

Introduction à l'algorithmique : géométrie du plan

Serge Haddad

LMF, ENS Paris-Saclay & CNRS

L3 et M2 FESUP

Plan

① Tests et opérations géométriques

Algorithme de Graham

Algorithme dichotomique

Algorithme de Jarvis

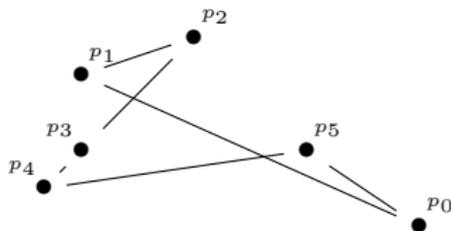
Algorithme de Chan

Polygones

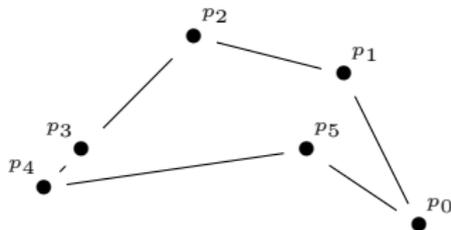
Un *polygone* \mathcal{P} est donné par une liste p_0, \dots, p_{n-1} de ses sommets dont les arêtes sont $(p_i p_{i+1 \% n})_{0 \leq i < n}$.

Par commodité, p_0 est le point le plus bas et en cas d'ex aequo le point le plus à gauche qu'on appelle *point extrême*.

Tous les points appartiennent donc au demi-plan horizontal supérieur dont l'axe passe par p_0 .

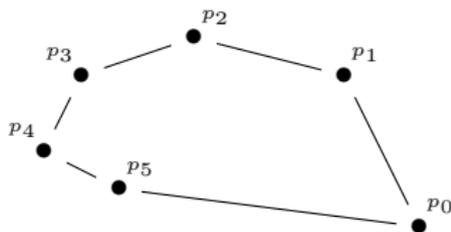


Un polygone est *simple* si les arêtes ne se croisent pas.



Un polygone simple est *convexe* si tous les angles $(p_{i+1 \% n} p_i p_{i-1 \% n})_{0 \leq i < n}$ sont *saillants* i.e., entre 0° et 180° exclus.

Cela implique que les points sont énumérés dans le sens direct.

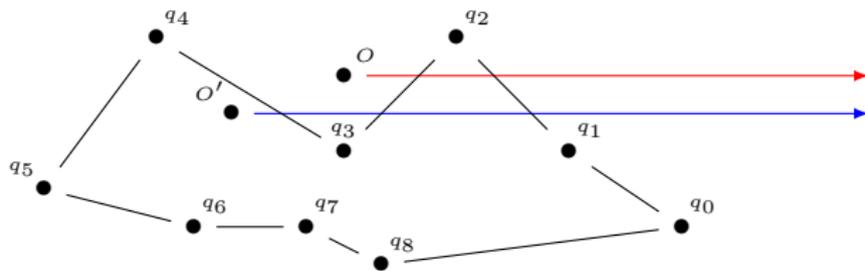


Appartenance à un polygone simple

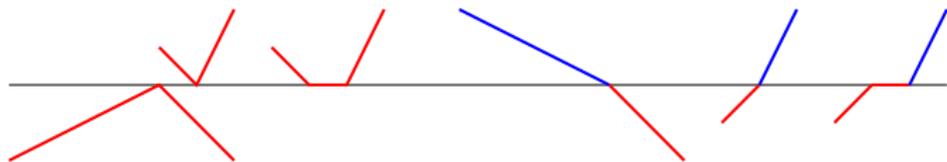
Soit O un point. On considère la demi-droite horizontale \overrightarrow{Ox} .

On compte le nombre d'intersections avec les segments du polygone.

O appartient au polygone si ce nombre est impair.

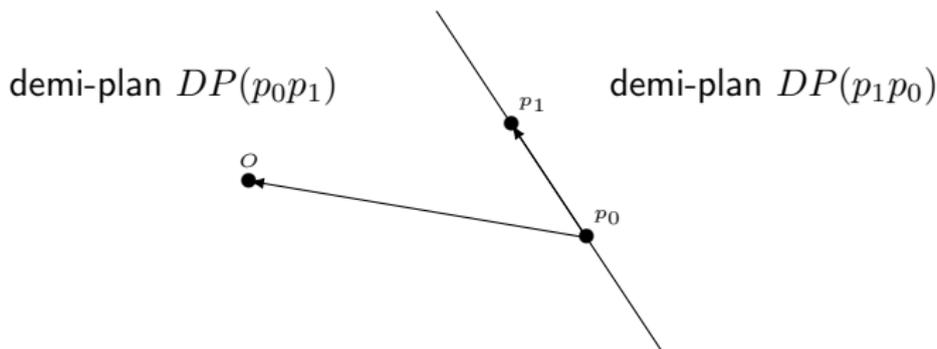


Cas dégénérés.



