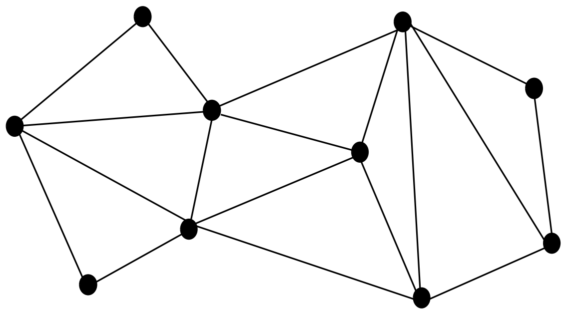
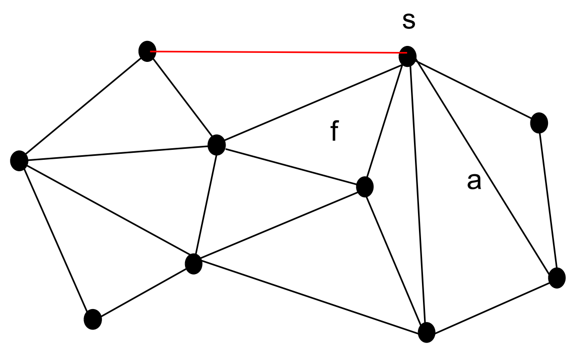
Méthodes applicables aux modèles continus

- Triangulation
  - Définition
    - Soit E un ensemble de n points de R^2, on appelle triangulation de E un ensemble de triangles dont les sommets sont les points de E, vérifiant:
      - L'intersection de 2 triangles est soit vide, soit 1 arête commune aux 2 triangles, soit 1 sommet commun aux 2 triangles

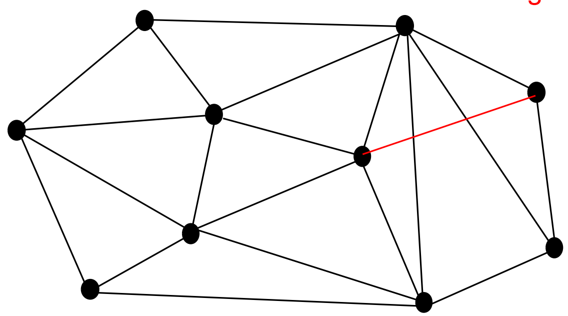


Méthodes applicables aux modèles continus

- Triangulation maximale
  - Soit T = (S, F, A), une triangulation constituée d’un ensemble de sommets (S), de faces (F) et d’arêtes (A).
  - T est une triangulation maximale si tout segment n’appartenant pas à A joignant deux points quelconques de S coupe au moins une arête de A ailleurs qu’en ses extrémités
  - 1 triangulation maximale pave l’enveloppe convexe



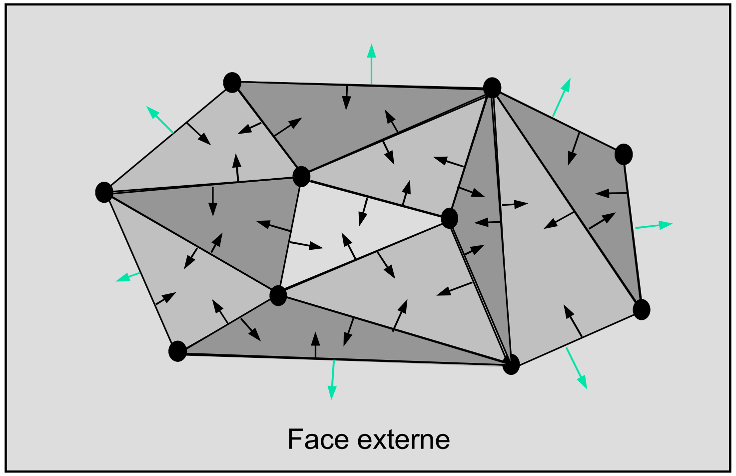
Triangulation non maximale



Triangulation maximale

Méthodes applicables aux modèles continus

- Dénombrement d'une triangulation
  - Chaque arête est commune à 2 faces
  - Si on relie par une flèche chaque arête à ses deux triangles adjacents, chaque triangle reçoit 3 flèches et la face externe reçoit autant de flèches qu'elle compte d'arêtes.



