« Algorithmique Avancée »

CHAPITRE 4 : Algorithmique Géométrique

Enveloppe Convexe

- Ordre polaire
- Enveloppe convexe
- Algorithme de Graham
- Algorithme de Jarvis
- Enveloppe Convexe Dynamique

Présentation

- Des applications à grande échelle : vision par ordinateur, robotique, CAO, synthèse d'image
- Opérations primitives sur points et lignes non cablées (contrairement aux opérations primitives sur les nombres et les textes) et difficiles à implémenter...
- Problèmes géométriques faciles résoudre "de visu" mais algorithmiquement compliqués (ex : point à l'intérieur d'un polygone ?)
- Calculer des objets géométriques : Enveloppe Convexe existe (maths) effective (algos)
- Toujours problèmes des cas dégénérés
  - points alignés (EC)
  - intersections multiples (intérieur)

Les bases

- Repère orthonormé $(O, \vec{i}, \vec{j})$
- Objets de base (en dimension 2)
  - point : coordonnées (cartésiennes, polaires)
  - ligne (segment) = couple de points,
  - polygone = suite de points (contour circulaire positif)
- Ordres sur $\mathcal{P}$
  - *ordre cartésien* : ordre lexicographique en coordonnées cartésiennes
    $M \leq P \Leftrightarrow \quad x_M < x_P, \quad \text{ou} \quad x_M = x_P \text{ et } y_M \leq y_P$
  - *ordre polaire* : ordre lexicographique en coordonnées polaires (angles compris entre 0 et 2π et normes)
    $M \leq_{O,Ox} P \Leftrightarrow \quad \text{angle}\,(Ox, OM) < \text{angle}\,(Ox, OP)$, ou $\text{angle}\,(Ox, OM) = \text{angle}\,(Ox, OP)$ et $|OP| \leq |OM|$
    Ordre total sur $\mathcal{P} - \{\mathcal{O}\}$. Mais pas efficace.

Implantation de l'ordre polaire

Eviter de calculer des mesures d'angles (coûteux et inverser des fonctions trigonométriques introduit erreurs d'arrondis). Utiliser uniquement des opérations arithmétiques pour comparer des points selon l'ordre polaire.

- *idée de base* : $0 < (\vec{u}, \vec{v}) < \pi \quad \Leftrightarrow \quad \det(\vec{u}, \vec{v}) > 0$, $\forall$ repère orthonormé direct.
- *précédence circulaire selon O* :

$$
\begin{aligned}
M \preceq_O P \quad \Leftrightarrow \quad & \det(\vec{OM}, \vec{OP}) > 0 \\
& \det(\vec{OM}, \vec{OP}) = 0 \quad \text{et} \quad \vec{OM} <> \vec{OP} \\
& \det(\vec{OM}, \vec{OP}) = 0, \vec{OM} == \vec{OP}, \quad \text{et} \quad |OP| \leq |OM|
\end{aligned}
$$

Relation ni antisymétrique, ni transitive ! !

Précédence circulaire, ordre polaire

Proposition : $M \leq_{O,Ox} P \Leftrightarrow$ M appartient au secteur angulaire $(\vec{Ox}, \vec{OP})$:

- ou bien $x \preceq_O M$ et $M \preceq_O P$,
- ou bien $x \preceq_O M$ et $P \preceq_O x$,
- ou bien $M \preceq_O P$ et $P \preceq_O x$,

Calcul au plus de 3 déterminants : $O(1)$.

Définition : $(P_1, \ldots, P_n)$ est un *circuit polaire par rapport à O* ssi $\forall i, P_i \in (\vec{OP}_{i-1}, \vec{OP}_{i+1})$
Propriété : $(P_1, \ldots, P_n)$ *est un circuit polaire par rapport à O ssi $\forall P_0$ fixé, il existe un conjugué* $(P_j, P_{j+1}, \ldots, P_n, P_1, \ldots, P_{j-1})$*, t.q.* $\forall k, P_k \leq_{O,OP_0} P_{k+1}$.
Calcul du circuit polaire de N points en $O(N \log N)$ comparaisons (voit TD).