Appartenance à un polygone convexe

Un point o appartient au demi-plan $DP(p_0p_1)$
si et seulement si $\det(\overrightarrow{p_0p_1}, \overrightarrow{p_0o}) \geq 0$.



Un polygone convexe $\mathcal{P} = (p_i)_{0 \leq i < n}$ est l'intersection
des demi-plans $DP(p_i p_{i+1 \% n})$.

Un point o appartient à \mathcal{P} si et seulement si pour tout $0 \leq i < n$,

$$\det(\overrightarrow{p_i p_{i+1 \% n}}, \overrightarrow{p_i o}) \geq 0$$

D'où un test en $O(n)$.

Ordre polaire

Soit deux vecteurs non nuls en coordonnées polaire $\mathbf{v} = (\rho, \theta)$ et $\mathbf{v}' = (\rho', \theta')$.

$$\mathbf{v} \prec \mathbf{v}' \text{ ssi } \theta < \theta' \text{ ou } (\theta = \theta' \text{ et } \rho > \rho')$$

Prec(\mathbf{v}, \mathbf{w}) (*teste si $\mathbf{v} \preceq \mathbf{w}$, \mathbf{v} et \mathbf{w} non nuls*)

$$d = \det(\mathbf{v}, \mathbf{w})$$

If $d = 0$ **then**

(vecteurs liés)

If $\mathbf{v} \cdot \mathbf{w} \geq 0$ **then return** $\mathbf{v} \cdot \mathbf{v} \geq \mathbf{w} \cdot \mathbf{w}$

(vecteurs de même direction)

Else return $\mathbf{v}[y] > 0$ **or** ($\mathbf{v}[y] = 0$ **and** $\mathbf{v}[x] > 0$)

(vecteurs de direction opposée)

(utilisation de la symétrie par rapport à l'axe horizontal)

If $\mathbf{v}[y] < 0$ **then return not** **Prec**(($-\mathbf{v}[x], \mathbf{v}[y]$), ($-\mathbf{w}[x], \mathbf{w}[y]$))

(\mathbf{v} est dans demi-plan supérieur et \mathbf{w} est dans demi-plan inférieur)

If $\mathbf{w}[y] < 0$ **then return true**

(\mathbf{v} et \mathbf{w} sont dans le demi-plan supérieur : le déterminant fournit la réponse)

return $d > 0$

Appartenance à un polygone convexe

Soit o un point.

L'appartenance de o à un triangle se fait en $O(1)$ par le calcul de 3 déterminants.

Soit $(p_i)_{0 \leq i < n}$ un polygone convexe.

```
If  $o = p_0$  then return true  
 $d \leftarrow 1; f \leftarrow n - 1$   
While  $f \geq d + 2$  do  
   $mid \leftarrow \lfloor \frac{d+f}{2} \rfloor$   
  If  $\overrightarrow{p_0 o} \prec \overrightarrow{p_0 p_{mid}}$  then  $f \leftarrow mid$  else  $d \leftarrow mid$   
return  $o \in Triangle(p_0, p_d, p_f)$ 
```

D'où un test en $O(\log(n))$.

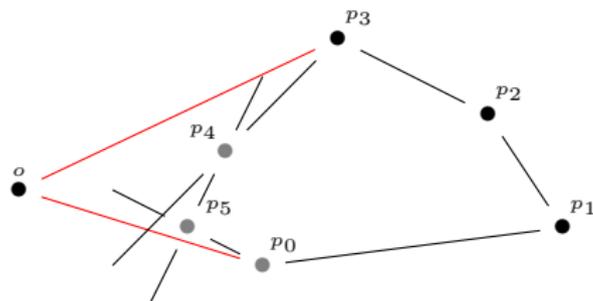
Ajout d'un sommet

Soit $\mathcal{P} = (p_i)_{0 \leq i < n}$ un polygone convexe.

