Chapitre 7 : Applications, Géométrie algorithmique

Juan Carlos QUEZADA

INSA Strasbourg

Recherche de l'enveloppe convexe

Attention

L'exercice suivant est à rendre sur Moodle pour la semaine prochaine

Introduction

- L'enveloppe convexe d'un ensemble Q de points est le plus petit polygone convexe P tel que chaque point de Q est soit sur le contour de P, soit à l'intérieur. L'enveloppe convexe de Q est notée EC(Q).
- Intuitivement, on peut se représenter les points de Q comme autant d'épingles plantées sur un tableau.
- L'enveloppe convexe est alors le contour formé par un élastique serré qui entoure toutes les épingles.

Recherche de l'enveloppe convexe

La figure 1 montre un ensemble de points et son enveloppe convexe.

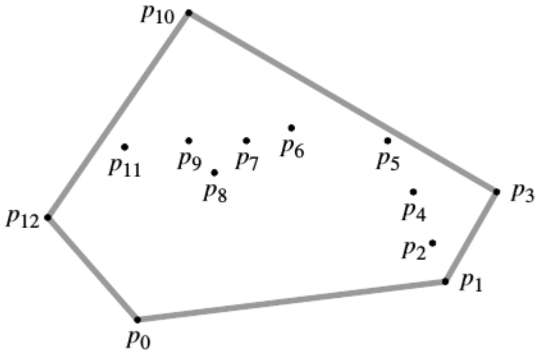


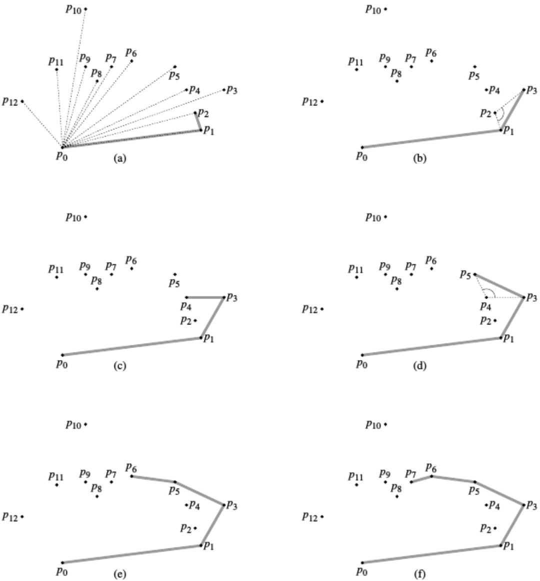
Figure – Un ensemble de points Q avec son enveloppe convexe EC(Q) en gris.

Recherche de l'enveloppe convexe

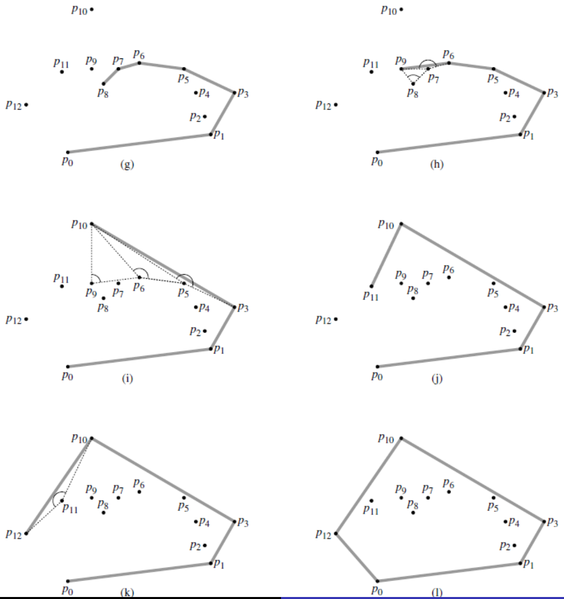
Introduction

- Il existe en fait plusieurs moyens de calculer les enveloppes convexes. Le *balayage de Graham* utilise une technique appelée « balayage circulaire », qui traite les sommets dans l'ordre des angles polaires qu'ils forment avec un sommet de référence.
- Cet algorithme affiche les sommets de l'enveloppe convexe dans l'ordre contraire des aiguilles d'une montre. Le balayage de Graham résout le problème de l'enveloppe convexe en gérant un ensemble S de points candidats. Chaque point de l'ensemble Q est gardé une fois, puis les points qui ne sont pas des sommets de EC(Q) finissent par être tous effacés. Quand l'algorithme se termine, l'ensemble S contient exactement les sommets de EC(Q), dans l'ordre trigonométrique de leur apparition sur le contour.