Exemple

Chemin fermé simple : soient N points, trouver chemin C

- qui passe par tous les points,
- ne se recoupe pas,
- revient au point de départ.

Remarque : pbs différents

- *voyageur de commerce* (trouver le plus court chemin)
- *enveloppe convexe* (trouver le plus court chemin qui englobe tous les points).

Principe : calculer l'ordre polaire de chacun des points par rapport à un point distingué S, puis trier selon cet ordre, et joindre les points en ordre.

Enveloppe convexe

Définitions

- *S convexe* ssi $\forall (x, y) \in S, [x, y] \subset S$
- *L'enveloppe convexe* de S est le plus petit ensemble convexe contenant S

Propriétés

1. *$EC(S)$ existe*
2. *Si S est un ensemble fini de points, $EC(S)$ est un polygone convexe dont les sommets sont dans S*

Tout point d'ordonnée minimale et tout point d'abscisse minimale de S appartiennent à l'enveloppe convexe $EC(S)$.

Algorithmes
ENTREE : ensemble S de N points
SORTIE : contour positif du polygone convexe

Enveloppe convexe et Tri

Propriété : *Le calcul de l'enveloppe convexe d'un ensemble de N points demande* $\Omega(N \log N)$ comparaisons.

- N nombres : coordonnées polaires de N points (de même rayon) .
  L'enveloppe convexe contient ces N points, triés en ordre polaire. Donc si on savaiit faire l'EC en moins de *NlogN* comparaisons, on saurait trier en moins de *NlogN* comparaisons.
- N points sur parabole $y = x^2$, dans le contour positif de l'EC les points sont triés par abscisses croissantes.

Méthode de Graham

Propriétés

1. *Si le point O est à l'intérieur d'un polygone convexe P, alors un contour positif de P est un circuit polaire par rapport à O.*
2. *Un polygone simple est convexe ssi pour tout triplet (P, Q, R) de sommets consécutifs en sens positif, $Q \prec_P R$ (PQR tour gauche)*

Déterminer un circuit polaire de tous les points de S
Le parcourir en supprimant tous les points P_i tels que (P_{i-1}, P_i, P_{i+1}) n'est pas un tour gauche.

Algorithme de Graham

- Choisir un point $O \notin S$, intérieur à $EC(S)$. $O(N)$
- Calculer le circuit polaire de S par rapport à O. $O(N \log N)$
- Soit P_0 le point d'abscisse max de S ($P_0 \in EC(S)$). $O(N)$
- Posons $P = P_0$
- Tant que $succ(P) \neq P_0$ $O(N)$
  - si $(P, succ(P), succ(succ(P))$ est un Tour Gauche alors $Avancer(P)$
  - sinon $Supprimer(succ(P))$, et si $pred(P) \neq P_0$ alors $Reculer(P)$

Boucle en $O(N)$ à condition que toutes opérations $pred, succ$, ... en $O(1)$
Complexité de Graham : $O(N \log N)$ dans le pire des cas.

Méthode de Jarvis

Principe du paquet cadeau

Propriété : *Si $(P_1, \ldots, P_h)$ est un contour positif de $EC(S)$, alors pour tout i, P_{i+1} est un point minimal de $S - \{P_i\}$ pour l'ordre polaire $<_{P_i, \overrightarrow{P_{i-1}P_i}}$*

On commence avec un point d'ordonnée minimale (et s'il y en a plusieurs on choisit celui d'abcisse minimale).

Algorithme de Jarvis

- Soit P_1 point de S d'ordonnée minimale. $O(N)$
- P_2 pt min de $S - \{P_1\}$ pour ordre polaire $<_{P_1, Hz}$. $O(N)$
- Soit $k = 2$
- Répéter $O(Nh)$
  - Soit Q point minimal de $S - \{P_k\}$ pour l'ordre polaire $<_{P_k, \overrightarrow{P_{k-1}P_k}}$
  - $k := k + 1$, $P_k := Q$
- Jusqu'à $P_k = P_1$

Boucle en $O(Nh)$, où h est le nombre de points de $EC(S)$
Complexité de Jarvis $O(Nh)$: interessant si $h << \log N$.
Mais au pire $h = N$, donc Jarvis en $O(N^2)$ dans le pire des cas.