Pour $0 \leq d, f < n$, on note $I[d, f] = \{d, d + 1 \% n, \dots, f\}$.

Lorsque $o \notin \mathcal{P}$, il existe deux indices d, f possiblement égaux tels que pour tout $0 \leq i < n$, $o \notin DP(p_{i-1 \% n}, p_i)$ ssi $i \in I[d, f]$.

Illustration. $d = 4$ et $f = 0$.



Ajout en $O(n)$.

- Si $d \neq f$, on supprime les points p_i tels que $i \in I[d, f - 1 \% n]$.
- On insère o avant f .

Enveloppe convexe

Soit S un ensemble de n points p_0, \dots, p_{n-1} dans le plan.

$EC(S)$, l'enveloppe convexe de S , est définie par :

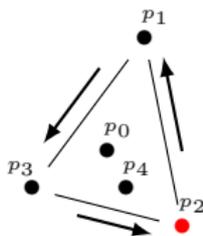
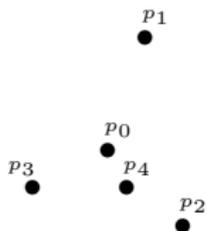
$$EC(S) = \{p \mid \exists \{\lambda_i\}_{i < n} \in (\mathbb{R}^+)^n \wedge \sum_{i < n} \lambda_i = 1 \wedge p = \sum_{i < n} \lambda_i p_i\}$$

$EC(S)$ est l'ensemble des points contenus

dans un polygone convexe $\mathcal{P}(S)$ (en incluant les cas segment et point).

L'ensemble des sommets de $\mathcal{P}(S)$ est un sous-ensemble de S .

Illustration.



Objectif. Construire la représentation *canonique* de $\mathcal{P}(S)$ i.e.,
l'énumération des points du polygone en sens direct à partir du point extrême.

Plan

Tests et opérations géométriques

② Algorithme de Graham

Algorithme dichotomique

Algorithme de Jarvis

Algorithme de Chan

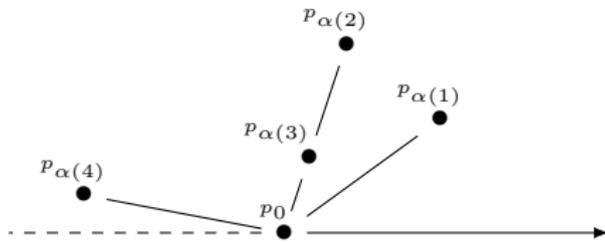
Circuit polaire

Soit $S = \{p_0, \dots, p_n\}$ avec p_0 le point extrême de S .

Soit α une permutation de $\{1, \dots, n\}$.

$(p_0, p_{\alpha(1)}, \dots, p_{\alpha(n)})$ est un circuit polaire si pour tout $0 < i < n$,

$$\overrightarrow{p_0 p_{\alpha(i)}} \prec \overrightarrow{p_0 p_{\alpha(i+1)}}$$



Observation. Tous les p_i se situent dans le demi-plan supérieur dont l'axe horizontal passe par p_0 .

Le circuit polaire s'obtient par un tri en $O(n \log(n))$.

Du circuit polaire à l'enveloppe convexe

Soit \mathcal{C} un circuit polaire de n points avec p_0 le point extrême.

```
 $p \leftarrow p_0; \mathcal{C}' \leftarrow \mathcal{C}$   
While  $\text{succ}(\mathcal{C}', p) \neq p_0$  do  
  If  $\det(\overrightarrow{\text{succ}(\mathcal{C}', p)p}, \overrightarrow{\text{succ}(\mathcal{C}', p)\text{succ}(\mathcal{C}', \text{succ}(\mathcal{C}', p))}) \geq 0$  then  
     $\text{Extraire}(\mathcal{C}', \text{succ}(p));$  If  $p \neq p_0$  then  $p \leftarrow \text{pred}(\mathcal{C}', p)$   
  Else  
     $p \leftarrow \text{succ}(\mathcal{C}', p);$ 
```