Recherche de l'enveloppe convexe



Recherche de l'enveloppe convexe



(g) (h) (i) (j) (k) (l)

Recherche de l'enveloppe convexe

On peut décrire les étapes de cet algorithme de la façon suivante :

1. Détermination du point initial p_0 qui est le point ayant l'ordonnée minimale.
2. Tri des points par rapport aux angles polaires θ formés entre p_0 et chaque point par rapport à l'horizontale (dans l'ordre inverse des aiguilles d'une montre). Si on décrit le vecteur allant de p_0 à chaque point p_i par $\overrightarrow{p_0 p_i}$ alors l'angle entre ce vecteur et l'horizontal peut très obtenu par :

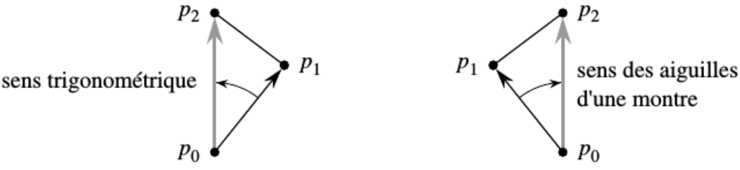
$$\theta = \arccos\left(\frac{p_0 p_i(x)}{\|\overrightarrow{p_0 p_i}\|}\right) = \arccos\left(\frac{x_i - x_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}}\right) \quad (1)$$

3. On fait une sélection initiale avec l'ensemble de points de p_0 à p_{12}. On recopie le point p_0 à la fin de la sélection.
4. Pour i de 1 à $N-1$, **si** l'angle formé entre le point p_i et les points p_{i-1} et p_{i+1} suivantes tourne à droite (dans le sens horaire), on enlève le point p_i (point intermédiaire) et on revient au point précédent pour refaire l'analyse ; **sinon** on avance au point suivant (cf. Fig. 3).

Recherche de l'enveloppe convexe

Pour déterminer si trois points consécutifs forment un angle que tourne à gauche ou à droite, on peut calculer le produit en croix de ces trois points :

$$(p_1 - p_0) \times (p_2 - p_0) = (x_1 - x_0)(y_2 - y_0) - (x_2 - x_0)(y_1 - y_0) \qquad (2)$$



(a) (b)

Recherche de l'enveloppe convexe

- Écrire le programme qui permet de trouver l'enveloppe convexe avec l'implémentation du Balayage de Graham pour une série de 12 points générés aléatoirement avec des coordonnées x,y allant de 0 à 100.
- Les coordonnées de points qui conforment l'enveloppe convexe pourront être stockés sous forme d'un tableau dynamique avec la commande *ArrayList* :

```
//ajouter dans la première ligne de le script
import java.util.*;

//génération tableau dynamique
ArrayList<Integer> CoordonneesX = new ArrayList<Integer>();

CoordonneesX.add(50); //ajoute d'un élément

CoordonneesX.get(0); //acceder au contenu de la première
                     //case du tableau

CoordonneesX.remove(0); //enlève l'élément de la position 0
```

Optionnel : Affichage du résultat

- Le fichier "PlotEnveloppe.java" (voir sur Moodle), permet de tracer les points générés aléatoirement et l'enveloppe convexe de ces points.
- Afin d'utiliser ce fichier, il faudra renommer votre script "ConvexHull.java", et le découper dans les sous-programmes suivants :

```
public static int[][] pointsinitiaux()
// Fonction qui renvoie un tableau avec les coordonnées de
// 12 points aleatoires

public static int[][] enveloppe(int [][] points)
// Fonction qui renvoie un tableau avec les coordonnées des
// points qui forment l'enveloppe convexe
```