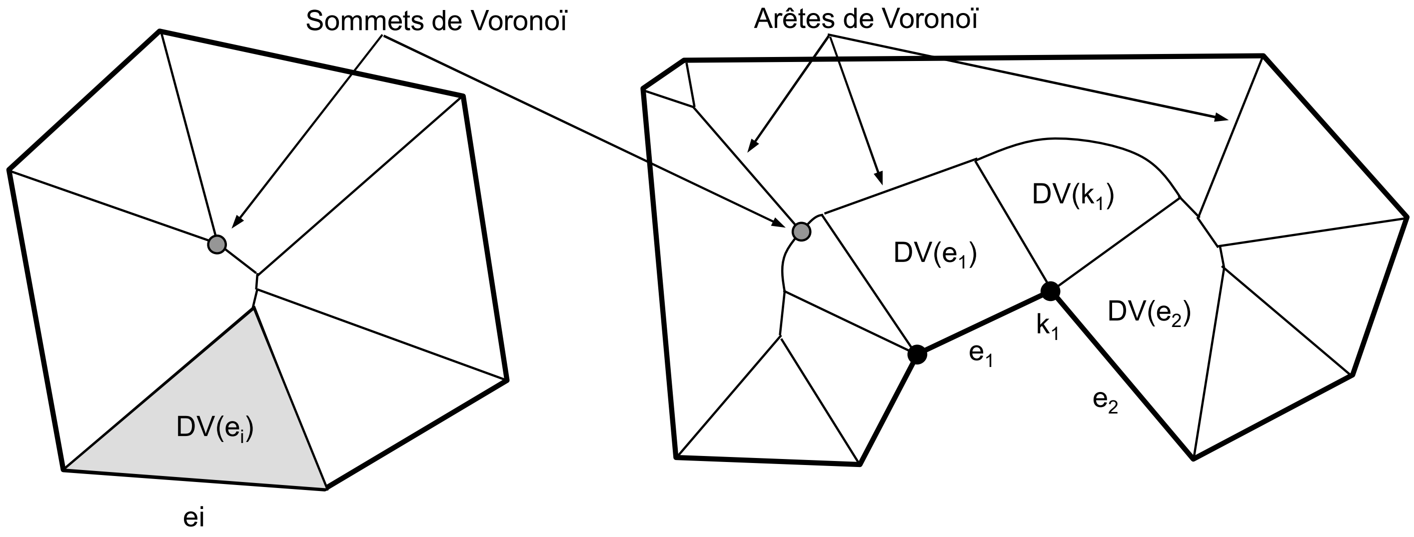
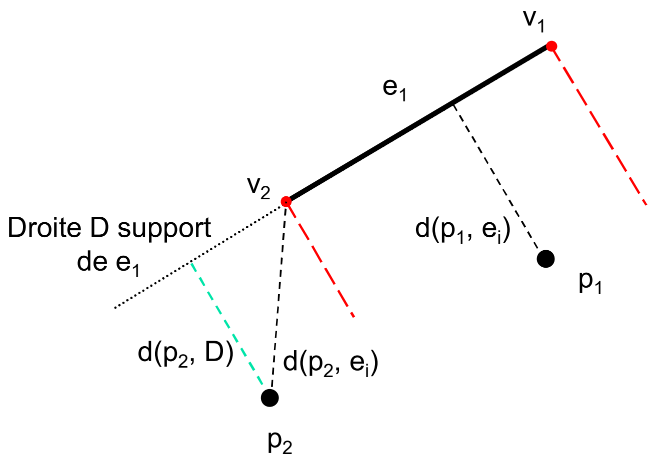
Méthodes applicables aux modèles continus

