2D Convex Hull

- Orientation d'un triplet de ponts, chirotope
- Matroides orientés de rang 3 uniforme et acyclique ou CC-systèmes
- Graham's Scan, Jarvis's March
- "Optimal" output sensitive algorithm
- Incremental randomized algorithm and Incremental deterministic algorithm

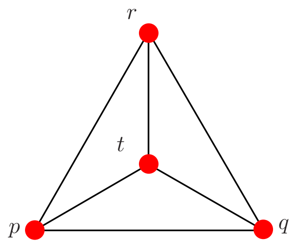
Chirotope

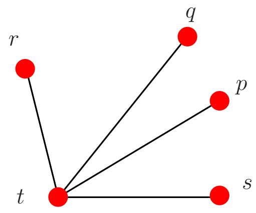
- $P = \{p_1, p_2, \ldots, p_n\} \subset \mathbb{R}^2$
- $\chi : (p, q, r) \in P^3 \mapsto$ orientation du triplet (p, q, r) donnée par le signe du déterminant, noté $|pqr|$, de la matrice $\begin{pmatrix} p & q & r \\ 1 & 1 & 1 \end{pmatrix}$ où p, q et r sont vus comme des vecteurs colonnes.
- #chirotopes $= 2^{\Theta(n \log n)}$
- Propriétés de χ sur au plus 5 points en position générale, i.e., $\chi(p, q, r) \neq 0$

$$pqr \iff \chi(p, q, r) = 1$$

P1: symétrie circulaire	$pqr \Longrightarrow qrp$
P2: antisymétrie	$pqr \Longrightarrow \neg prq$
P3: non dégénérescence	$pqr \vee prq$
P4: intériorité	$tqr \wedge ptr \wedge pqt \Longrightarrow pqr$
P5: transitivité	$tsp \wedge tsq \wedge tsr \wedge tpq \wedge tqr \Longrightarrow tpr$
P6: transitivité duale	$stp \wedge stq \wedge str \wedge tpq \wedge tqr \Longrightarrow tpr$

Chirotope





P4: intériorité $\quad tqr \wedge ptr \wedge pqt \Longrightarrow pqr$
P5: transitivité $\quad tsp \wedge tsq \wedge tsr \wedge tpq \wedge tqr \Longrightarrow tpr$

Proof. De $0 = \det \begin{pmatrix} p & q & r & t \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} = |qrt| - |prt| + |pqt| - |pqr|$, je déduis P4. Dans ce cas nous écrivons $t \in \Delta pqr$ et nous notons que t est combinaison convexe de p, q et r : (régle de Cramer)

$$|pqr|t = |tqr|p + |ptr|q + |pqt|r.$$

Pour P5 raisonnons par l'absurde. Si $\neg tpr$ alors $t \in \Delta pqr$ et par suite $0 = |tst|$ est combinaison convexe des termes positifs $|tsp|, |tsq|$ et $|tsr|$, contradiction. $\square$

Matroide Orienté Uniforme et Acyclique de Rang 3

- Matroide orienté de rang 3 uniforme et acyclique ou CC-système

$$\chi^* : (i, j, k) \in [n]^3 \mapsto ijk \in \{V, F\}$$

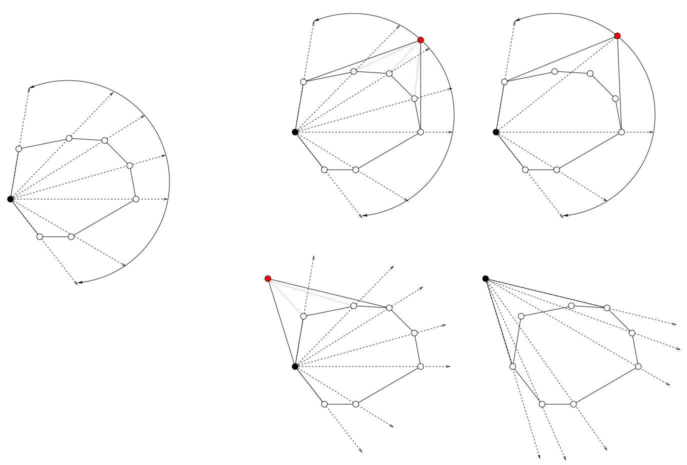
  vérifiant les propriétés P1, P2, P3, P4, P5 et P6
- Matroide réalisable (ou étirable)
- #Matroides orientés de rang 3 $= 2^{\Theta(n^2)}$

