction
- Arrangements dans le plan
- Triangulation de polygones
- Diagrammes de Voronoï
- Triangulation de Delaunay
- Recherche/localisation
- Arbres de partition binaire

- Algorithme de balayage plan pour le calcul d'un diagramme de Voronoï- (DDV)
  - Aussi appelé algorithme de Fortune

    Steven Fortune, A sweepline algorithm for Voronoi diagrams, Algorithmica 2 (1-4),pp. 153-174, 1987
    DOI: 10.1007/BF01840357

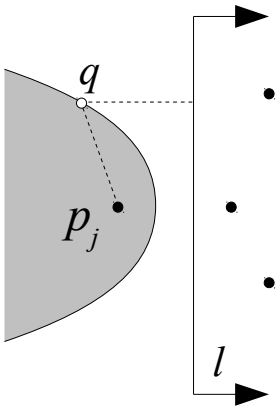
    Algorithme en $O(n \log n)$



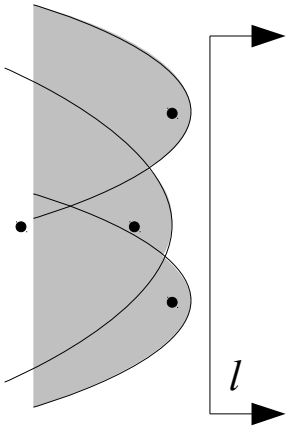
Processus rôdé :
- une ligne imaginaire l balaye le plan
- un statut T est mis à jour à chaque événement
- une file d'événements F classe les événements dans l'ordre d'apparition
- le DDV est construit en relation avec T

- Le paradigme du balayage plan implique la maintien d'un invariant, ici une portion du DDV que l'on sait ne plus changer pour des événements subséquents.
  - Ce n'est pas la portion du DDV située immédiatement à gauche de l.

    En effet une partie de $Vor(P)$ située à gauche de l dépend de points p_i situés plus à droite.



  - Plus précisément, la partie ne changeant plus est l'ensemble des points q tels que la distance à la ligne l est supérieure ou égale à la distance aux point p_j situés à gauche de l.
  - La limite du domaine prend la forme d'une suite d'arcs de parabole – ou « ligne de rivage »
  - La ligne de rivage est monotone en y.
  - Le statut T choisi est cette ligne de rivage. Comment évolue-t-il ?

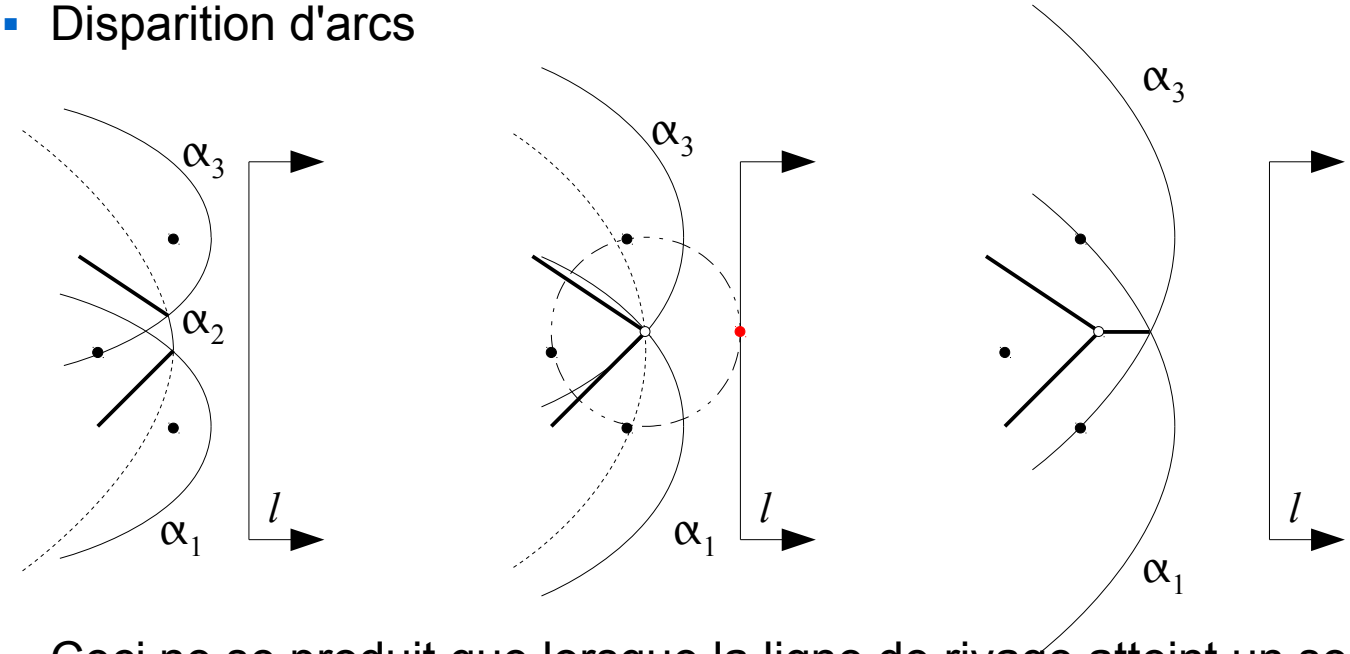


- Évolution de la ligne de rivage :
  - Apparition de nouveaux arcs

  - Seulement lorsque la ligne l atteint un point p_i. Ceci constitue une première catégorie d'événements (les points de P)

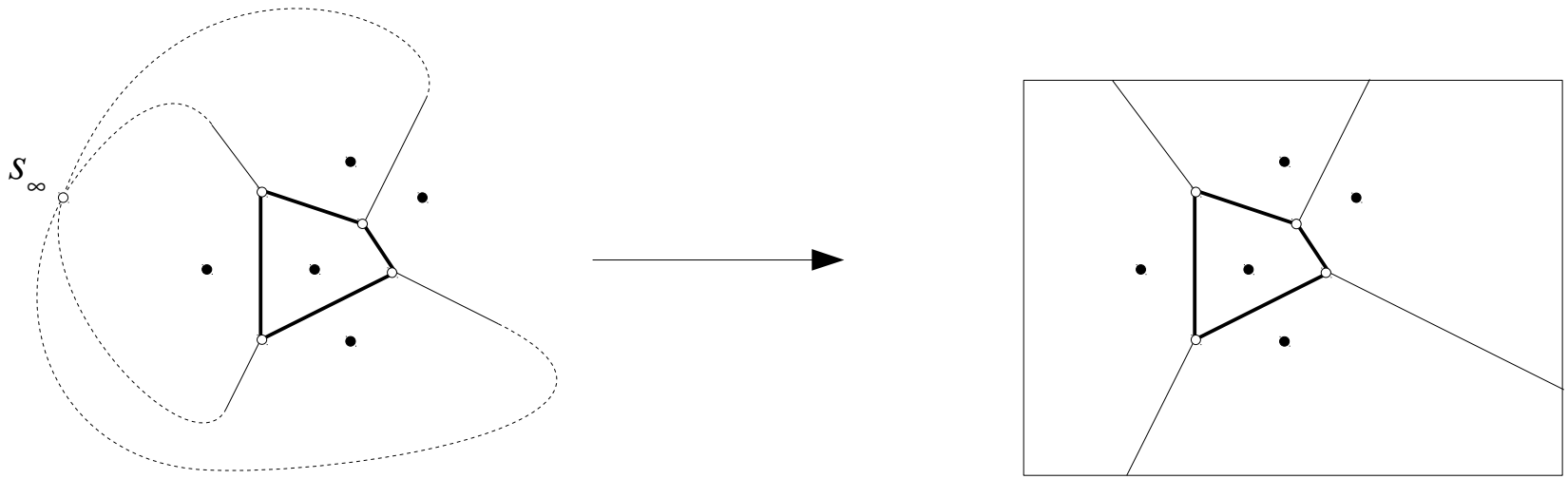
    Preuve : cf Livre , p 153

- Évolution de la ligne de rivage :
  - Disparition d'arcs



  - Ceci ne se produit que lorsque la ligne de rivage atteint un sommet du DDV, autrement dit, quand l atteint le point le plus à droite d'un cercle passant par trois points p_i correspondant à trois arcs consécutifs α_i de la ligne de rivage.
  - Ceci constitue une seconde catégorie d'événements (cercles limites)