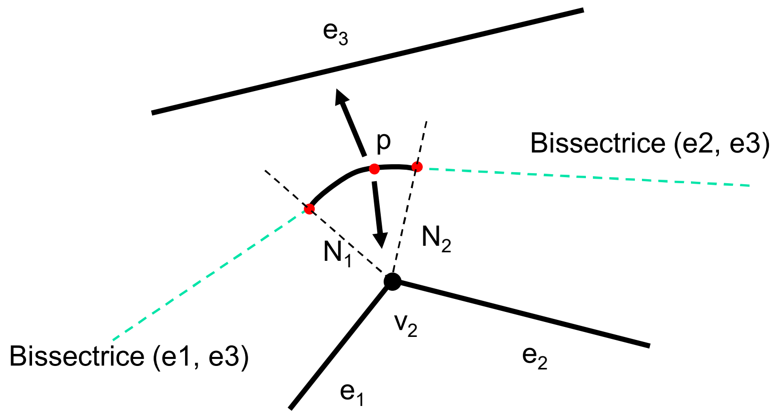
- Diagramme de Voronoï généralisé (DVG)
  - Soit C, un polygone composé de n arêtes e et de k sommets concaves.
  - 1 cellule de Voronoï, DV(e_i), est l'ensemble des points de C plus proches de e_i que de tout autre élément (arêtes, sommets concaves) de C.
  - 1 arête de Voronoï est associée à exactement deux cellules
  - 1 sommet de Voronoï est le point de rencontre de trois arêtes de Voronoï



Méthodes applicables aux modèles continus

- Diagramme de Voronoï généralisé (DVG)
  - Rappel : distance d'un point à un segment et points équidistants entre un point et un segment

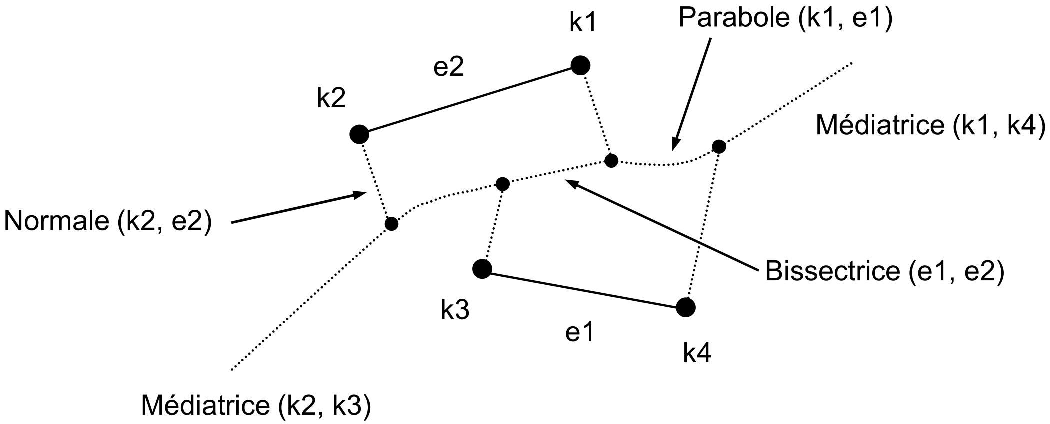




Quels sont les points P à égale distance de e1 et e3 ?
- Les points sur la bissectrice de e1,e3 si P est à gauche de la normale N1
- Les points situés sur la parabole entre e3 et v2 si P est à droite de la normale N1

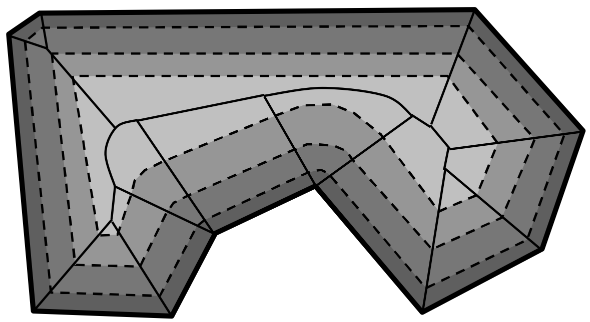
Méthodes applicables aux modèles continus