Graham's Scan

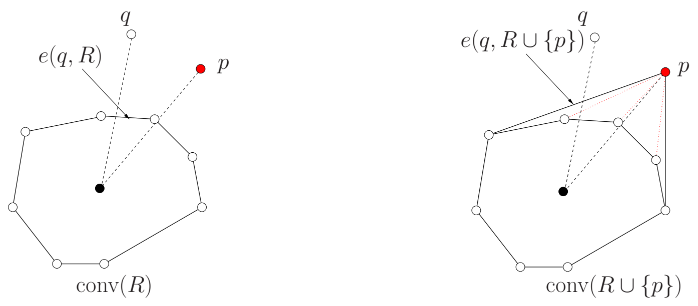
Algorithm GrahamHull2D(P) where $P \subset \mathbb{R}^2$

1. Choose a point p inside the convex hull of P,
   sort circularly the points of P ccw around p, and
   store the points in a circular list $L = \ldots \mathrm{succ}^{-1}(p)p\,\mathrm{succ}(p)\ldots$
2. $(p, q, r) \leftarrow$ three consecutive points of L
3. **Repeat**
4. **if** pqr **then** $(p, q, r) \leftarrow (q, r, \mathrm{succ}(r))$
5. **else** delete q from L and $(p, q, r) \leftarrow (\mathrm{succ}^{-1}(p), p, r)$
6. **until** To be filled
7. **return** the list L

Knuth Incremental Algorithm



Randomized Incremental Algorithm



1. Choisir une permutation aléatoire $p_1p_2p_3\ldots p_n$ des n points de P et poser $R_k = \{p_1, p_2, \ldots, p_k\}$
2. $\mathrm{conv}(R_3) \to \mathrm{conv}(R_4) \to \cdots \to \mathrm{conv}(R_n)$
3. Maintenir une arête visible de chaque point
4. Coût de l'insertion de p est le cardinal de l'ensemble des $q \in P$ tels que $e(q, R) \neq e(q, R \cup \{p\})$
5. Le coût moyen de l'algorithme est donc $\sum_{q \in P} \sum_{k=1}^{k=n} \mathrm{Prob}[e(q, R_{k-1}) \neq e(q, R_k)]$

Rand. Incr. Alg. : Backward Analysis

Pour évaluer $\mathrm{Prob}[e(q, R_{k-1}) \neq e(q, R_k)]$ nous observons que

$$\mathrm{Prob}[e(q, R_{k-1}) \neq e(q, R_k) \mid R_k = R] \leq \frac{2}{k}.$$

Il suit que

$$\mathrm{Prob}[e(q, R_{k-1}) \neq e(q, R_k)] \leq \frac{2}{k}.$$

Il suit que le nombre moyen de changement de $e(q, R)$ est $\sum_{k=1}^{n} 2/k = O(\log n)$, et la complexité moyenne de l'algorithme est un $O(n \log n)$.

Marche de Jarvis (Jarvis's March)

Algorithm JarvisHull2D(P) where $P \subset \mathbb{R}^2$

$p_0 \leftarrow (0, +\infty)$
$p_1 \leftarrow$ the leftmost point of P
for $k = 1, 2, \ldots$ **do**
  $p_{k+1} \leftarrow$ the point q from P that maximizes $\angle p_{k-1} p_k q$
  if $p_{k+1} = p_1$ **then return** the list $p_1, \ldots p_k$

(Open Question) Is it possible to find an extreme point of a CC-system in linear time?