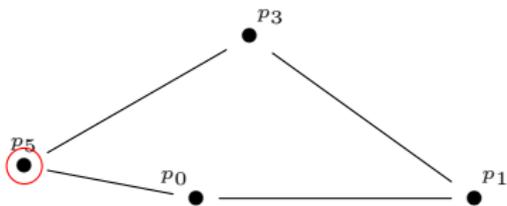
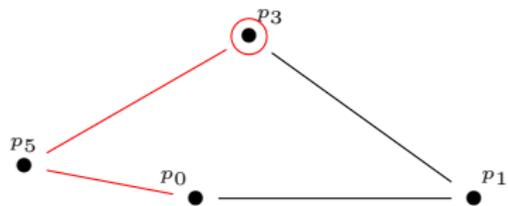
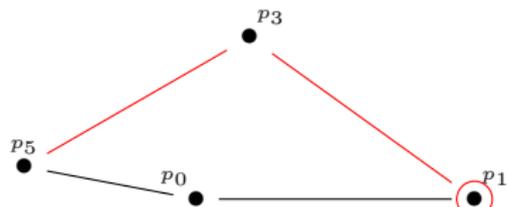
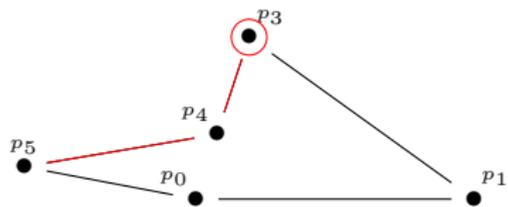
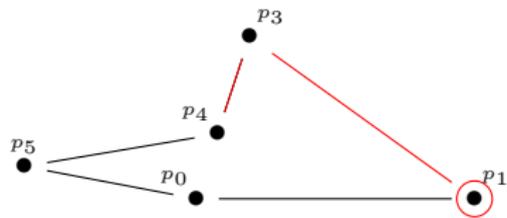
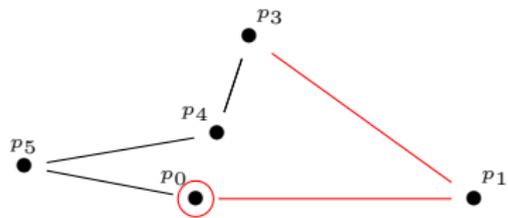
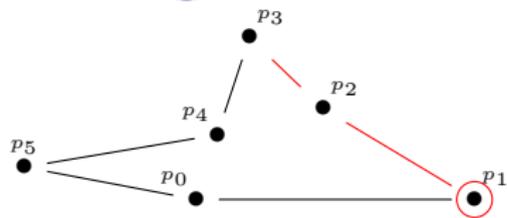
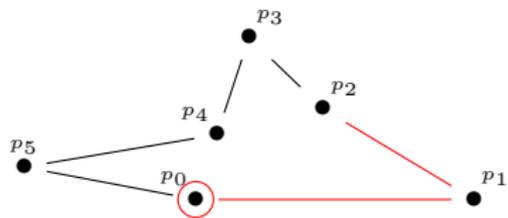
Complexité.

Soit $\ell = |\mathcal{C}'| + \min(i > 0 \mid \text{succ}^{(i)}(\mathcal{C}', p) = p_0)$.

ℓ est initialement égal à $2n$ et décroît à chaque étape.

D'où un algorithme en $O(n)$.

Déroulement de l'algorithme



Correction de l'algorithme

Invariants de la boucle.

- ▶ L'enveloppe convexe de $EC(C)$ reste égale à $EC(C')$.
- ▶ Les angles formés par les segments de p_0 à $\text{succ}(C', p)$ sont *rentrants* i.e, $180^\circ < \theta < 360^\circ$.

Extraction d'un sommet.

- ▶ Un sommet $\text{succ}(C', p)$ retiré appartient au « triangle » $\langle p_0, p, \text{succ}(C', \text{succ}(C', p)) \rangle$.
- ▶ Puisque p devient $\text{pred}(C', p)$, la deuxième propriété reste vérifiée.

Passage au successeur.

- ▶ C' est inchangé.
- ▶ La deuxième propriété reste vérifiée par le test dans la boucle.

Observation.

A la fin de l'algorithme l'angle en p_0 est nécessairement rentrant (choix de p_0).

Rappel : un polygone simple dont tous les angles sont rentrants est convexe.