- Structures de données
  - DDV : un DCEL . Toutefois, on ne peut représenter les demi-droites et les droites. Il faut donc une boite englobante pour limiter celles ci et pouvoir utiliser un DCEL classique



- Structures de données
  - Statut T : La ligne de rivage – y-monotone – est un arbre binaire balancé. **Les feuilles** stockent les arcs de celle ci (arcs paraboliques) et sont classés par coordonnée y croissante. A chaque feuille correspond un pointeur vers un événement de la file d'événements – l'événement cercle qui correspond à la disparition de l'arc. Si celui ci n'existe pas (ou pas encore), mettre NULL.

    Les **nœuds internes** de l'arbre représentent les « cassures » dans la ligne de rivage. Celles ci sont identifiés par un tuple ordonné (p_i, p_j) , où p_i désigne le sommet correspondant à la parabole inférieure, et p_j celui de la parabole supérieure. A chaque nœud interne, on peut stocker un pointeur vers une demi-arête du DDV

    On peut faire une recherche en $O(\log n)$ de l'arc immédiatement à la gauche d'un sommet p_k quelconque. Lorsque l'on traverse l'arbre, à chaque nœud interne il suffit de comparer la coordonnée y du point p_k avec la coordonnée y de la cassure, calculée « à la volée » connaissant la position de l (qui passe alors par p_k).

- Structures de données
  - File d'événements F : Une file de priorité pour laquelle les événements sont classés lexicographiquement par x croissants, puis par y croissants.

    Dans cette file, figurent tous les points de $p_i \in P$, ainsi que les événements « cercles » . Le point utilisé pour le classement de ces derniers est le point le plus à droite (x_{max}), de plus un pointeur vers l'arc correspondant dans le statut T y est ajouté.
  - Tous les points p_i sont connus d'avance, par contre les événements cercles ne le sont pas.
- Détection des événements cercles ?

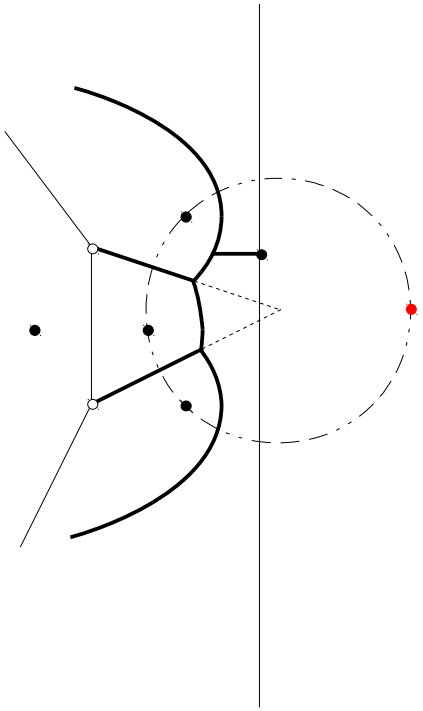
- Détection des événements cercles
  - A chaque événement, des triplets contigus d'arcs apparaissent ou disparaissent
    - Ceux ci correspondent potentiellement à un événement cercle
  - Il faut s'arranger pour que à chaque nouveau triplet, un événement cercle soit inséré dans la file F.

    Deux remarques :
    - Il est possible que pour un triplet donné,les deux arêtes concernées du DDV divergent. Pour ce triplet, pas besoin d'insérer un événement cercle
    - Si les arêtes convergent, il est aussi possible que le triplet disparaisse (à cause d'un événement) avant le traitement de l'événement cercle. Dans ce cas, il s'agit d'une « fausse alerte » et l'événement doit être retiré de L.

Deux remarques :

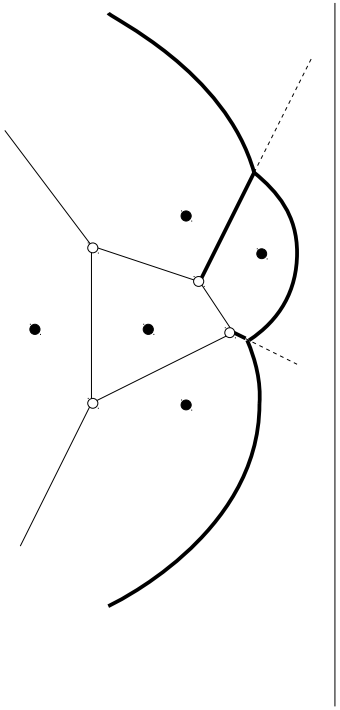
- Si les arêtes convergent, il est aussi possible que le triplet disparaisse (à cause d'un événement) avant le traitement de l'événement cercle. Dans ce cas, il s'agit d'une « fausse alerte » et l'événement doit être retiré de L.

  Ceci est facile si le lien entre les arcs dans le statut et les événements cercle potentiels existe.



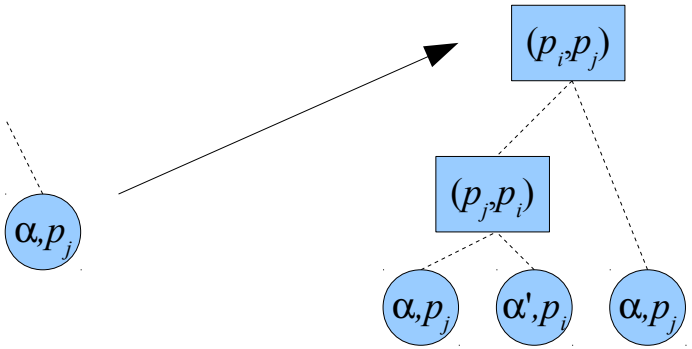
Deux remarques :

- Il est possible que pour un triplet donné, les deux arêtes concernées du DDV divergent. Pour ce triplet, pas besoin d'insérer un événement



- Algorithme global

Voronoi(P)
En entrée : un ensemble de points $P=\{p_0,p_1 \ldots , p_{n-1}\}$
En sortie : un DCEL du DDV de P dans sa boite englobante (V)
{
Initialiser la file d'événements *F* avec les éléments de *P* (événements « points »)
Initialiser un statut *T* vide
Initialiser un DCEL *V* vide
Tant que F est non vide
{
Enlever le premier événement de F (le premier lexicographiquement)
Si c'est un événement « point » alors TraiteEvtPoint(pi)
Sinon TraiteEvtCercle(f) // f est la feuille du statut correspondant à l'arc qui va disparaître
}
Les noeuds internes dans T correspondent aux demi-droites du DDV. Il convient de les limiter par la boite englobante (calculer celle ci de telle façon qu'elle contienne tous les sommets du DDV) et de les insérer dans le DCEL.
Parcourir le DCEL et créer les faces (cellules)
}



TraiteEvtPoint(p_i)

{

Si T est vide, insérer p_i dans T

Sinon

{

Rechercher l'arc α immédiatement situé à gauche de p_i ; celui ci est associé à une autre point p_j

Si la feuille contenant l'arc α a un pointeur vers un événement cercle dans F, éliminer cet événement.