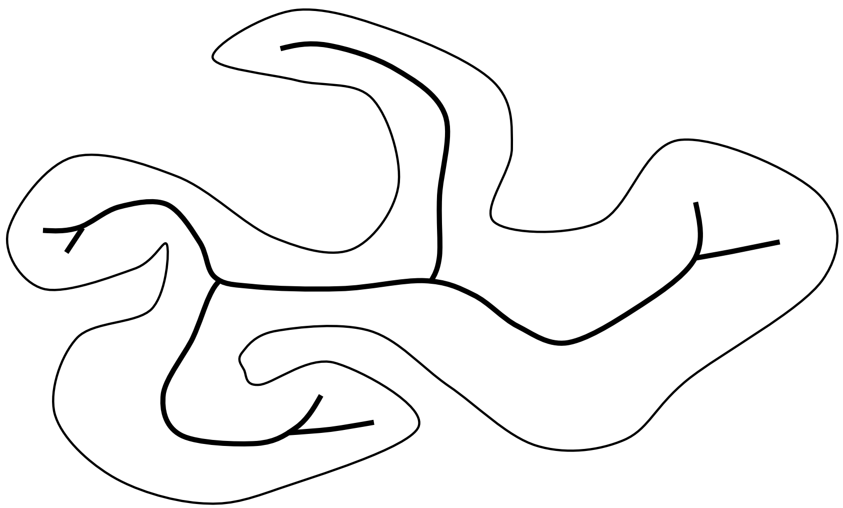
- Diagramme de Voronoï généralisé
  - 1 arête de Voronoï peut être:
    - Une **médiatrice** entre deux sommets concaves.
    - Une **bissectrice** entre deux arêtes.
    - Une **portion de parabole** entre un sommet concave et une arête.
    - Une **normale** entre un sommet concave et son arête adjacente.

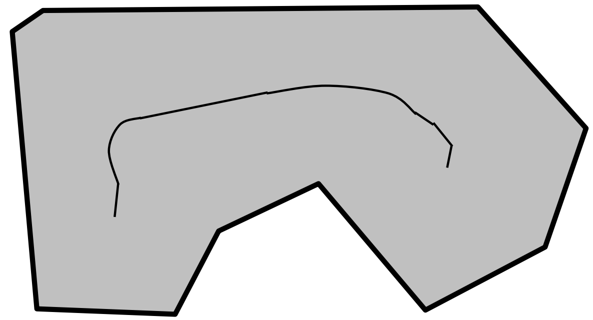


Méthodes applicables aux modèles continus

- DVG et Axe Médian (notion de squelettes)
  - Le DVG permet de tracer des contours " parallèles " au contour polygonal externe et permet ainsi de trouver le squelette de formes







Méthodes applicables aux modèles continus

- Diagramme de Voronoï et triangulation de Delaunay : Extensions

  - La construction du diagramme de Voronoï ou de la triangulation de Delaunay d'un ensemble de points permet de répondre aux questions suivantes :

    - Points les plus proches d'un site donné
    - Couple de points de distance minimale
    - Etc…

  - Une autre problématique est la recherche des « plus courts chemins entre les points ».

  - La réponse à cette question est donnée par la construction d'un **AREM** (Arbre de REcouvrement Minimum) ou « Minimum Spanning Tree » (**MST**)

Méthodes applicables aux modèles continus

- Diagramme de Voronoï et triangulation de Delaunay : Extensions

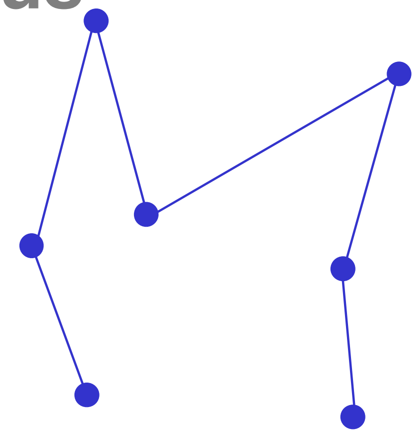
  - Un arbre de recouvrement d'un ensemble de points est un arbre qui passe exactement par tous les points de cet ensemble.

  - Le problème
    - étant donné un ensemble de N points du plan, construire un arbre de recouvrement dont la longueur cumulée des arêtes est minimale et dont les sommets sont exactement les N points.

Méthodes applicables aux modèles continus

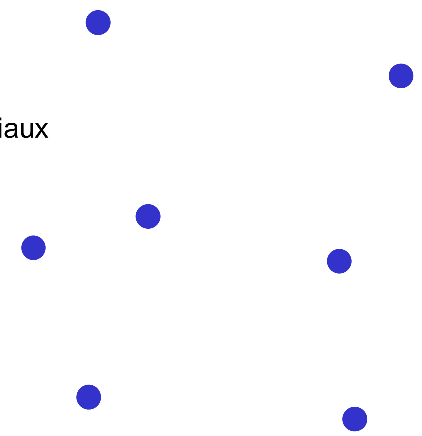
- Diagramme de Voronoï et triangulation de Delaunay : Extensions