Output sensitive algorithm

Algorithm Hull2D (P, m, H) where $P \subset \mathbb{R}^2$, $3 \le m \le n$, and $H \ge 1$;

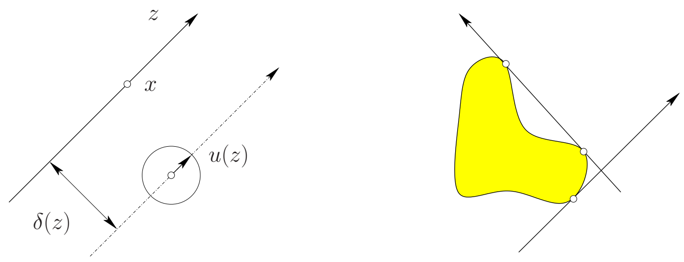
partition P into subsets $P_1, P_2, \ldots, P_{\lceil n/m \rceil}$ each of size at most m
for $i = 1, \ldots \lceil n/m \rceil$ **do**
  compute the convex hull of P_i by Graham's scan and store its vertices in an array in ccw order
$p_0 \leftarrow (0, +\infty)$
$p_1 \leftarrow$ the leftmost point of P
for $k = 1, \ldots, H$ **do**
  for $i = 1, \ldots \lceil n/m \rceil$ **do**
    compute the point $q_i \in P_i$ that maximizes $\angle p_{k-1} p_k q_i$ by performing a binary search on the vertices of the convex hull of P_i
  $p_{k+1} \leftarrow$ the point q from $\{q_1, \ldots, q_{\lceil n/m \rceil}\}$ that maximizes $\angle p_{k-1} p_k q$
  if $p_{k+1} = p_1$ **then return** the list $p_1, \ldots p_k$
return *incomplete*

Output sensitive algorithm

Algorithm Hull2D(P) where $P \subset \mathbb{R}^2$;

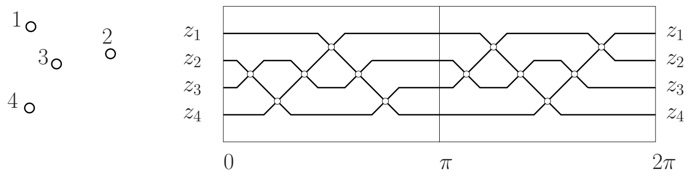
for $t = 1, 2, \ldots$ **do**
  $L \leftarrow$ Hull2D(P, m, H), where $m = H = \min\{2^{2^t}, n\}$
  if $L \neq$ *incomplete* **then return** L

Space of directed lines – Support functions



1. A directed line z is a line with a direction $u(z) \in \mathbb{S}^1$ assigned to it and indicated in the figures by an arrow. The direction-coordinate of z is its direction $u(z)$ and its distance-coordinate is the real $\delta(z)$ defined as the common value of the scalar products $\langle x, \imath u \rangle$ where x ranges over the undirected version of the line. The space of directed line is denoted $\mathbb{D}$ and is homeomorphic to $\mathbb{S}^1 \times \mathbb{R}$ via $z \mapsto (u(z), \delta(z))$
2. The support function of $X \subseteq \mathbb{R}^2$ is the map $\lambda_X : u \in \mathbb{S}^1 \to \inf_{x \in X} \langle x, \imath u \rangle \in \mathbb{R}$ and the directed lines $(u, \lambda_X(u))$ are called the supporting lines of X
3. The support function of X coincides with the support function of its convex hull conv(X).

Convex hulls and lower envelopes

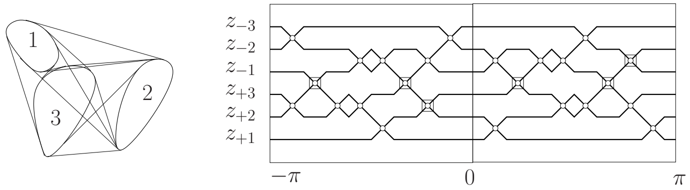


1. $P = \{p_i\}$ une collection de n points en position générale

$$\begin{aligned} \lambda_p(u) &= \langle p, \imath u \rangle \\ \lambda_{\mathrm{conv}(P)} &= \inf_{p \in P} \lambda_p \end{aligned}$$