Remplacer la feuille dans T par par un sous arbre avec 3 feuilles correspondant aux 3 nouveaux triplets

Dans ce sous-arbre :

La feuille du milieu correspond à un nouvel arc associé à p_i ;

Les deux autres sont une copie de l'ancien arc α

Insérer deux nouveaux nœuds internes (p_i,p_j) et (p_j,p_i)

Re-balancer l'arbre.

Créer deux nouvelles demi-arêtes dans le DCEL V, pour l'arête séparant $V(p_i)$ et $V(p_j)$

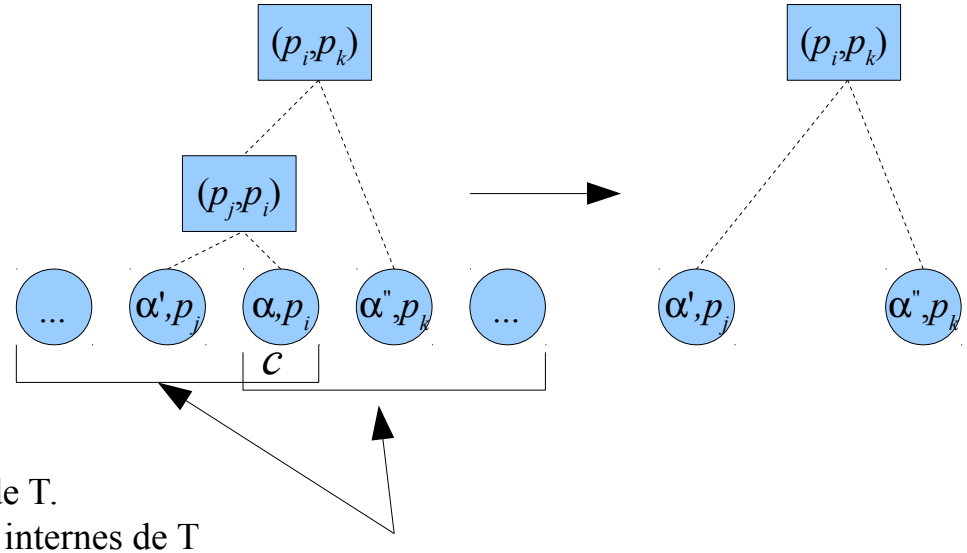
Vérifier que les nouveau triplets d'arcs consécutifs avec l'arc associé à p_i situé en dessous ou dessus (pas au milieu) correspondent à deux arêtes convergentes.

Si c'est le cas, ajouter un événement cercle dans la file F.

}

}

Diagrammes de Voronoï



TraiteEvtCercle(*c*)

{

Effacer la feuille *c* (contenant l'arc α) de T.

Mettre à jour les tuples dans les nœuds internes de T

Effacer tous les événements cercles associés à α (ils peuvent être trouvés en parcourant T à partir de *c*)

Ajouter le centre du cercle comme sommet dans le DCEL du DDV

Créer deux demi arêtes correspondant aux deux nouveaux triplets

Vérifier pour chaque nouveau triplet consécutif qu'il correspondent à deux arêtes convergentes

si c'est le cas,

insérer le événement cercle correspondant dans F.

}

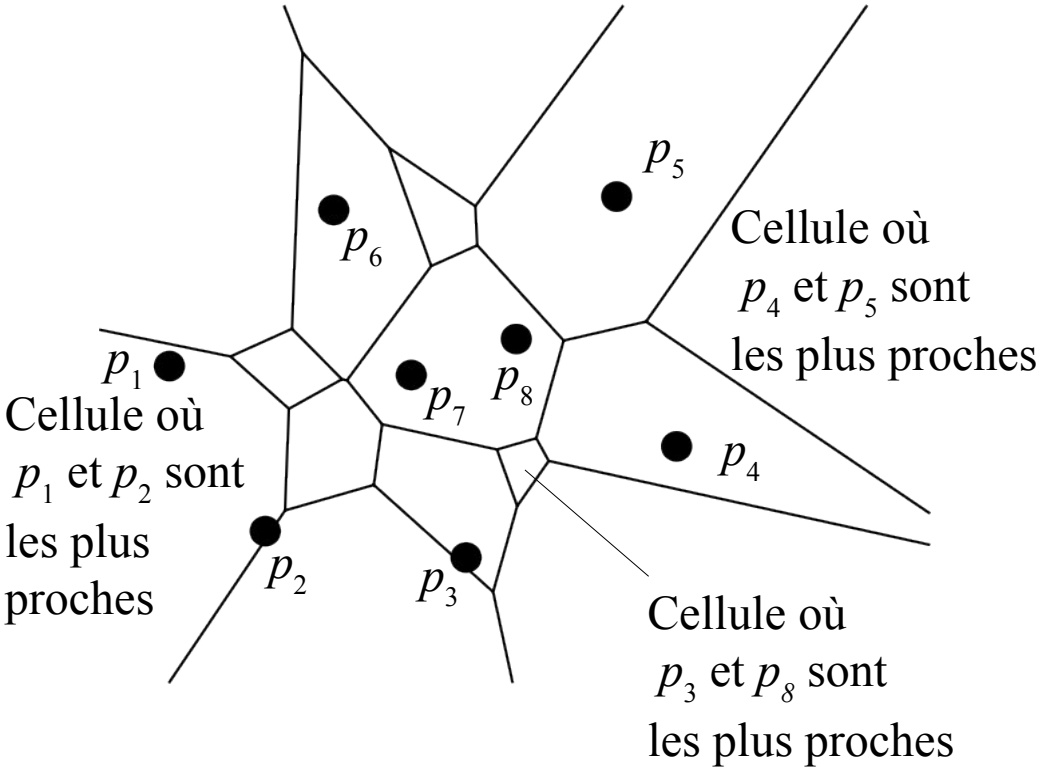
- Cas dégénérés :
  - Au début de l'algorithme : points de même coodonnée x
  - Points cocycliques (4 ou plus)
  - Événement point exactement à la droite d'une cassure de la ligne de rivage
  - Points correspondants à trois arcs successifs colinéaires

Autres types de diagrammes de Voronoï

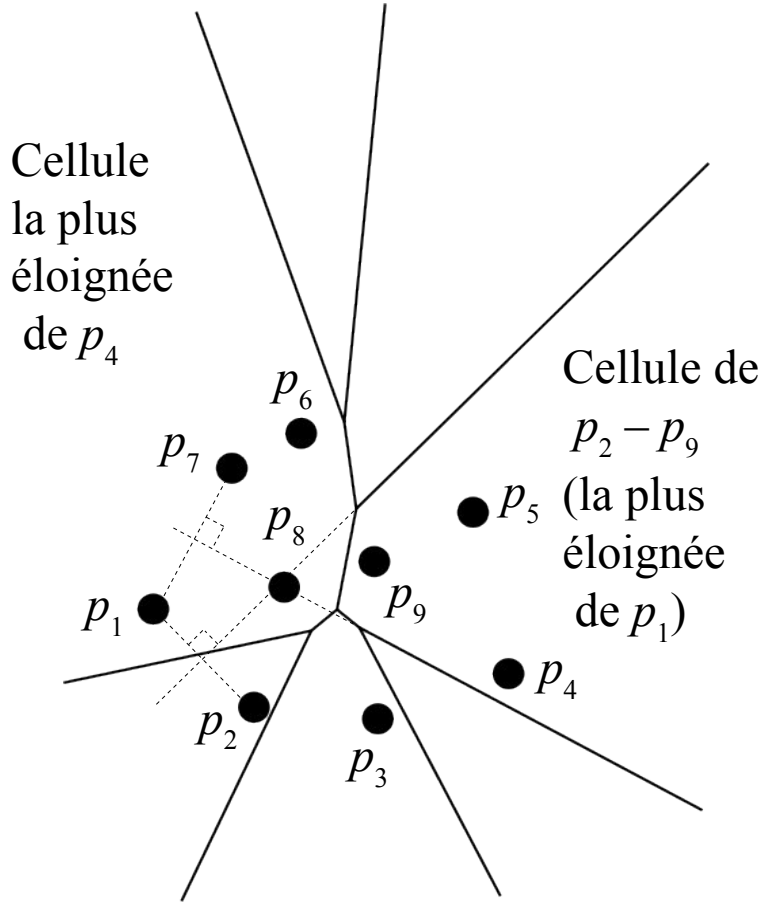
- Diagrammes d'ordre p
  - Construction des régions des p plus proches sites
  - Diagrammes d'ordre 1 = diagramme de Voronoï classique
  - Avec n sites, il est possible de construire un diagramme d'ordre maximum $n-1$
  - Le diagramme d'ordre $n-1$ est le diagramme de *distance maximale* (DDM)