Méthodes applicables aux modèles continus

- Soit,
  - n_s = le nombre de sommets
  - n_a = le nombre d'arêtes
  - n_{a_ext} = le nombre d'arêtes de la face externe
  - n_t = le nombre de triangles
  - n_f = le nombre total de faces (triangles (n_t) + face externe)
    d'où $n_f = n_t + 1$ donc $n_t = n_f - 1$
  - On a : $2n_a = n_{a\_ext} + 3n_t = n_{a\_ext} + 3(n_f-1)$
  - La relation d'Euler étant : $n_s + n_f – n_a = 2$ on a $n_f = 2 - n_s + n_a$ d'où $2n_a = n_{a\_ext} + 3(n_f-1) = n_{a\_ext} + 3*(2 - n_s + n_a - 1)$
  d'où

    $\mathbf{n_a = 3(n_s - 1) – n_{a\_ext}}$

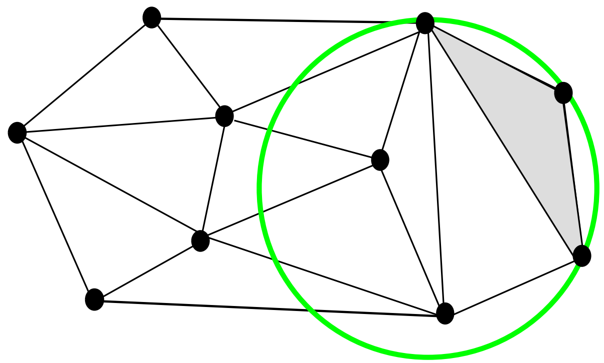
    et

    $\mathbf{n_t = 2(n_s - 1) – n_{a\_ext}}$

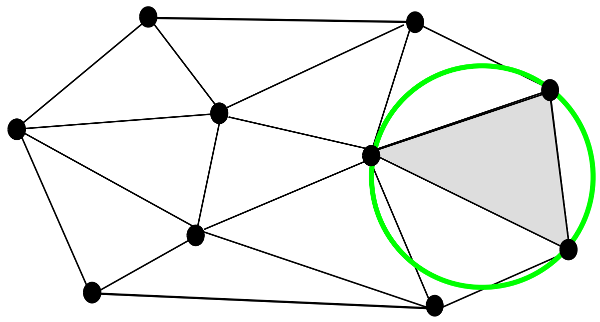
  - Nb : Si tous les sommets sont sur la face externe (polygone convexe sans trous => $n_{a\_ext} = n_s$) on a :
    - $n_t = n_s - 2$ et $n_a = 2\,n_s - 3$
- Le nombre d'arêtes et de triangles est donc toujours en O(n)

Méthodes applicables aux modèles continus

- Triangulation de Delaunay
  - Soit E un ensemble de n points $p_1, \ldots, p_n$ de R^2, on appelle triangulation de Delaunay de E, notée Del(E) :
    - Une triangulation de E dont tous les triangles sont circonscrits par un cercle n'englobant aucun des p_i