Plan

Tests et opérations géométriques

Algorithme de Graham

3 Algorithme dichotomique

Algorithme de Jarvis

Algorithme de Chan

Diviser pour régner : le retour

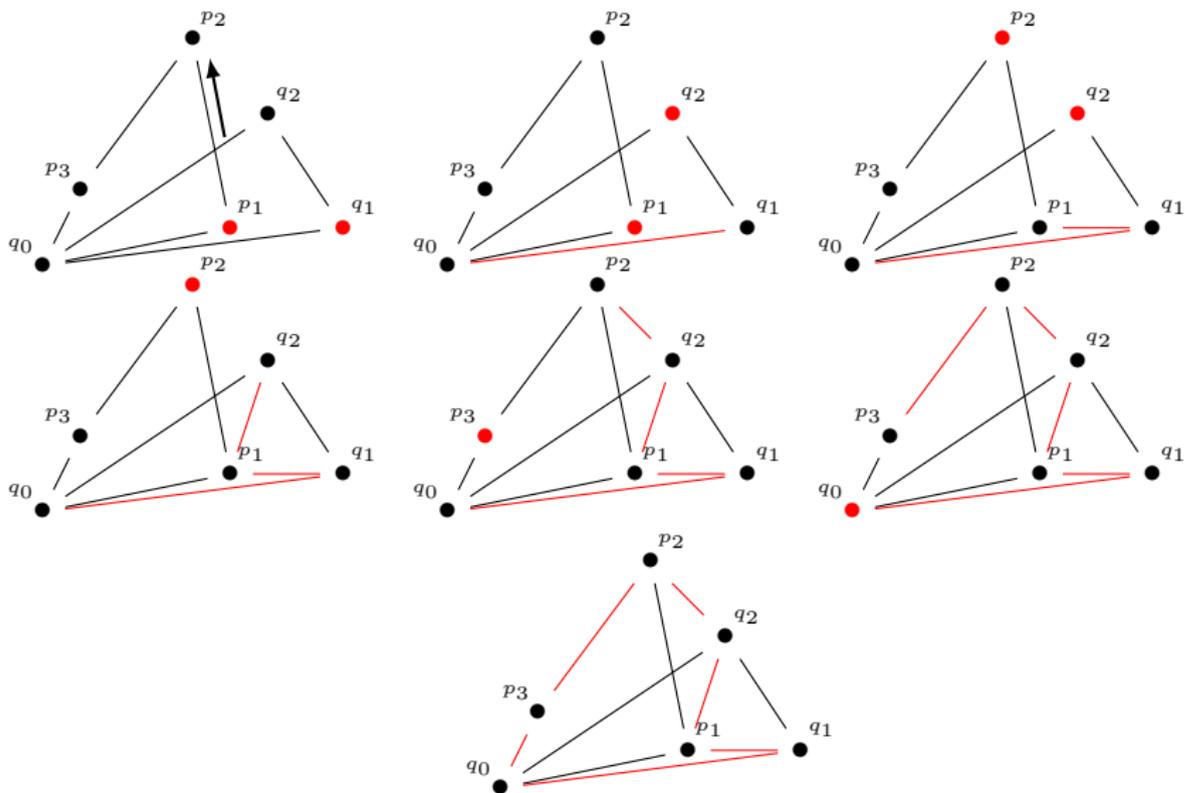
Principe.

- ▶ Si $n \leq 3$ alors éliminer les points redondants en $O(1)$.
- ▶ Répartir les points en 2 sous-ensembles de tailles approximativement égales.
- ▶ Calculer de manière récursive les polygones associés aux sous-ensembles : $\mathcal{P} = (p_0, \dots, p_{r-1})$ et $\mathcal{Q} = (q_0, \dots, q_{s-1})$
- ▶ Fusionner les polygones en $O(r + s)$.

Fusion.

- ▶ Déterminer le point extrême r : p_0 ou q_0 en $O(1)$.
- ▶ Si $r = p_0$ alors ajouter p_0 à \mathcal{Q} en $O(s)$ sinon ajouter q_0 à \mathcal{P} en $O(r)$.
On suppose donc que $q_0 = p_0$.
- ▶ Fabriquer par fusion un circuit polaire à partir des listes p_1, \dots, p_{r-1} et q_1, \dots, q_{s-1}
- ▶ Transformer le circuit polaire en polygone par la procédure de Graham.

Des polygones au circuit polaire



Plan

Tests et opérations géométriques

Algorithme de Graham

Algorithme dichotomique

4 Algorithme de Jarvis

Algorithme de Chan

Observations

Soit $\mathcal{S} = \{p_0, \dots, p_{n-1}\}$ avec p_0 le point extrême de \mathcal{S} .

Soit $\mathcal{P} = (p_0, p_{\alpha(1)}, \dots, p_{\alpha(k-1)})$ le polygone de $EC(\mathcal{S})$.

On étend α par $\alpha(0) = 0$.