Diagrammes de Voronoï

- Diagramme d'ordre 2



- Diagramme d'ordre $n-1$

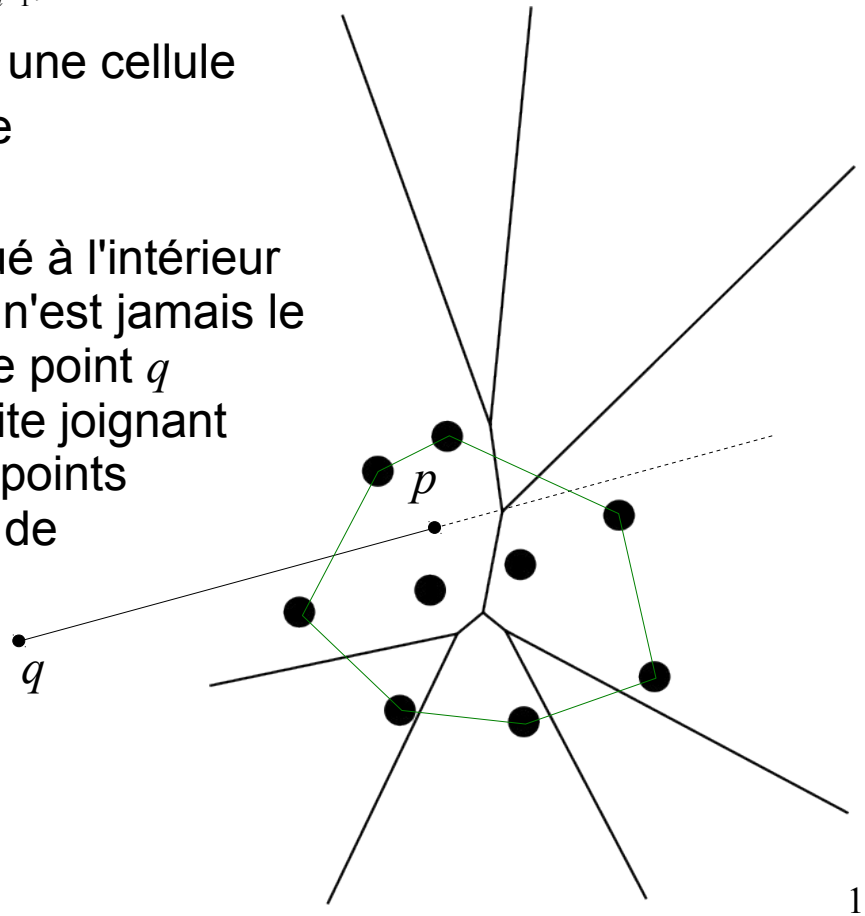


- Propriétés du DDM

  Soit un ensemble $P = \{p_1 \ldots p_{n-1}\}$ de points dans le plan

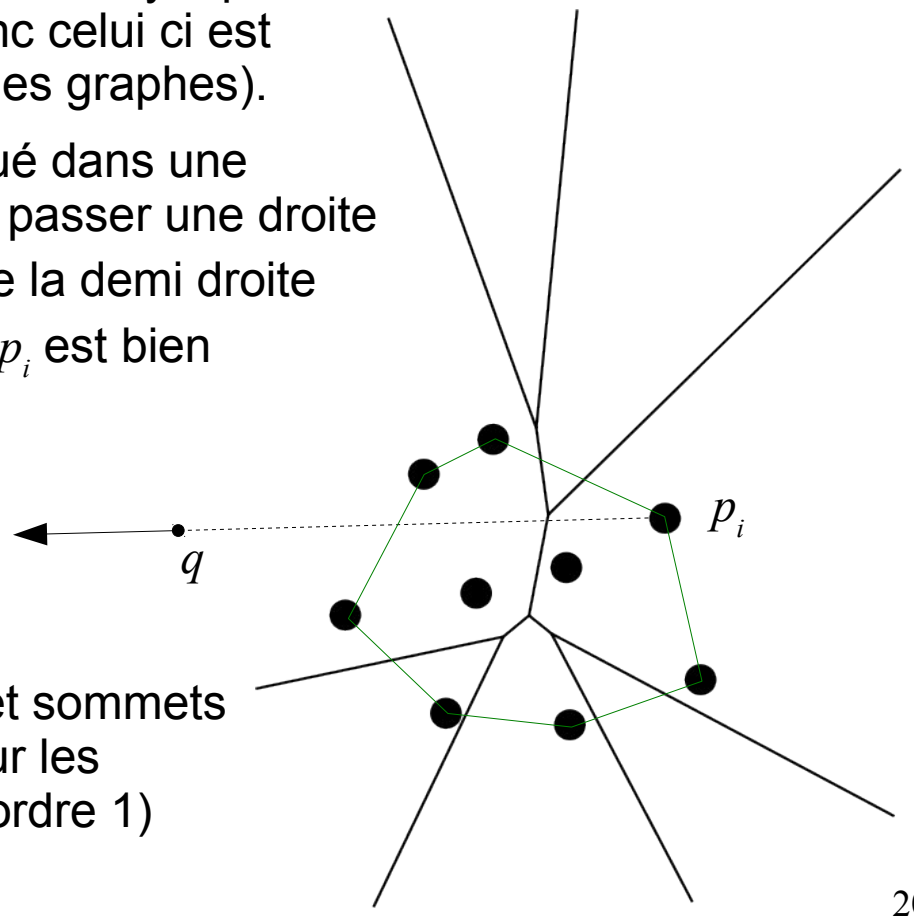
  - Un point p_i de P est associé à une cellule du DDM de P si il fait partie de l'enveloppe convexe de P.

    Preuve triviale : un point p situé à l'intérieur de l'enveloppe convexe de P n'est jamais le point le plus éloigné d'un autre point q quelconque : prendre une droite joignant p et q, forcément il existe des points sur la droite et sur la frontière de l'enveloppe convexe tels que la distance à q est supérieure.