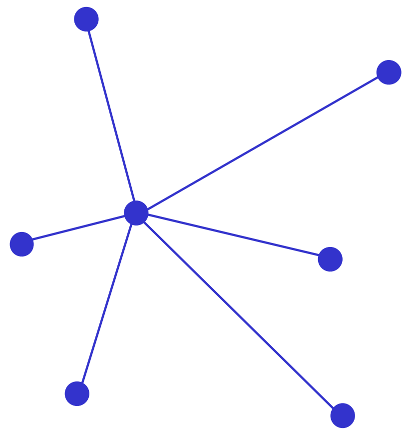
  - Exemples d'arbres recouvrant



Arbre 1

Ensemble de points initiaux

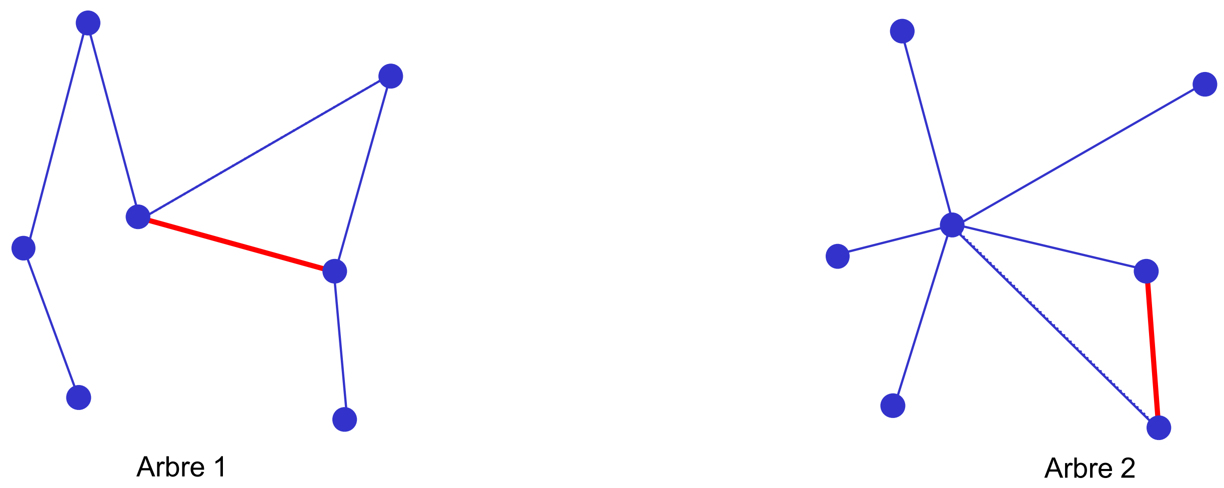




Arbre 2

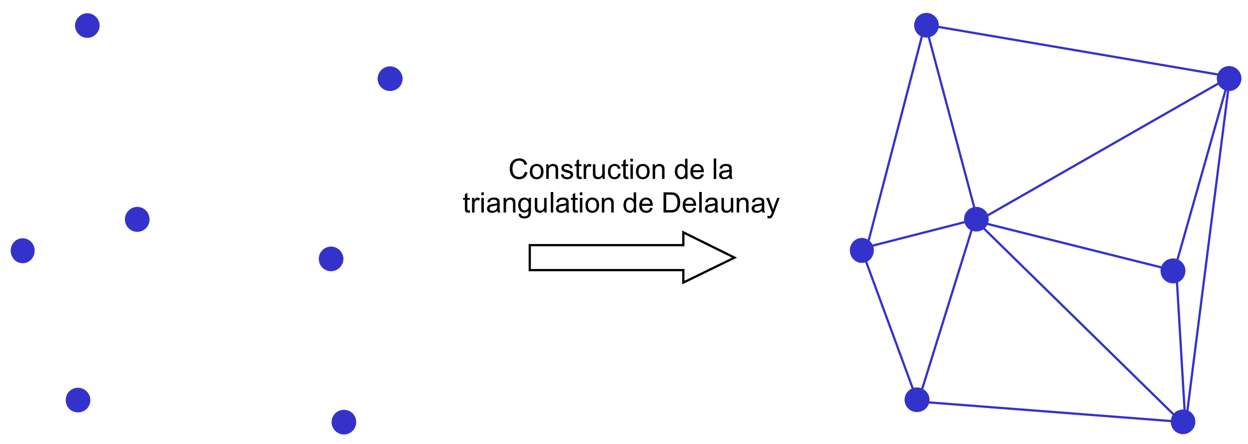
Méthodes applicables aux modèles continus

- Diagramme de Voronoï et triangulation de Delaunay : Extensions
  - Ces arbres de recouvrement ne sont pas minimaux car si on supprime une arête on peut la remplacer par une autre de longueur inférieure.