Soit θ_i , l'angle du vecteur $\overrightarrow{p_{\alpha(i-1 \% k)} p_{\alpha(i)}}$.

Pour $j \neq \alpha(i)$, soit \mathbf{w}_{ij} le vecteur rotate($\overrightarrow{p_{\alpha(i)} p_j}$, $-\theta_i$).

Alors pour l'ordre polaire :

$$\alpha(1) = \arg \min(\overrightarrow{p_0 p_j} \mid j \neq 0)$$

$$\text{pour } i \geq 1, \alpha(i + 1 \% n) = \arg \min(\mathbf{w}_{ij} \mid j \neq \alpha(i))$$

Algorithmme de Jarvis

$\mathcal{P} \leftarrow \emptyset$; Ajouter(\mathcal{P}, p_0)

$p \leftarrow p_1$

For i **from** 2 **to** $n - 1$ **do**

If $\overrightarrow{p_0 p_i} \prec \overrightarrow{p_0 p}$ **then** $p \leftarrow p_i$

Ajouter(\mathcal{P}, p)

$q \leftarrow p_0$

Loop

$\theta = \text{angle}(\overrightarrow{qp})$

$o \leftarrow p_0$

For i **from** 1 **to** $n - 1$ **do**

If $p_i \neq p$ **and** $\text{rotate}(\overrightarrow{pp_i}, -\theta) \prec \text{rotate}(\overrightarrow{po}, -\theta)$ **then** $o \leftarrow p_i$

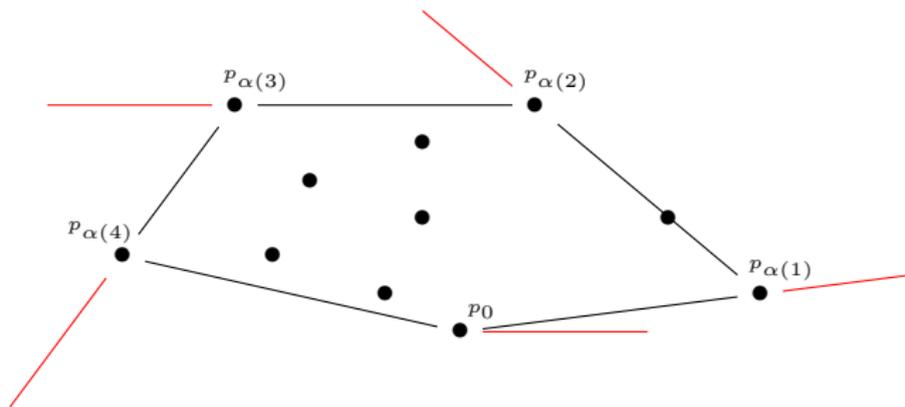
If $o = p_0$ **then return**(\mathcal{P})

 Ajouter(\mathcal{P}, o)

$q \leftarrow p$; $p \leftarrow o$

End Loop

Illustration et complexité



Complexité.

La complexité de l'algorithme est en $O(nk)$

où k est le nombre de sommets de l'enveloppe convexe.

Si $k = o(\log(n))$ alors cet algorithme plus efficace que les précédents.

Plan

Tests et opérations géométriques

Algorithme de Graham

Algorithme dichotomique

Algorithme de Jarvis

5 Algorithme de Chan

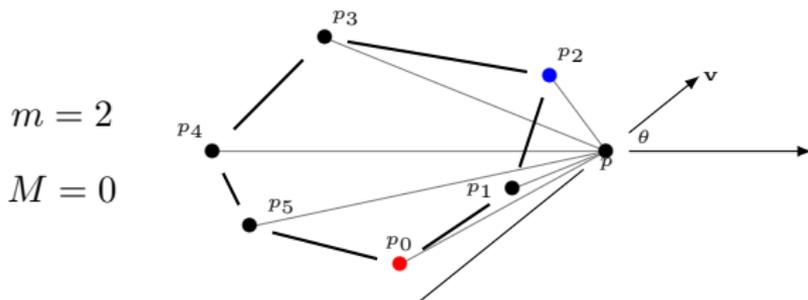
Point minimal d'un polygone convexe

Soit p un point et \mathbf{v} un vecteur non nul d'angle θ .

Soit un polygone convexe $\mathcal{P} = (p_0, \dots, p_{n-1})$ contenu dans :

- ▶ le demi-plan $DP(p, \mathbf{v})$
- ▶ privé de la demi-droite partant de p et d'orientation \mathbf{v} .