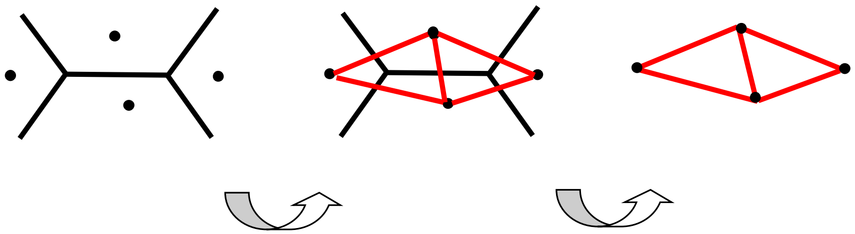
Triangulation maximale

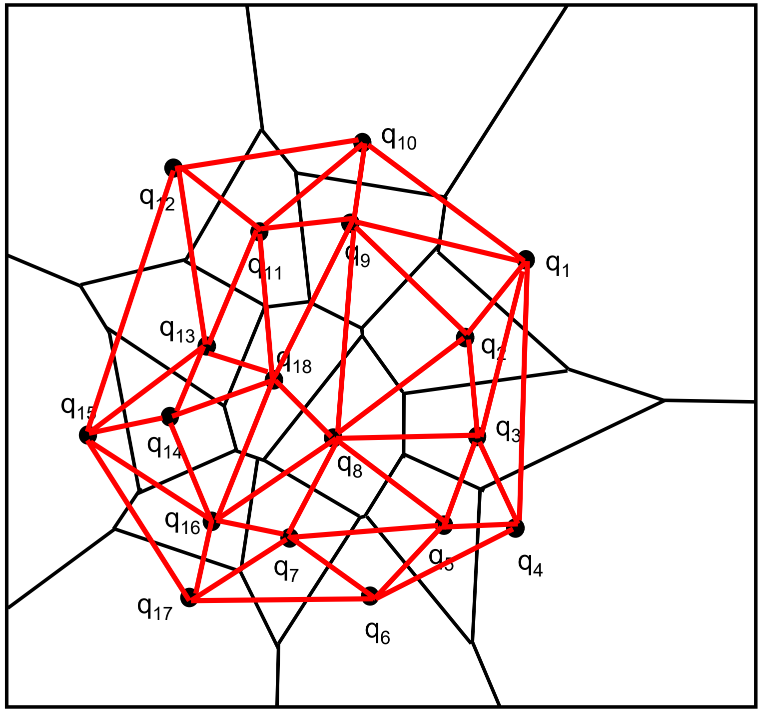


Triangulation de Delaunay

Méthodes applicables aux modèles continus

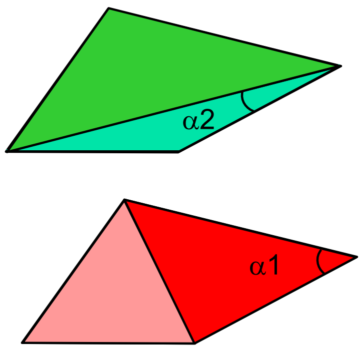
- Triangulation de Delaunay
  - Elle peut être construite en reliant par un segment toutes les paires de sites dont les régions de Voronoï correspondantes sont voisines
  - On démontre que la triangulation de Delaunay est le dual du Diagramme de Voronoï.





Méthodes applicables aux modèles continus

- Triangulation de Delaunay
  - Propriétés
    - La triangulation de Delaunay est unique car c'est le dual de Voronoï qui lui-même est unique.
    - Toute arête de la frontière de l'enveloppe convexe de E est une arête de Del(E)
    - Critère du cercle : Le cercle circonscrit à un triangle de Del(E) ne contient aucun site en son intérieur.
    - Cette propriété est équivalente à celle de l'angle min-max :
      - D'après sa définition la triangulation de Delaunay est parmi toutes les triangulations maximales de E celle qui maximise l'angle minimum de tous les triangles…
      - $\alpha 1 >> \alpha 2$

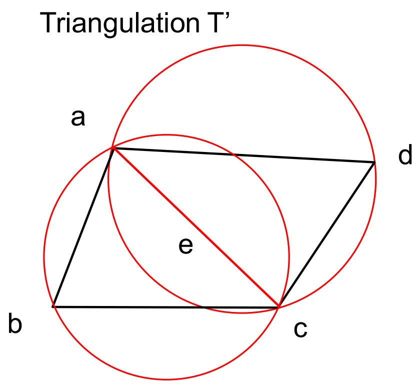


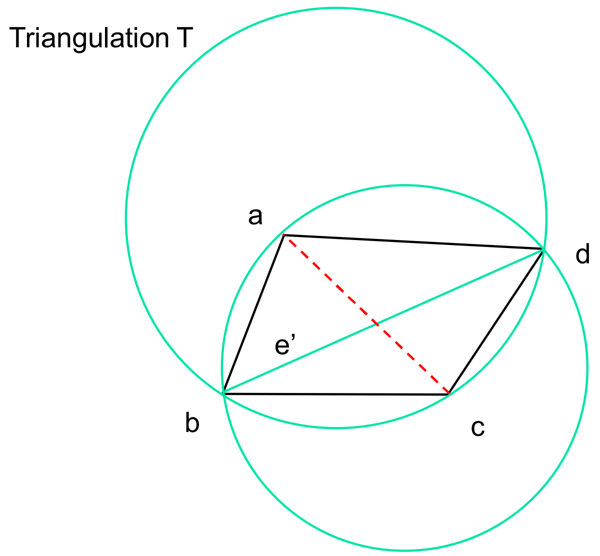
Méthodes applicables aux modèles continus

- Triangulation de Delaunay
  - Critère de l'angle Min-Max
    - Il est équivalent de choisir la triangulation qui maximise l'angle minimal ou de choisir celle qui est donnée par le "critère du cercle"
    - Soit T une triangulation maximale de E. Si pour chaque triangle on retient l'angle minimum, on peut alors classer plusieurs triangulations maximales de E :
      - $T_1 < T_2 < T_3$
      - Dans ce cas la triangulation de Delaunay est celle qui maximise l'angle minimum (ici T3)

Méthodes applicables aux modèles continus

- Procédure LOP (Locally Optimal Procedure) :
  - Soit e une arête interne de DEL(E) et Q le quadrilatère formé par les deux triangles partageant e.
  - e est localement optimal si son remplacement par e’ brise le critère du cercle.
  - Si T est une triangulation maximale de E, et T’ une triangulation maximale de E obtenue après succès de la procédure LOP, alors T < T’