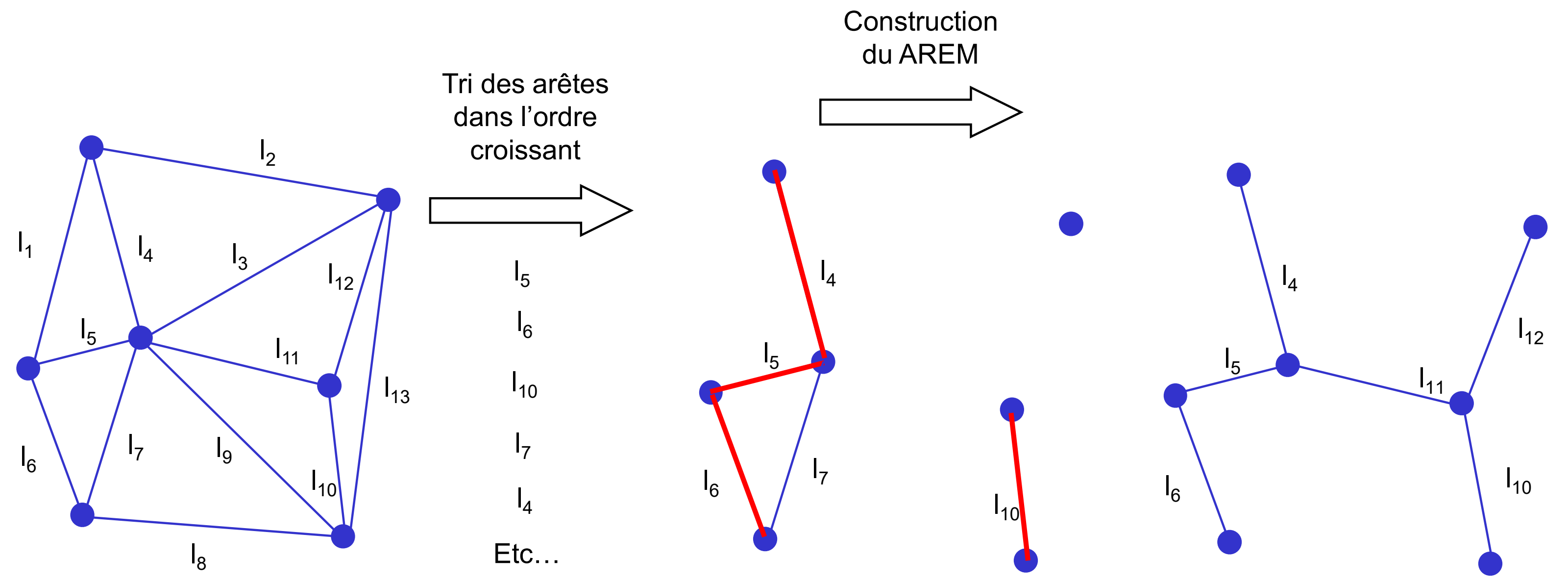
Méthodes applicables aux modèles continus

- Diagramme de Voronoï et triangulation de Delaunay : Extensions
  - Calcul de l’arbre de recouvrement minimum : La somme des longueurs d’arêtes est minimale comparée à tous les arbres de recouvrement.



Méthodes applicables aux modèles continus

- Diagramme de Voronoï et triangulation de Delaunay : Extensions
  - Exemple d'algorithme de construction



Méthodes applicables aux modèles continus

- Diagramme de Voronoï et triangulation de Delaunay : Extensions

  - Les applications des AREMs sont

    - La résolution des problèmes liés au positionnement et à la connectivité d'éléments :
      - Circuits électriques
      - Réseaux téléphoniques