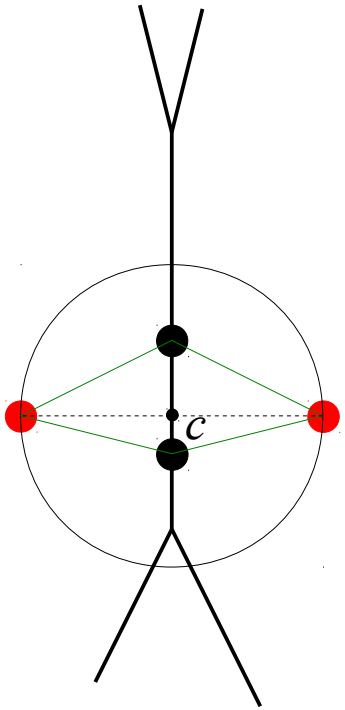
- Propriétés du DDM
  - Les cellules sont infinies, donc il n'y a pas de cycles dans le DDM, donc celui ci est réduit à un arbre (au sens des graphes).

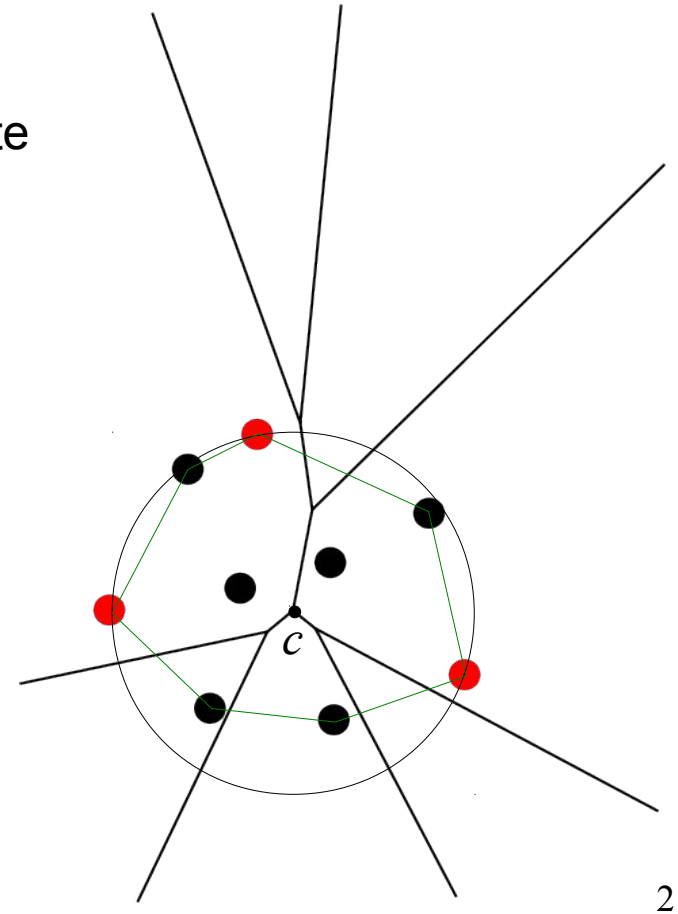
    Preuve : par un point q ; situé dans une cellule associée à p_i, on fait passer une droite (qp_i). Pour tous les points de la demi droite issue de q à l'opposé de p_i, p_i est bien le point le plus éloigné.



  - Il y a $O(n)$ cellules, arêtes, et sommets (preuve similaire à celle pour les diagrammes de Voronoï d'ordre 1) en particulier, $n_a \leq 2n-3$

- Propriétés du DDM
  - Le centre c du cercle circonscrit à l'ensemble P est situé sur un sommet du DDM ; ou exactement sur le milieu d'un segment joignant deux sites associés à une arête du DDM.





- Calcul du diagramme de distance maximale

  - Calcul de l'enveloppe convexe de P. Les sommets situés sur l'enveloppe convexe sont ordonnés dans un cycle $s_0 \ldots s_{h-1}$ (cf 1er cours)

  - On « mélange » les sommets situés sur l'enveloppe convexe aléatoirement. Soit $p_0 \ldots p_{h-1}$ cet ordre.

  - On prend les sommets un à un dans l'ordre : k=0... h–4 ($p_{h-1} \ldots p_3$)

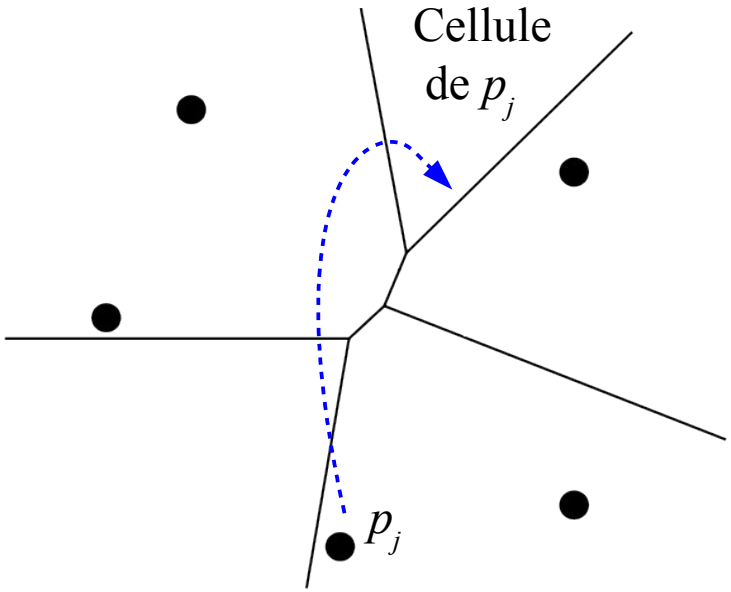
  - On stocke leur précédent (sens horaire) et leur successeur (sens antihoraire) :
    à $p_{h\text{-}k\text{-}1}$ correspond s_j, et on y associe $s_{j-1 \bmod (h\text{-}k)}$ et $s_{j+1 \bmod (h\text{-}k)}$

  - On retire $p_{h\text{-}k\text{-}1}$ (donc s_j) du cycle (en mettant à jour le cycle …).
    Ansi , $p_{h\text{-}k\text{-}1}$ ne peut plus être le précédent ou le successeur des points éliminés par la suite

  - Au final ; on obtient une structure qui permet ; partant de 4 points de l'enveloppe convexe, de reconstruire celle ci point par point tout en maintenant en tout temps l'ordre dans le cycle.

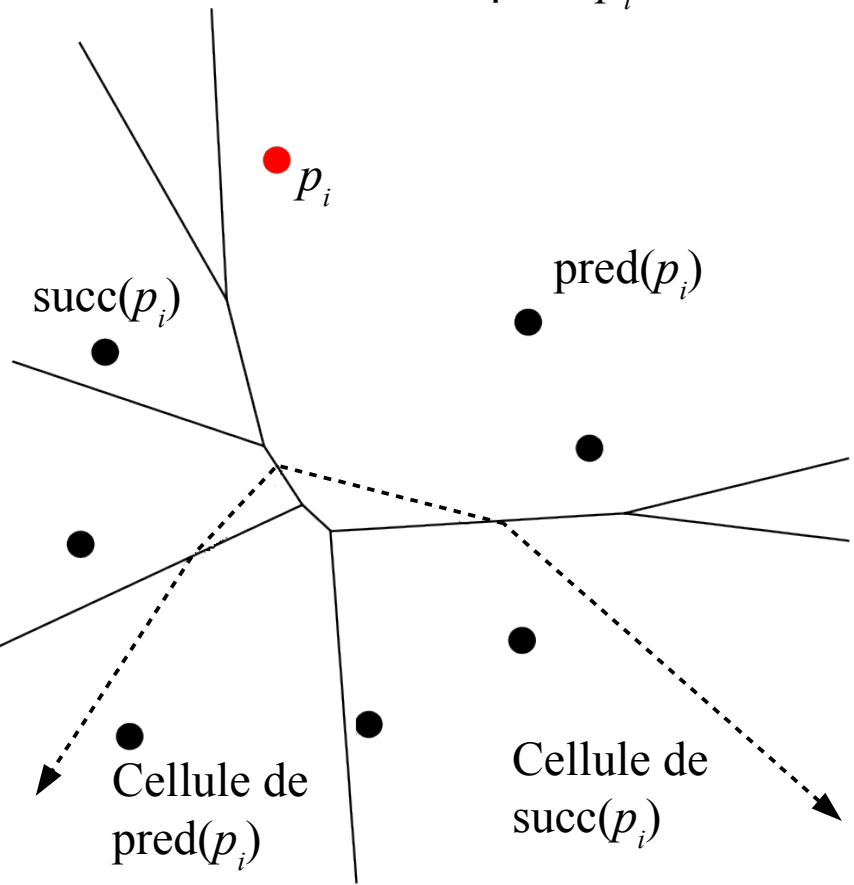
- Calcul du diagramme de distance maximale