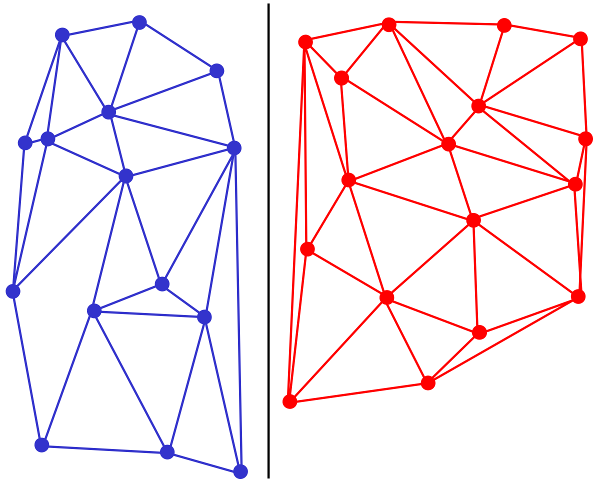


Méthodes applicables aux modèles continus

- Triangulation de Delaunay
  - Algorithmes de construction
    - Algorithme par balayage [Fortune 87]
    - Algorithme par Divide and Conquer [Lee 80]
  - Complexité
    - On montre, de la même manière que pour l'enveloppe convexe, que la triangulation de Delaunay se réduit au problème du tri.

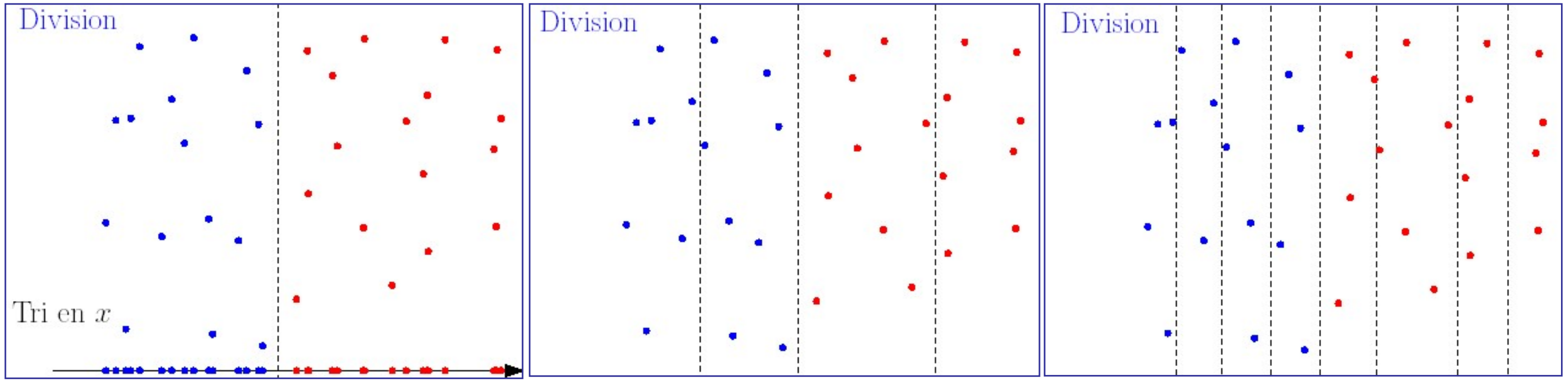
Méthodes applicables aux modèles continus

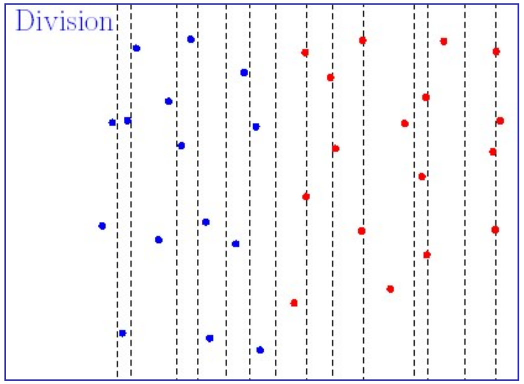
- Algorithme par divide and conquer (Division-Fusion)
  - Les étapes classiques
    - On trie les points pour diviser l'ensemble
    - On fusionne chaque triangulation gauche et droite
      - On doit supprimer des arêtes rouges et des arêtes bleues
      - On doit rajouter des arêtes (bleues-rouges)
    - On démontre que la fusion est linéaire, donc l'algorithme est en O(n log n)



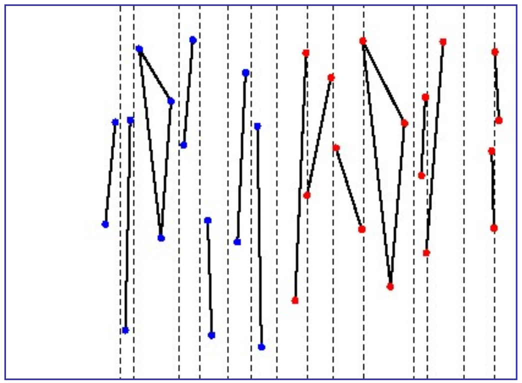
Méthodes applicables aux modèles continus