    - D'autres applications comme celle du voyageur de commerce

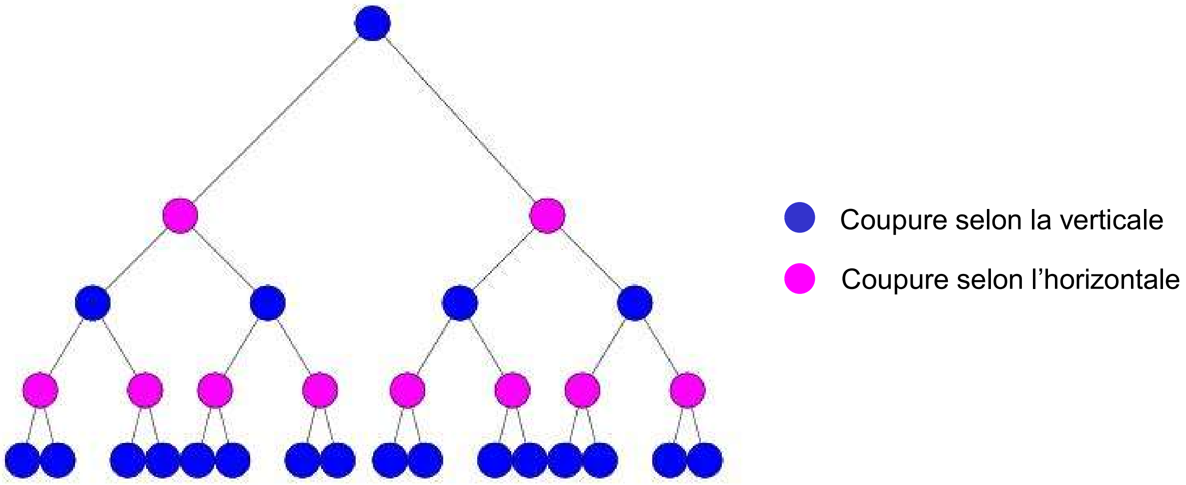
  - Dans certains cas, la distance peut être remplacée par d'autres grandeurs (coût, vitesse, etc…)

Méthodes applicables aux modèles continus

- Partitionnement
  - Les méthodes de partitionnement consistent à diviser un espace à 1, 2, 3 ou k dimensions afin de pouvoir effectuer des opérations « facilement »
  - Des exemples typiques sont la localisation d'objets (points, lignes, etc…), la gestion des intersections entre différents objets (lignes, solides 3D, etc…)
  - Plusieurs types de structures peuvent être utilisées en fonction des données et des problèmes à résoudre

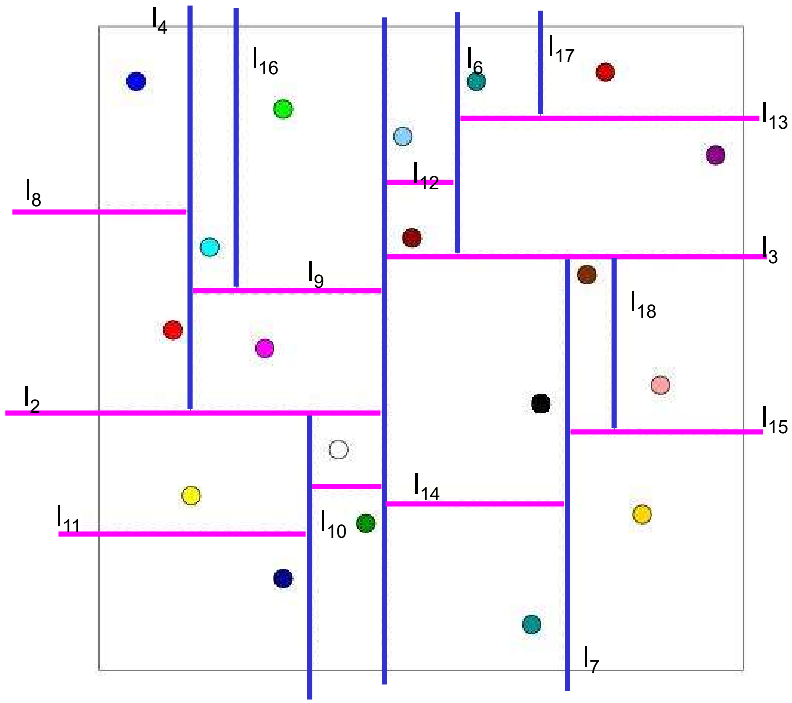
Méthodes applicables aux modèles continus