Comme pour Jarvis, on cherche $\arg \min(\text{rotate}(\overrightarrow{pp_i}), -\theta) \mid 0 \leq i < n$.



Soit p_m le point minimal et p_M le point maximal. Alors :

- ▶ pour tout $i \in I[m, M - 1\%n]$, $\text{rotate}(\overrightarrow{pp_i}, -\theta) \prec \text{rotate}(\overrightarrow{pp_{i+1\%n}}, -\theta)$;
- ▶ pour tout $i \in I[M, m - 1\%n]$, $\text{rotate}(\overrightarrow{pp_{i+1\%n}}, -\theta) \prec \text{rotate}(\overrightarrow{pp_i}, -\theta)$.

Observation

Soit $(\mathbf{v}_i)_{0 \leq i < n}$ une suite de vecteurs et $0 \leq m, M < n$ qui vérifient :

- ▶ pour tout $i \in I[m, M - 1\%n]$, $\mathbf{v}_i \prec \mathbf{v}_{i+1\%n}$;
- ▶ pour tout $i \in I[M, m - 1\%n]$, $\mathbf{v}_{i+1\%n} \prec \mathbf{v}_i$.

Supposons que $m \in I[d_1, f_1] \uplus I[d_2, f_2] \uplus I[d_3, f_3]$ avec :

$I[d_2, f_2] \subseteq I[f_1 + 1\%n, d_3 - 1\%n]$, $\mathbf{v}_{d_1} \prec \mathbf{v}_{d_2}$ et $\mathbf{v}_{d_1} \prec \mathbf{v}_{d_3}$.

Alors $m \notin I[d_2, f_2]$.

Preuve par l'absurde. Supposons $m \in I[d_2, f_2]$.

- ▶ Puisque $\mathbf{v}_{d_1} \prec \mathbf{v}_{d_2}$ et $\mathbf{v}_m \prec \mathbf{v}_{d_2}$, $M \in I[d_1 + 1\%n, m - 1\%n]$
- ▶ Puisque $\mathbf{v}_{d_1} \prec \mathbf{v}_{d_3}$ et $\mathbf{v}_m \prec \mathbf{v}_{d_3}$, $M \in I[m + 1\%n, d_1 - 1\%n]$

Or $I[d_1 + 1\%n, m - 1\%n] \cap I[m + 1\%n, d_1 - 1\%n] = \emptyset$

ce qui est contradictoire.



Recherche du point minimal

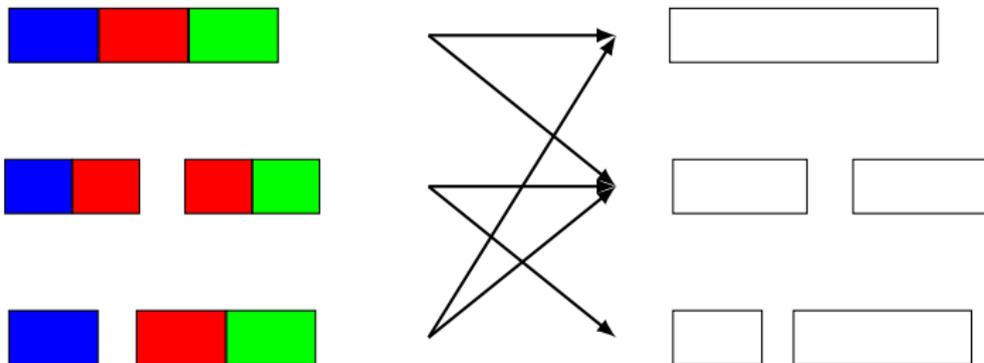
Principe.

On démarre avec l'intervalle $I[0, n - 1]$.

On maintient un ou deux intervalles où se trouve m .

On découpe cet (resp. ces) intervalle(s) en trois parties et par le test précédent, on élimine une des parties.

Découpage d'un ou deux intervalles en trois et élimination d'une partie.



Réduction

- ▶ d'au moins d'un facteur $\frac{3}{4}$ à chaque étape ;
- ▶ d'au moins d'un facteur $\frac{1}{2}$ pour deux étapes.