Enveloppe Convexe Dynamique

1. Algorithme statique en $O(n \log n)$
   Graham, dichotomique, ...
2. Ensemble dynamique : ajout et suppression de points
   - Ajouts uniquement
     Algorithme en $O(n \log n)$ (Preparata)
   - Ajouts et suppressions
     Algorithme en $O(n \log^2 n)$ (Overmars, Van Leeuwen)

Algorithme de Preparata : fondé sur algorithme de cône enveloppant, avec représentation de l'enveloppe convexe par un arbre AVL.

Algorithme du cône enveloppant

Soit S un ensemble de points, et $EC(S) = (q_0, \dots, q_k)$.
Soit P un nouveau point.

1. si $P \subset EC(S)$ alors $EC(S \cup \{P\}) = EC(S)$
2. sinon, soit δ demi-droite issue de P et ne coupant pas $EC(S)$
   - soit $R = min(S)$ pour l'ordre polaire par rapport à δ ;
     $r \in EC(S)$
   - soit $L = max(S)$ pour l'ordre polaire par rapport à δ ;
     $l \in EC(S)$
3. $EC(S) = (q_0, \dots, L, \dots, R, \dots, q_k)$ et
   $EC(S \cup \{p\}) = (q_0, \dots, L, P, R, \dots, q_k)$

Implantation avec Listes

Liste doublement chaînée de $EC(S) = (q_0, q_1, \dots, q_k)$ en ordre direct.
$EC(S \cup \{p\})$ se calcule en $O(k)$ à partir de $EC(S)$, avec $k \leq ||S||$

1. vérifier si p est à l'intérieur de $EC(S)$ – coût $O(k)$,
2. trouver le min et le max de $EC(S)$ pour l'ordre polaire par rapport à δ – coût $O(k)$,
3. supprimer de L à R et insérer p – coût $O(1)$, si accès à L et R.

Donc la complexité de construction de l'EC de n points est en $O(\sum_{k=3}^{n-1} k) = O(n^2)$

Implantation avec AVL

$EC(S)$ représenté par un arbre équilibré T, de hauteur $O(\log k)$
Circuit polaire de $EC(S) \equiv$ Parcours symétrique de T

1. p est à l'intérieur de $EC(S)$? – coût $O(\log k)$ (recherche dans l'arbre)
2. min et max dans l'ordre polaire – coût $O(\log k)$ (recherche dans l'arbre)
3. nouvelle EC – coût $O(\log k)$ (coupure, concaténation, insertion)

Donc la complexité de construction de l'EC de n points est en $O(\sum_{k=3}^{n-1} \log k) = O(n \log n)$

Arbres AVL

Définition : un arbre AVL est un ABR tel que les hauteurs de 2 noeuds frères diffèrent au plus de 1.
Propriété : *tout arbre AVL contenant n noeuds et de hauteut H vérifie*

$$\log_2(n+1) \leq H+1 < 1,44 \log_2(n+2)$$

- Recherche d'un élément : $O(\log n)$
- Ajout d'un élément : $O(\log n)$ – avec au plus 1 rotation
- Suppression d'un élément : $O(\log_n)$ – avec au plus $\log n$ rotations
- Concaténation – $O(\log n)$: (G, x, D) avec G et D AVL et $G \leq x < D$, renvoie un AVL contenant tous les éléments
- Coupure – $O(\log n)$: (T, x) avec T AVL renvoie le couple (G, D) avec G et D AVL et $G \leq x < D$

Concaténation

Concatenation : AVL-EC * AVL-EC * Element → AVL-EC
Hypothèse : $G \leq x < D$
renvoie *un arbre AVL contenant G,x et D*

```
Fonction Concatenation( G,D,x)
   Si |h(G) - h(D)| <= 1 Retourne ArbreBinaire(x,G,D)
   SinonSi h(G) > h(D) +1 Retourne
     Equilibrage(ArbreBinaire(rac(G),SousArbreGauche(G),
                    Concatenation(SousArbreDroit(G),x,D)))
   Sinon Retourne
     Equilibrage(ArbreBinaire(rac(D),Concatenation(G,x,SousArbreGauche(D)),
                    SousArbreDroit(D)))
   FinSi
Fin Fonction Concatenation
```