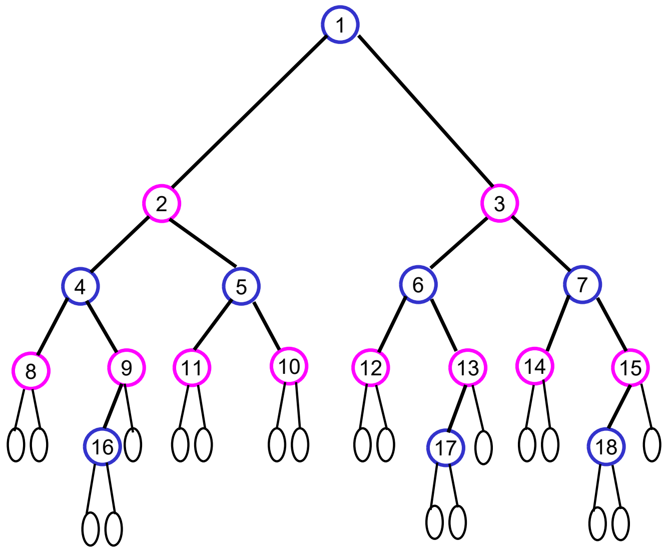
- Partitionnement : k-d arborescences
  - Soit E un ensemble de n points $p_1, \ldots, p_n$ de R^2.
  - On coupe cet ensemble récursivement en deux parties de cardinalité équivalente selon une direction (verticale ou horizontale)



Méthodes applicables aux modèles continus

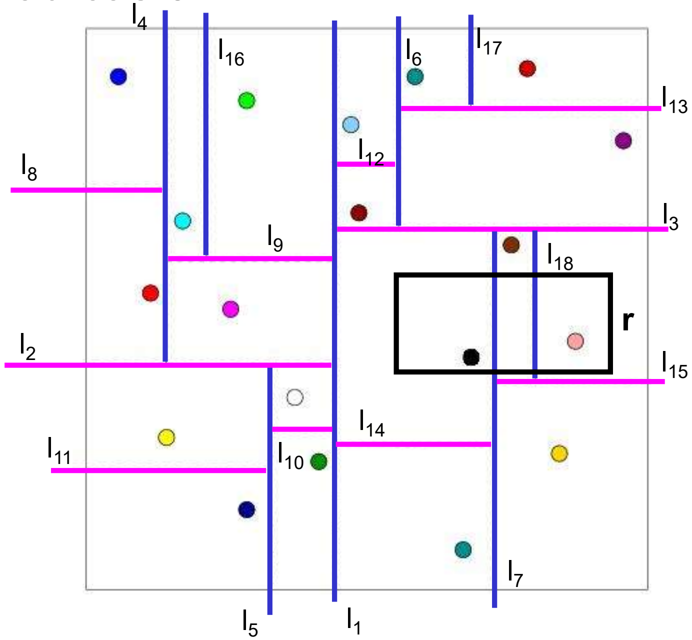
- Partitionnement : k-d arborescences
  - Une k-d arborescence peut être construite en O(n log(n))

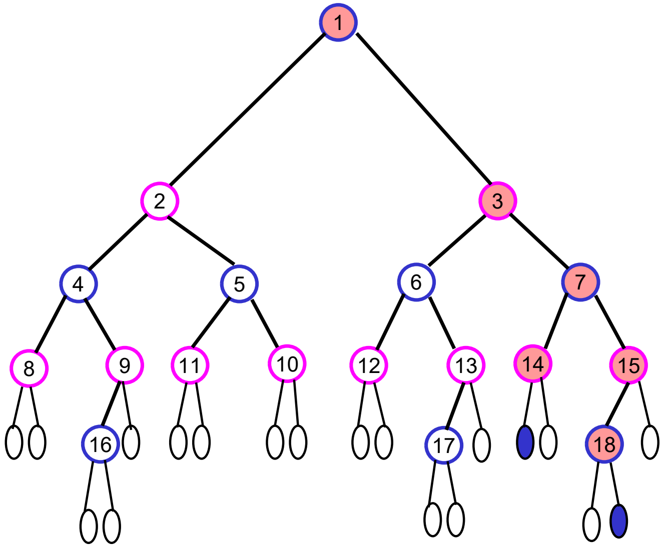




Méthodes applicables aux modèles continus

- Partitionnement : k-d arborescences
  - Permet de répondre à la question : Quels sont les points contenus dans le rectangle r ?
  - Le temps de recherche est de $\sqrt{n}+k$
  - Extension au cas 3D