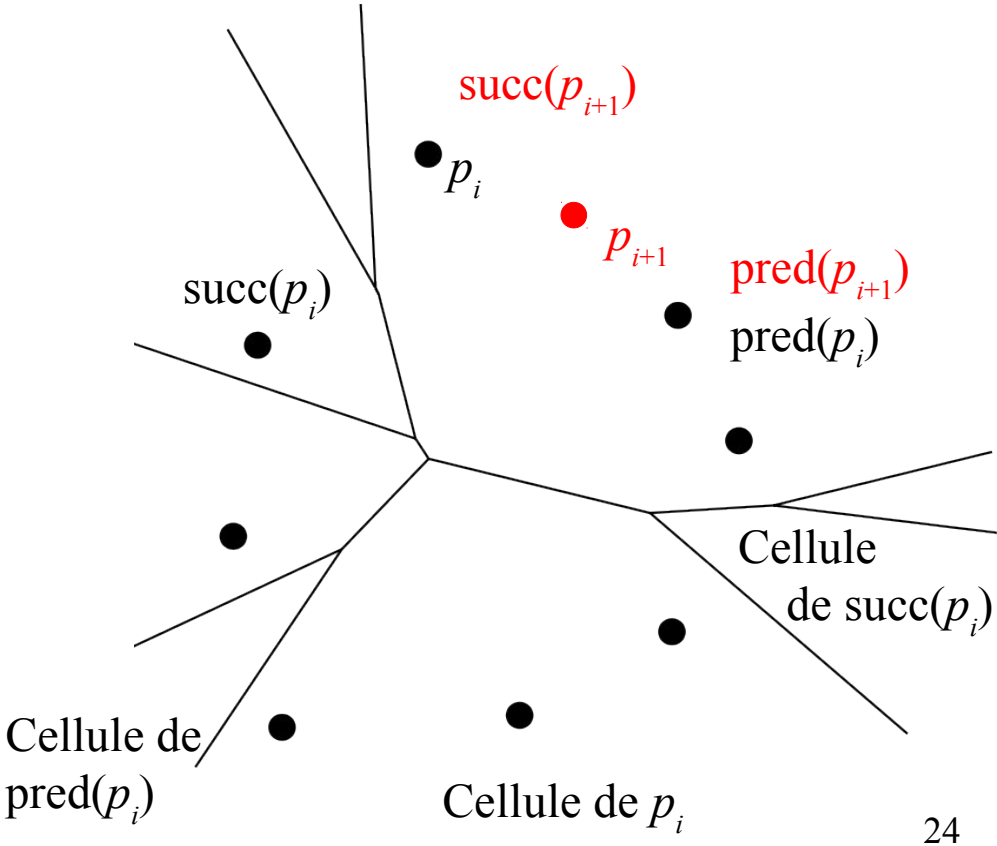
  - Associé à un point p_j, on conserve un pointeur sur l'arrête la plus dans le sens anti-horaire de la cellule associée à p_j.



- Calcul du diagramme de distance maximale

  - Approche incrémentale : à partir du DDM pour p_{i-1} , on construit le DDM pour p_i





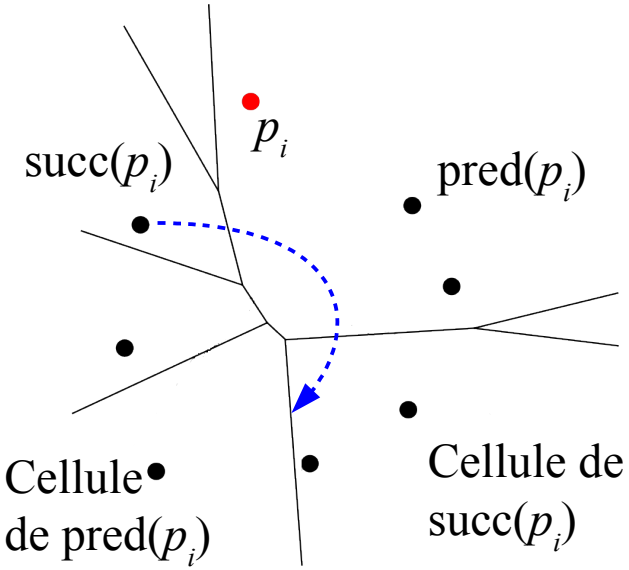
- Calcul du diagramme de distance maximale

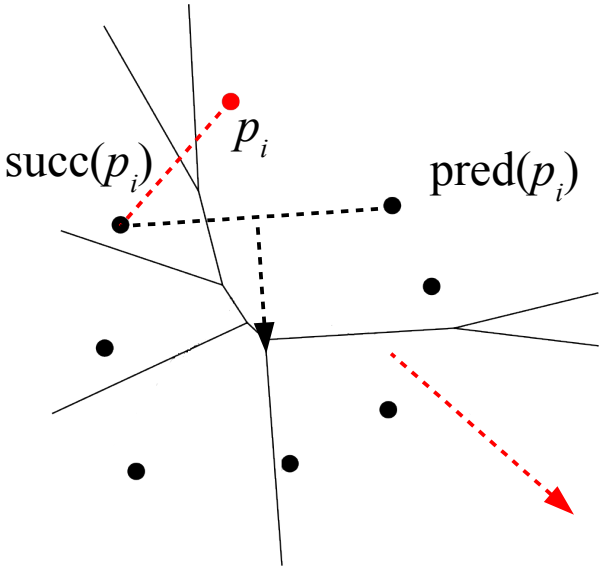
  - La cellule pour p_i vient « entre » celle de $\mathrm{pred}(p_i)$ et $\mathrm{succ}(p_i)$

  - Ces deux cellules sont séparées par une arête semi-infinie qui est une partie de la bisectrice de $\mathrm{pred}(p_i)$ et $\mathrm{succ}(p_i)$

  - $\mathrm{succ}(p_i)$ possède en fait un pointeur sur cette arête.