- Algorithme par divide and conquer (Division-Fusion)
  - Division





Puis triangulation des sous-ens obtenus



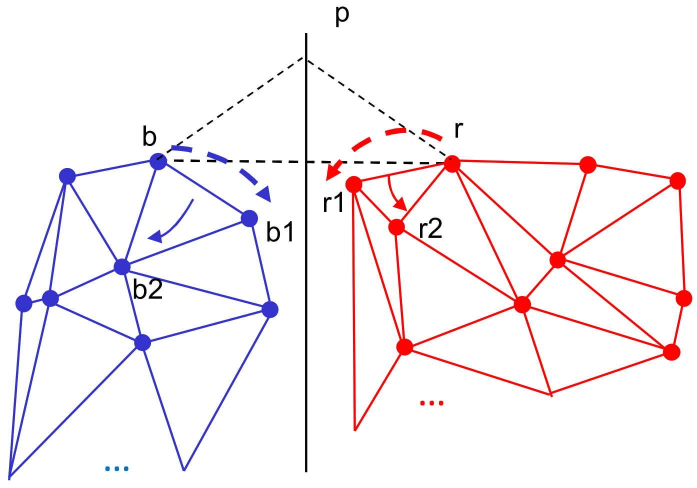
Méthodes applicables aux modèles continus

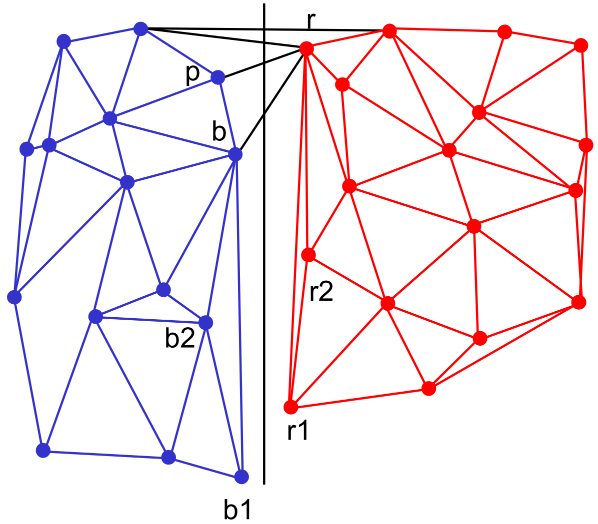
- Algorithme par divide and conquer
  - Fusion
    - On a un triangle courant (p, b, r)
      b et r sont les points de la tangente haute.
    - Tant que b et r ne sont pas les points de la tangente basse
      - r1 est le point extrémité de l'arête issue de r de l'env convexe,
      - r2 est le point extrémité de l'arête suivante issue de r
      - idem pour b1 et b2

      - Tant que r2 appartient à C(r, b, r1) et r2≠r et r2≠p
        - Rajouter l'arête r, r1 dans la liste des arêtes rouges à supprimer
        - r1 prend la place de r2

      - Tant que b2 appartient à C(r, b, b1) et b2≠b et b2≠p
        - Rajouter l'arête b, b1 dans la liste des arêtes bleues à supprimer
        - b1 prend la place de b2

      - Choisir l'arête b, r1 ou r, b1 à rajouter en fonction du critère du cercle



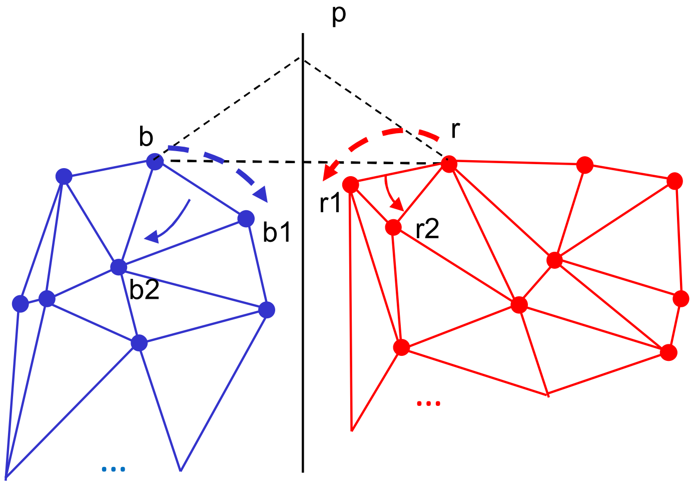


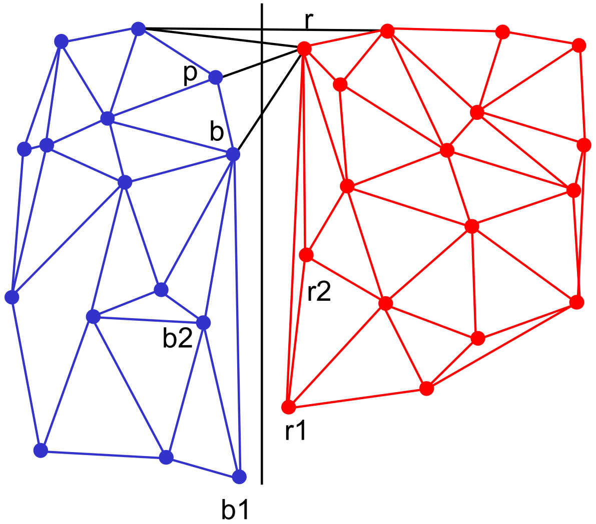
Méthodes applicables aux modèles continus