Nombre d'appels récursifs : $O(|h(G) - h(D)|)$
Complexité en $O(\log n)$

Coupure

Coupure : AVL-EC * Element → AVL-EC * AVL-EC
renvoie *un couple d'arbres AVL (G,D) tels que G < x et x < D*

```
Fonction Coupure( T,x)
   Si EstVide(T) Retourne (vide,vide)
   SinonSi Estfeuille(T)
     Si x<=rac(T) Retourne (vide,T)
     Sinon Retourne (T,vide)
     FinSi
   SinonSi x<=rac(T)
     Soit (G0,D0) = Coupure( SousArbreGauche(T),x)
     Retourne (G0,Concatenation(D0,rac(T), SousArbreDroit(T)))
   Sinon
     Soit (G0,D0) = Coupure( SousArbreDroit(T),x)
     Retourne Concatenation((SousArbreGauche(T),rac(T),G0),D0)
   FinSi
Fin Fonction Coupure
```

Complexité en $O(\log n)$

Représentation de l'EC par un AVL

L'arbre T qui représente l'enveloppe convexe AVL de hauteur $O(\log k)$

- Parcours symétrique de $T \equiv$ Circuit polaire de $EC(S)$ double chaînage en ordre symétrique
- Le min de T est le premier élément du circuit polaire accès au min
- Chaque noeud contient la hauteur du sous-arbre dont il est la racine

EXEMPLE

Position des points

Soit $EC(S)$ l'enveloppe convexe des k points et P un nouveau point.

Définition : soit $Q \in EC(S)$:

- si $Q = L$ ou $Q = R$, alors Q est dit *porteur* par rapport à P
- si Q est entre L et R , alors Q est dit *réflexe* par rapport à P
- si Q est entre R et L , alors Q est dit *concave* par rapport à P

Calcul de la position en temps $O(1)$

- Q concave : $(P, Q, succ(Q))$ est un tour gauche, $det(\overrightarrow{PQ}, \overrightarrow{Psucc(Q)}) > 0$
- Q réflexe : $(P, Q, succ(Q))$ est un tour droit, $det(\overrightarrow{PQ}, \overrightarrow{Psucc(Q)}) < 0$
- Q porteur : $det(\overrightarrow{PQ}, \overrightarrow{Ppred(Q)})$ et $det(\overrightarrow{PQ}, \overrightarrow{Psucc(Q)})$ de même signe.

Les différents cas

18 cas possibles : $(\overrightarrow{Pm}, \overrightarrow{PM})$ ($\leq \pi$ ou $> \pi$), m est concave ou réflexe ou porteur par rapport à P, M est concave ou réflexe ou porteur par rapport à P.

1. m concave, M concave et $0 \leq (\overrightarrow{Pm}, \overrightarrow{PM}) \leq \pi$
2. m concave, M réflexe et $0 \leq (\overrightarrow{Pm}, \overrightarrow{PM}) \leq \pi$
3. m réflexe, M réflexe et $0 \leq (\overrightarrow{Pm}, \overrightarrow{PM}) \leq \pi$
4. m réflexe, M concave et $0 \leq (\overrightarrow{Pm}, \overrightarrow{PM}) \leq \pi$
5. m réflexe, M réflexe et $(\overrightarrow{Pm}, \overrightarrow{PM}) > \pi$
6. m réflexe, M concave et $(\overrightarrow{Pm}, \overrightarrow{PM}) > \pi$
7. m concave, M concave et $(\overrightarrow{Pm}, \overrightarrow{PM}) > \pi$
8. m concave, M réflexe et $(\overrightarrow{Pm}, \overrightarrow{PM}) > \pi$

Les 10 autres cas se ramènent aux 8 cas ci-dessus.