  - La bissectrice de pi et de succ(pi) induit une nouvelle arête semi-infinie (en rouge)



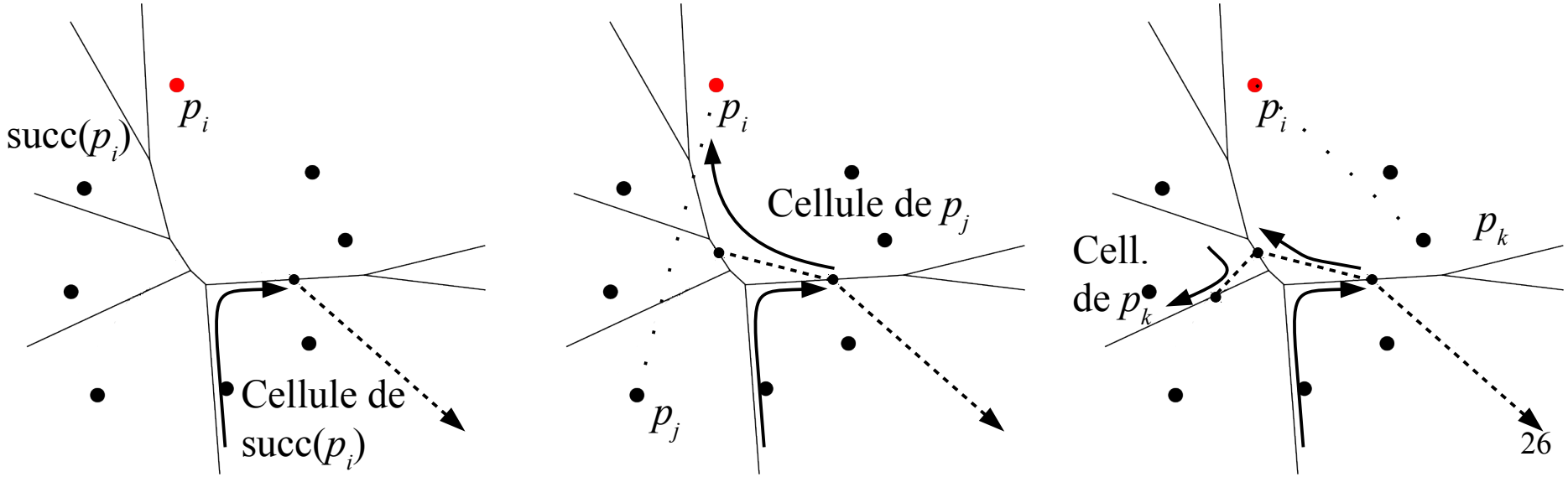


- Calcul du diagramme de distance maximale

  - On traverse la cellule de $\mathrm{succ}(p_i)$ dans le sens horaire pour savoir quelle arête est intersectée par la nouvelle arête semi-infinie.

  - De l'autre côté de cette arête se trouve une autre cellule, correspondant à un autre point p_j

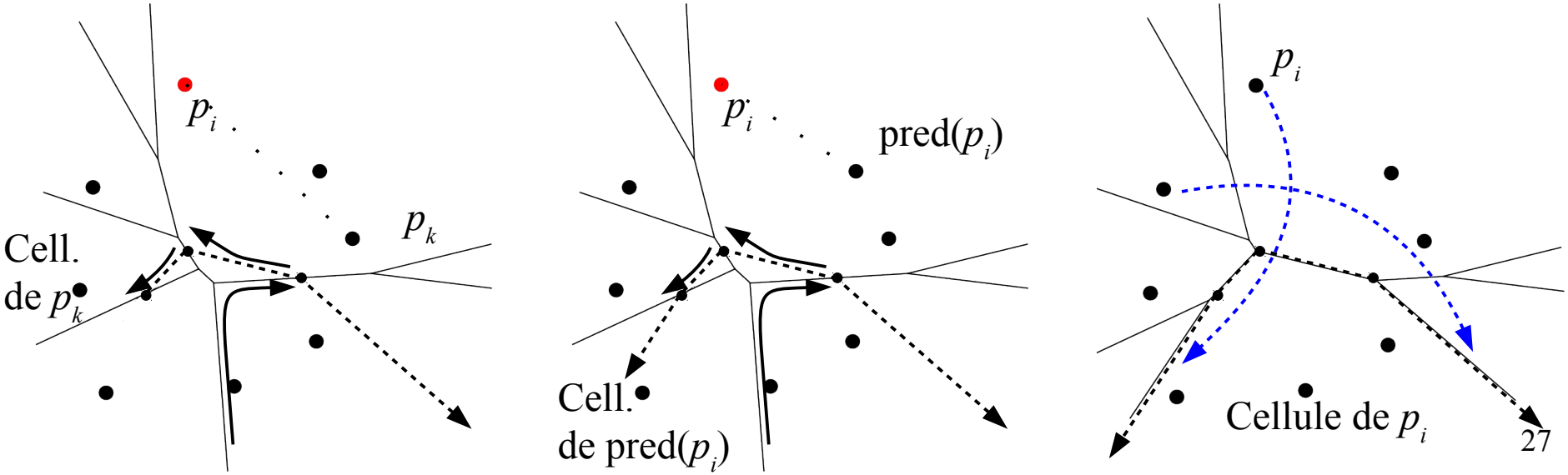
  - On parcourt de nouveau cette cellule dans le sens horaire et on calcule l'intersection avec la bissectrice de p_i-p_j



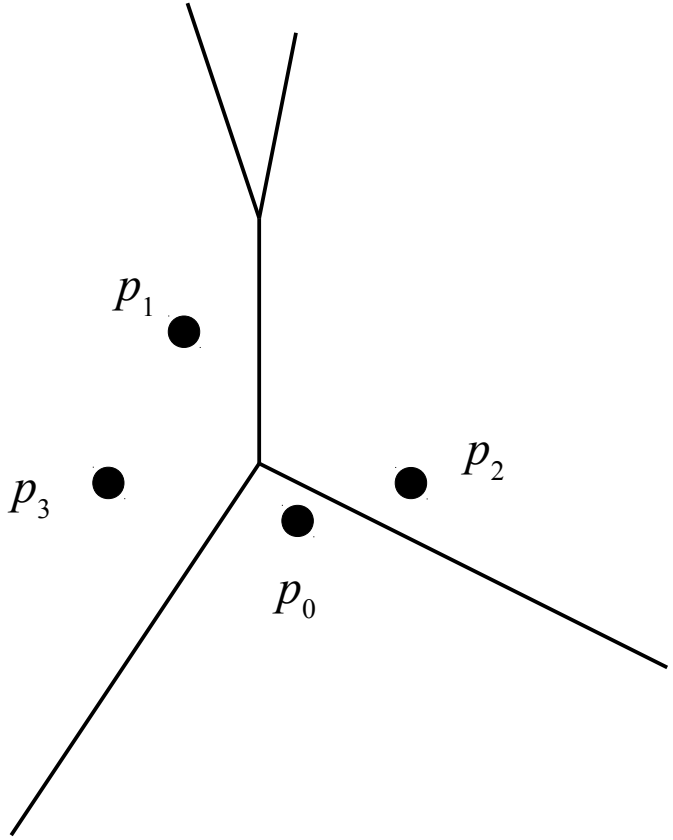
- Calcul du diagramme de distance maximale

  - on répète ces opérations jusquà rencontrer pred(pi), qui donne l'autre arrête semi – infinie bornant la cellule de pi.

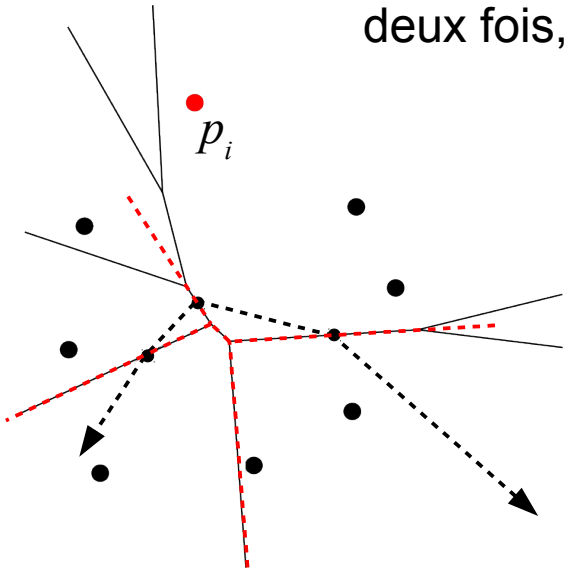
  - Ne pas oublier de mettre à jour les pointeurs entre les points pi déjà existants et les arrêtes semi-infinies correspondantes.