- Algorithme par divide and conquer
  - Fusion
    - Choisir l'arête b, r1 ou r, b1 à rajouter en fonction du critère du cercle
      - Si b1 n'est pas à l'intérieur du cercle C(r,b,r1) et si r2 n'est pas à l'intérieur du cercle C(r,b,r1)

      ⇒ choix de l'arête b,r1
        - ✓ p prend la place de r
        - ✓ r prend la place de r1
        - ✓ On supprime la liste des arêtes rouges marquées comme à supprimer
        - ✓ On réinitialise la liste des arêtes bleues marquées comme à supprimer
      - Si r1 n'est pas à l'intérieur du cercle C(r,b,b1) et si b2 n'est pas à l'intérieur du cercle C(r,b,b1)

      ⇒ choix de l'arête r,b1
        - ✓ p prend la place de b
        - ✓ b prend la place de b1
        - ✓ On supprime la liste des arêtes bleues marquées comme à supprimer
        - ✓ On réinitialise la liste des arêtes rouges marquées comme à supprimer





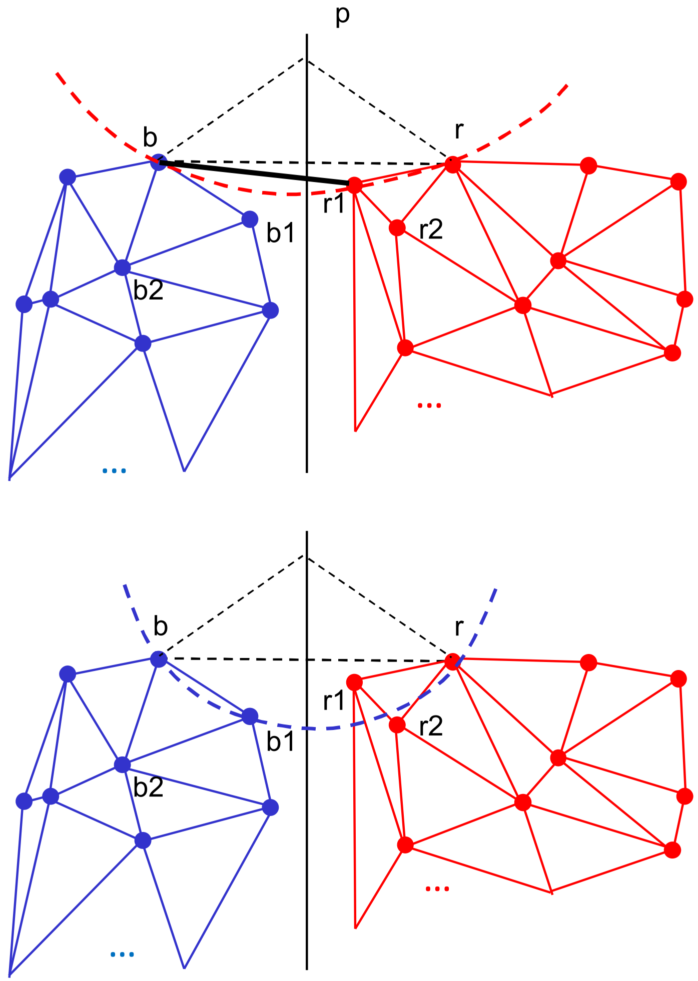
Méthodes applicables aux modèles continus

- Algorithme par divide and conquer
  - Fusion illustration par l'exemple
    - r2 ∉ C(r, b, r1)
    - b2 ∉ C(r, b, b1)
    - Choix de l'arête b, r1 ou r, b1 (critère du cercle) ?

      b1 n'est pas à l'intérieur du cercle C(r,b,r1)