Points porteurs du circuit

Porteurs : point * AVL-EC → point ∪ {vide } * point ∪ {vide }
***renvoie** le couple de points porteurs*

```
Fonction Porteurs (P, T, m)
    Si EstVide(T) Retourne (vide,vide) FinSi
    Soient M=rac(T) et m= min(T)
      Si m = M Retourne (vide,vide) FinSi
      SelonCas
        1 ou 5 : Retourne Porteurs(P,SousArbreDroit(T), succ(M))
        2 : Retourne L=chL(T-SousArbreDroit(T)) et R=chR(SousArbreDroit(T))
        3 ou 7 : Retourne Porteurs(P,SousArbreGauche(T), m)
        4 :Retourne L=chL(T-SousArbreGauche(T)) et R=chR(SousArbreGauche(T))
        6 :Retourne L=chL(SousArbreDroit(T)) et R=chR(T-SousArbreDroit(T))
        8 :Retourne L=chL(SousArbreGauche(T)) et R=chR(T-SousArbreGauche(T))
      FinSelon
Fin Fonction Porteurs
```


chL : AVL-EC → point
***renvoie** le le point porteur L*

```
Fonction chL(T)
    C=rac(T)
    SelonCas
      C porteur par rapport à P : Retourne C
      C reflexe par rapport à P : Retourne chL(SousArbreGauche(T))
      C concave par rapport à P : Retourne chL(SousArbreDroit(T))
    FinSelon
Fin Fonction chL
```

chR : AVL-EC → point
***renvoie** le le point porteur R*

```
Fonction chR(T)
    C=rac(T)
    SelonCas
      C porteur par rapport à P : Retourne C
      C concave par rapport à P : Retourne chR(SousArbreGauche(T))
      C reflexe par rapport à P : Retourne chR(SousArbreDroit(T))
    FinSelon
Fin Fonction chR
```

Complexité

1. Remarque :
   Si P est intérieur à EC alors la fonction `Porteurs` renvoie ($vide$, $vide$).
2. Complexité
   - `chL` et `chR` en $O(\log n)$ car recherche dans un AVL
   - `Porteurs` en $O(\log n)$ car
     - soit appelle `chL` ou `chR` dans sous-arbres gauche ou droit
     - soit appelle `Porteurs` dans sous-arbres gauche ou droit

Réorganisation de l'AVL-EC

Etant donnés T, L, R et P, modifier T pour qu'il soit un AVL représentant l'enveloppe convexe de $S \cup \{P\}$.

Deux cas de figure :

- m est concave par rapport à P ou m=L
- m est réflexe par rapport à P ou m=R

Il faut faire des coupures et/ou des concaténations dans l'AVL. Complexité $O(\log n)$.

Algorithme de réorganisation

Reorganisation : : AVL-EC * point * point * point → AVL-EC
***renvoie** l'AVL représentant l'enveloppe convexe incrémentée*

Fonction Reorganisation(T,L,R,P)
  m=min(T)
  Si m est concave par rapport à P ou m=L
    (G1,D1)= Coupure(T,L)
    (G2,D2)= Coupure(D1,R)
    Retourne Concatenation(G1,D2,P)
  Sinon ; m réflexe ou m=R
    (G1,D1)= Coupure(T,L)
    (G2,D2)= Coupure(G1,R)
    Retourne ajout(P,D2)
  FinSi
  succ(P)=R pred(P)=L succ(L)=P pred(R)=P
Fin Fonction *Reorganisation*

Enveloppe convexe incrémentale

AjoutEC : AVL-EC * point → AVL-EC
***renvoie** l'AVL représentant l'enveloppe convexe incrémentée*

Fonction AjoutEC(T, P)
  Soit m = min(T)
  Soient (L,R)= Porteurs(P,T,m)
  Si L=R=vide **Retourne** T
  Sinon Retourne Reorganisation(T,L,R,P)
  FinSi
Fin Fonction *AjoutEC*

EnvConvIncr : : Ens de n points → AVL-EC
***renvoie** l'AVL représentant l'enveloppe convexe des n points*

Fonction EnvConvIncr(p1,p2,...,pn)
  Soit T= AVL(p1,p2,p3)
  Pour i=4 à n T=AjoutEC(T,pi) **FinPour**
  Retourne T
Fin Fonction *EnvConvIncr* O(nlog n)