- Calcul du diagramme de distance maximale
  - Le calcul est initialisé sur les 4 sommets initiaux $p_0 \ldots p_3$ dont le DDM est trivial

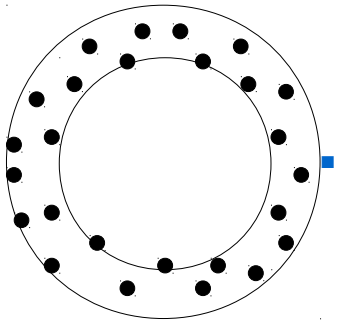


- Complexité de l'algorithme
  - Enveloppe convexe initiale : $O(n \log n)$, et contient au pire les n points.
  - Après l'insertion de p_i, la cellule correspondante comporte k arêtes.
  - Par conséquent, pour construire cette cellule à partir du DDM d' après l'insertion de p_{i-1} , l'algorithme a parcouru k cellules voisines.
  - Ces k cellules voisines sont séparées par un sous-arbre qui admet au plus $2k-3$ arêtes distinctes, et au pire, on parcourt toutes ces arêtes deux fois, soit au total, $4k-6$ tests (linéaire)

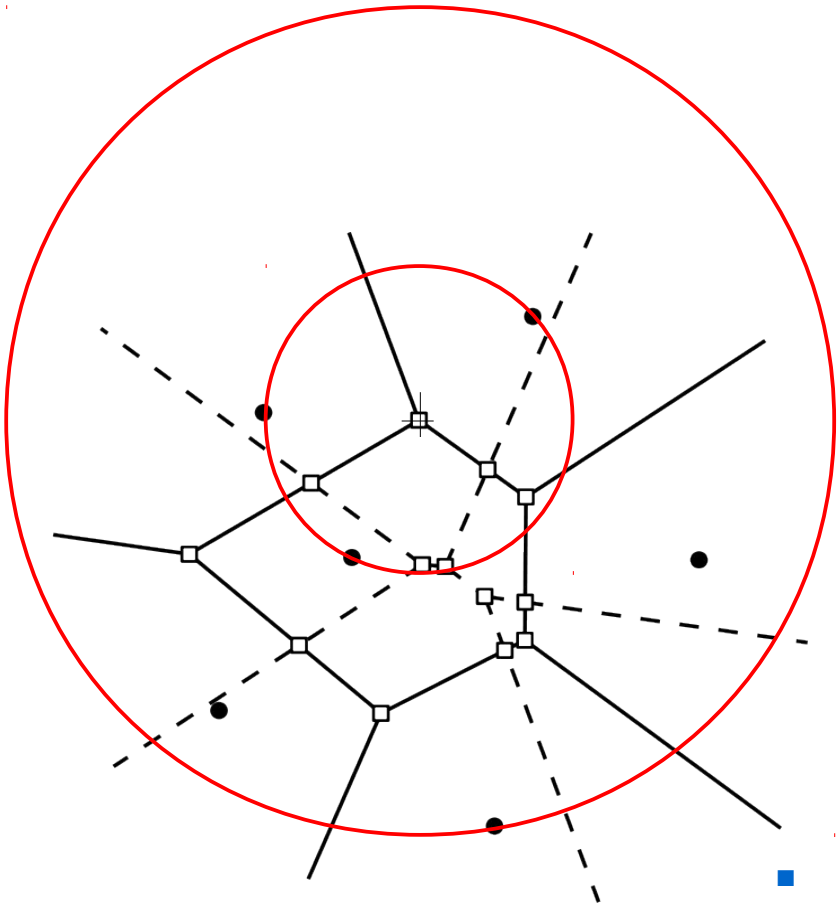


  - Le DDM à l'étape i a au plus $2i-3$ arêtes. Chaque arête sépare deux cellules. Donc en moyenne, chaque cellule a moins que $(4i-6)/i$ arêtes, soit moins que 4.
  - Chaque point p_i a la même probabilité d'être ajouté à la fin, donc globalement, la complexité est $O(n)$ pour la construction du DDM.
  - Globalement , la complexité est en $O(n \log n)$ avec un stockage en $O(n)$.

- Application du diagramme de distance maximale
  - Métrologie (en relation avec le DDV classique)
    - Déterminer les deux cercles min-max de même centre approximant au mieux un nuage de points de mesure (e.g. pièce cylindrique)
    - Plusieurs cas :
      - Le cercle interne passe par 3 (ou plus) points – son centre est donc sur un des sommets du DDV classique. Le cercle externe passe dans le cas général par un seul point (celui correspondant à la cellule du DDM)
      - Le cercle externe passe par 3 (ou plus) points – son centre est donc sur un des sommets du DDM. Le cercle interne passe dans le cas général par un point (celui correspondant à la cellule du DDV)
      - Enfin, le cercle externe passe par deux points, ainsi que le cercle interne. Le centre est donc à l'intersection des arêtes du DDM et du DDV.
    - Afin de calculer ces cercles, il faut calculer l'arrangement entre le DDV et le DDM du même ensemble de points.



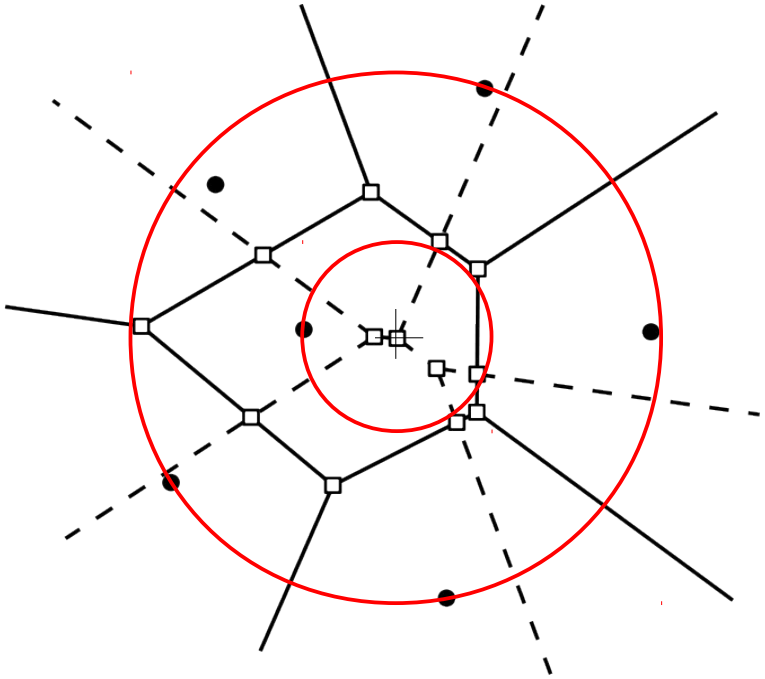
Diagrammes de Voronoï



- Pour chaque sommet du DDM, trouver le point de P qui est le plus proche (cela donne 4 points en tout)
- Pour chaque sommet du DDV, trouver le point de P qui est le plus éloigné (4 points encore)
- Pour chaque intersection entre le DDV et le DDM, déterminer les 4 points de P
- Pour tous les groupes de candidats, choisir celui qui donne l'anneau de largeur minimale.

- Ceci peut être effectué globalement en $O(n^2)$
  - Il y a $O(n^2)$ sites supplémentaires dus à l'intersection des deux graphes.

Diagrammes de Voronoï



- Cas présent : 13 combinaisons à vérifier.

- Diagrammes en norme L^p