2. $z_i = \lambda_i \circ \exp$ $\qquad z_i(\theta + \pi) = -z_i(\theta)$

Convex hulls and lower envelopes



1. $O = \{o_i\}$ une collection de n convexes en position générale

$$\begin{aligned} \lambda_{+o}(u) &= \inf_{p \in o} \langle p, \imath u \rangle \\ \lambda_{-o}(u) &= \sup_{p \in o} \langle p, \imath u \rangle \\ \lambda_{\mathrm{conv}(O)} &= \inf_{o \in O} \lambda_{+o} \end{aligned}$$

2. $z_{\epsilon o} = \lambda_{\epsilon o} \circ \exp$ $\qquad z_{-o}(\theta + \pi) = -z_{+o}(\theta)$

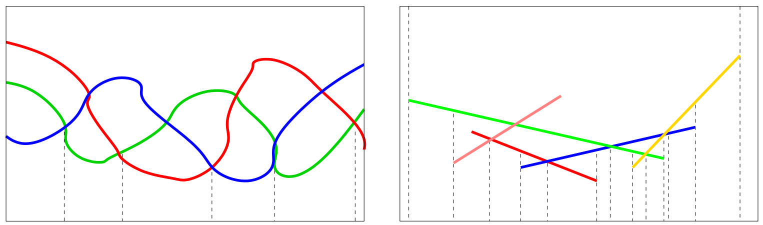
Suites de Davenport-Schinzel

$$ABRACADABRABRICABRA \in \Lambda_8([6])$$

Tout mot sur un alphabet à n lettres ne contenant aucun facteur de type aa et aucun sous-mot alterné de type $abab\ldots$ de longueur $\geq s+2$ est appelé une suite de Davenport-Schinzel de paramètre s sur un alphabet à n lettres.

$$\lambda_s(n) = \max_{\Lambda_s(n)} \{ |u| \}$$

2D Lower Envelopes



Soit $f = \min \mathcal{F}$ l'enveloppe inférieure d'une famille $\mathcal{F} = \{f_i\}$ de n fonctions num. continues définies sur $\mathbb{R}$ dont les graphes sont sécants deux-à-deux en au plus s points.

Soit $x_0 = 0 < x_1 < \cdots < x_m = 1$ la suite minimale de réels telle que pour $x \in [x_{i-1}, x_i]$ on ait $f(x) = f_{j_i}(x)$. On pose $U(\mathcal{F}) = j_0 j_1 \ldots j_m$.

$$\mathcal{F} \overset{\text{comb. equivalence}}{\longrightarrow} U(\mathcal{F}) \in \Lambda_s(n)$$

Bornes

$$\begin{array}{rclrcl}\lambda_1(n) & = & n & \lambda_2(n) & = & 2n-1 \\ \lambda_3(n) & = & \Theta(n\alpha(n)) & \lambda_4(n) & = & \Theta(n2^{\alpha(n)})\end{array}$$

$\alpha(n)$ est un pseudo-inverse de la fonction d'Ackermann $A(n) = A_n(n)$

$$\begin{array}{rcl}A_1(n) & = & 2n \\ A_{k+1}(n) & = & \overbrace{A_k \circ \cdots \circ A_k}^{n}(1)\end{array}$$

$A_2(n) = 2^n$, $A_3(n) = 2^{2^{\cdot^{\cdot}}}$ est la tour de 2 de hauteur n (le pseudo-inverse de $A_3(n)$ est noté $\log^*$). $A_4(3) = 2^{16}$ et $A_4(4) = A_3(A_4(3))$ est une tour de 2 de hauteur $A_4(3) = 2^{16} = 65536$; pour dire que $\alpha(n) \leq 4$ pour tout entier "machine".

Borne en $O(n \log n)$ sur $\lambda_3(n)$

Bibliographie

References

[1] Donald E. Knuth. Axioms and Hulls, volume 606 of Lecture Notes Comput. Sci. Springer-Verlag, Heidelberg, Germany, 1992.

[2] Timothy M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. Discrete Comput. Geom., 16:361–368, 1996.

[3] Micha Sharir and P. K. Agarwal. Davenport-Schinzel Sequences and Their Geometric Applications. Cambridge University Press, New York, 1995.