Recherche du point minimal en $O(\log(n))$

```
MinPoint( $\mathcal{P} = (p_i)_{0 \leq i < n}, p, \theta$ )
  For  $i \in \{0, n - 1\}$  do  $\mathbf{v}_i \leftarrow \text{rotate}(\overrightarrow{pp_i}, -\theta)$ 
   $Int \leftarrow [0, n - 1]$ 
  While  $\text{nbpoints}(Int) > 3$  do
    //  $Int$  est un intervalle ou l'union de deux intervalles contenant le minimum
     $i \leftarrow \text{debfirstpart}(Int)$ 
     $j \leftarrow \text{debsecondpart}(Int)$ 
     $k \leftarrow \text{debthirdpart}(Int)$ 
     $\text{min} \leftarrow \arg \min(\mathbf{v}_x \mid x \in \{i, j, k\})$ 
    If  $\text{min} = i$  then  $Int \leftarrow Int \setminus \text{secondpart}(Int)$ 
    If  $\text{min} = j$  then  $Int \leftarrow Int \setminus \text{thirdpart}(Int)$ 
    If  $\text{min} = k$  then  $Int \leftarrow Int \setminus \text{firstpart}(Int)$ 
  return  $\arg \min(\mathbf{v}_x \mid x \in Int)$ 
```

Un algorithme incomplet à la Jarvis

Première étape.

On choisit un paramètre $m \leq n$ supposé être une borne supérieure du nombre de points de l'enveloppe convexe.

On partitionne \mathcal{S} en $\lceil n/m \rceil$ sous-ensembles \mathcal{S}_i d'au plus m points.

On calcule $(\mathcal{P}_i)_{i \leq \lceil n/m \rceil} = (\mathcal{P}(\mathcal{S}_i))_{i \leq \lceil n/m \rceil}$ les polygones des enveloppes convexes des \mathcal{S}_i par Graham en $O(\lceil n/m \rceil m \log(m)) = O(n \log(m))$.

Deuxième étape.

On applique une variante de l'algorithme de Jarvis qui :

- ▶ échoue si l'enveloppe convexe contient plus de m points ;
- ▶ recherche le nouveau point à insérer comme « minimum » des points minimaux des \mathcal{P}_i ;
- ▶ et par conséquent opère en $O(m \lceil n/m \rceil m \log(m)) = O(n \log(m))$.
car m est une borne du nombre de points insérés
et $\lceil n/m \rceil$ est le nombre de recherches en $O(\log(m))$ pour chaque insertion.

L'algorithme de la deuxième étape

```
 $\mathcal{P} \leftarrow \emptyset$ ; Ajouter( $\mathcal{P}, p_0$ )  
For  $i$  from 1 to  $\lceil n/m \rceil$  do  
   $s \leftarrow \text{Minpoint}(\mathcal{P}_i \setminus \{p_0\}, p_0, 0)$   
  If  $i = 1$  or  $\overrightarrow{p_0 s} \prec \overrightarrow{p_0 p}$  then  $p \leftarrow s$   
Ajouter( $\mathcal{P}, p$ )  
 $q \leftarrow p_0$   
For  $k$  from 1 to  $m$  do  
   $\theta \leftarrow \text{angle}(\overrightarrow{qp})$   
  For  $i$  from 1 to  $\lceil n/m \rceil$  do  
     $s \leftarrow \text{Minpoint}(\mathcal{P}_i \setminus \{p\}, p, \theta)$   
    If  $i = 1$  or  $\text{rotate}(\overrightarrow{ps}, -\theta) \prec \text{rotate}(\overrightarrow{po}, -\theta)$  then  $o \leftarrow s$   
  If  $o = p_0$  then return  $(\top, \mathcal{P})$   
Ajouter( $\mathcal{P}, o$ )  
 $q \leftarrow p$ ;  $p \leftarrow o$   
return  $(\perp, -)$ 
```

L'algorithme de Chan

L'algorithme de Chan essaye des valeurs de m très rapidement croissantes.

```
For  $i$  from 1 to  $\lceil \log(\log(n)) \rceil$  do  
   $m \leftarrow \min(2^{2^i}, n)$   
   $(b, \mathcal{P}) \leftarrow \text{IncJarvis}(\mathcal{S}, m)$   
  If  $b = \top$  then Return( $\mathcal{P}$ )
```

Analyse de complexité

Soit k le nombre de points de l'enveloppe convexe.

L'algorithme s'arrête après $\lceil \log(\log(h)) \rceil$ itérations.

La i ème itération s'exécute en $O(n2^i)$.

L'algorithme s'exécute donc en $O(n2^{\lceil \log(\log(h)) \rceil + 1}) = O(n \log(k))$.