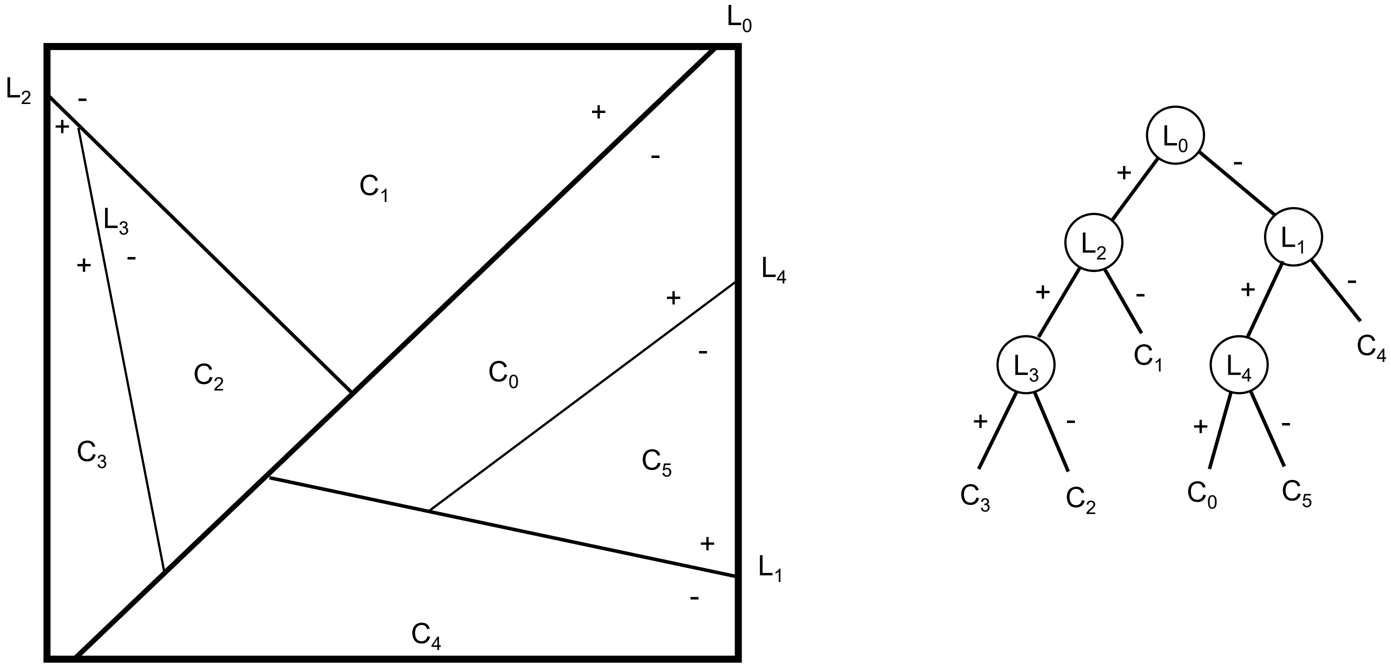


Méthodes applicables aux modèles continus

- Partitionnement : arbres BSP
  - Ces arbres sont des partitions binaires de l’espace (Binary Space Partitions)
  - Soit une droite d, elle divise le plan en deux demi-plans.
  - Il est facile de savoir si un point est à gauche ou à droite de la droite
  - Chaque demi-plan peut être à son tour divisé en deux par une nouvelle droite.

Méthodes applicables aux modèles continus

- Partitionnement : arbres BSP



Méthodes applicables aux modèles continus

- Partitionnement : arbres BSP
  - Les arbres BSP divisent l’espace en régions convexes (intersection de demi-plans)
  - Chaque feuille de l’arbre est une partition convexe
  - Les directions des droites peuvent être quelconques
  - Les arbres BSP peuvent aussi être utilisés en 3D

Méthodes applicables aux modèles continus

- Partitionnement : arbres BSP
  - Quelques applications
    - Optimisation des calculs d'intersection pour le lancer de rayon
      - La scène est partitionnée par un arbre BSP 3D (en utilisant les boites englobantes des objets
      - Chaque feuille contient la liste des objets
      - On ne calcule les intersections que sur les feuilles traversées par les rayons
    - Calcul d'intersections entre polyèdres ou polygones
      - Pour calculer une intersection de deux polygones on fusionne les arbres BSP en mettant à jour les droites de séparation en fonction des intersections trouvées entre ces dernières