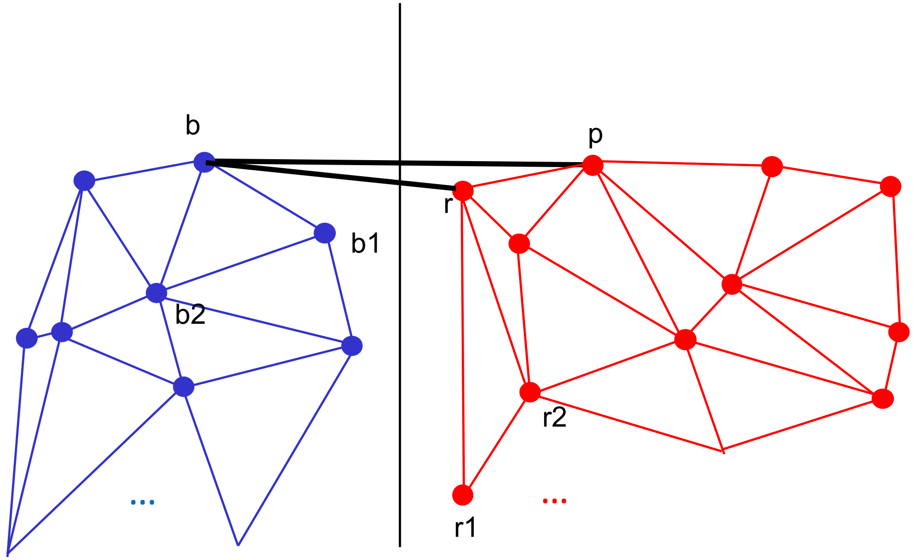
      r2 n'est pas à l'intérieur du cercle C(r,b,r1)

⇒ On choisit arête b r1



Méthodes applicables aux modèles continus

- Algorithme par divide and conquer
  - Fusion illustration par l'exemple (suite)
    - Arête b r1 choisie
    - ⇒ p prend la place de r
    - ⇒ r prend la place de r1
    - ⇒ Suppression liste des arêtes rouges à supp (ens vide)
    - ⇒ Réinitialisation liste des arêtes bleues à supp (ens vide)



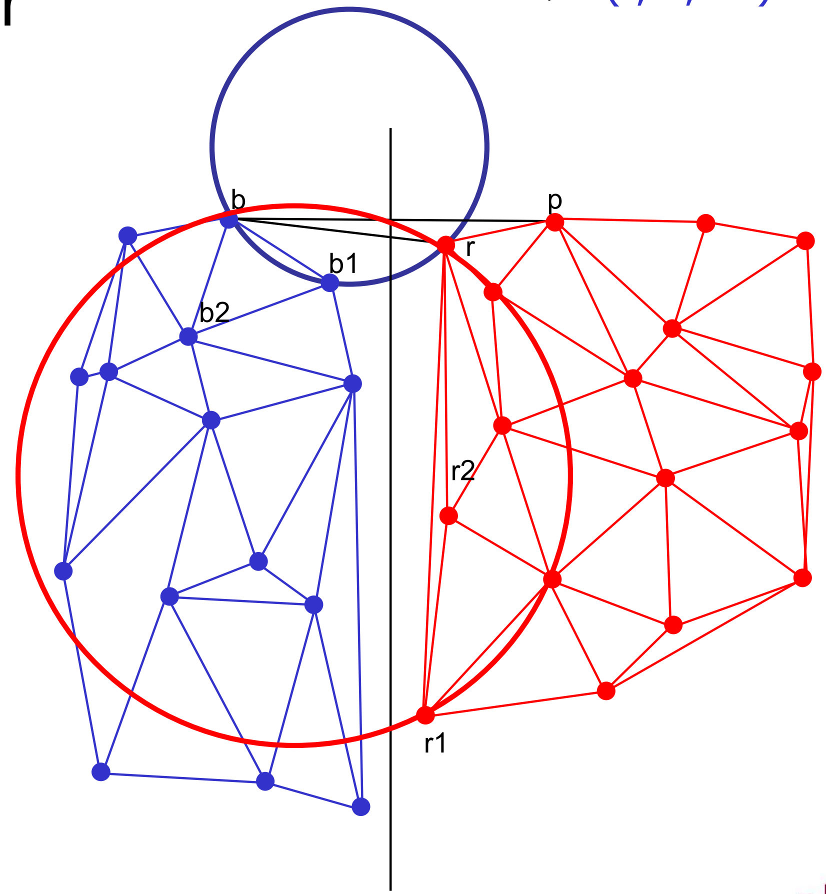
Méthodes applicables aux modèles continus

- Algorithme par divide and conquer
  - Fusion illustration par l'exemple (suite)

r2 $\in$ C(r, b, r1)

⇒Rajout arête r,r1 dans la liste des arêtes rouges à supprimer
⇒r1 prend la place de r2

b2 $\notin$ C(r, b, b1)



Méthodes applicables aux modèles continus

- Algorithme par divide and conquer
  - Fusion illustration par l'exemple (suite)

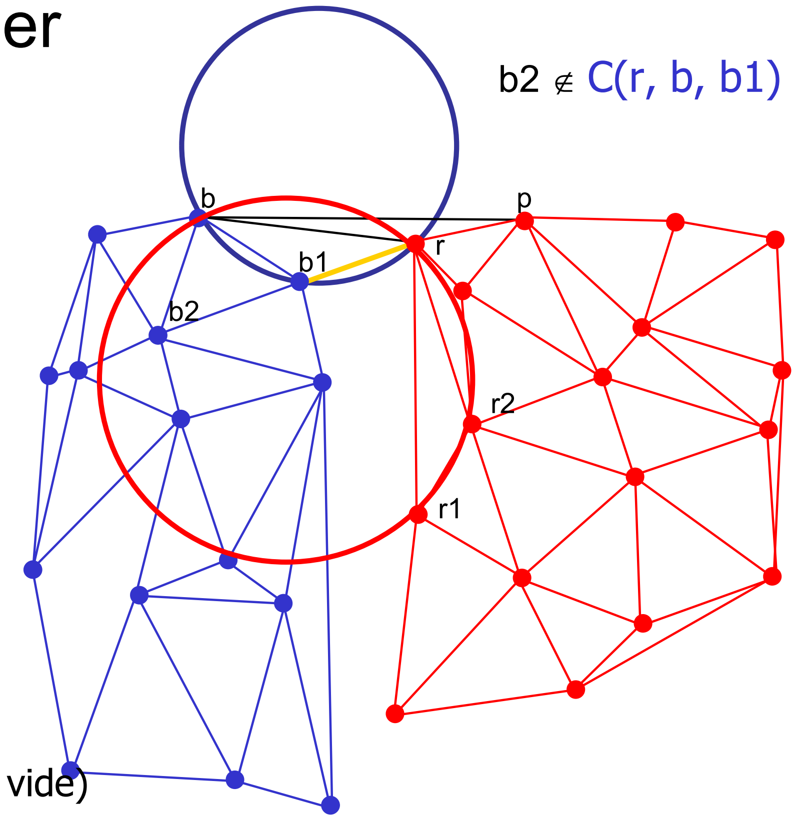
r2 ∉ C(r, b, r1)

Choix b r1 ou r b1 ?

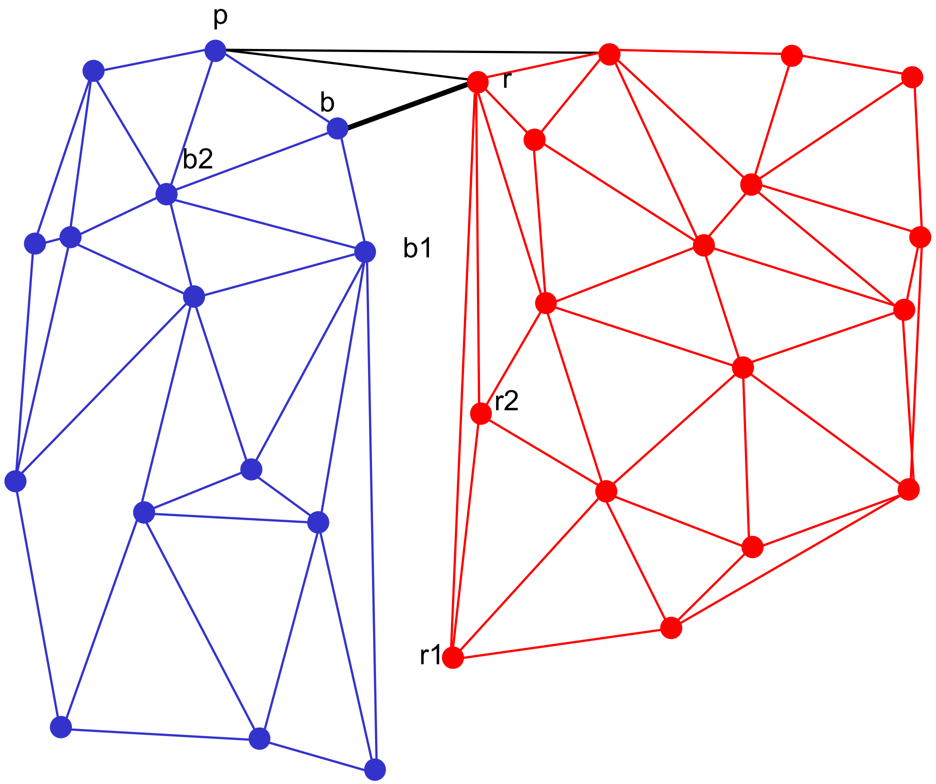
b1 **est à l'intérieur** de C(r, b, r1)
Et r2 n'est pas à l'intérieur de C(r, b, b1)

r1 n'est pas à l'intérieur de C(r, b, b1)
Et b2 n'est pas à l'intérieur de C(r, b, r1)

⇒Arête r b1 est choisie
⇒p prend la place de b
⇒b prend la place de b1
⇒Suppression liste des arêtes bleues à supp (ens vide)
⇒Réinitialisation liste des arêtes rouges à supp



Méthodes applicables aux modèles continus



Etc …

Méthodes applicables aux modèles continus

Et ainsi de suite jusqu'à …