Géométrie Algorithmique Notes de Cours

Francis Lazarus

3 décembre 2014

Table des matières

1	Intr	Introduction et Rappels			
	1.1	Qu'est	t-ce que la géométrie algorithmique?	5	
	1.2	Algori	thmes et Complexité	6	
	1.3	Tri .		8	
	1.4	Arbres	s binaires	9	
	1.5	Borne	inférieure pour la complexité moyenne des tris	10	
	1.6	Coefficients du binôme			
		1.6.1	Formulaire I - Identités	11	
		1.6.2	Formulaire II - Estimations	12	
	1.7	Proba	bilité discrète élémentaire	13	
		1.7.1	Définitions et propriétés élémentaires	13	
		1.7.2	Probabilités conditionnelles	16	
		1.7.3	Lois classiques	17	
		1.7.4	Technique de Chernoff	18	
	1.8	Exem	ples d'applications de la méthode probabiliste	20	
		1.8.1	Analyse arrière	20	
		1.8.2	Nombres harmoniques	20	
		1.8.3	Ensembles indépendants	20	
		1.8.4	Arbre binaire de recherche aléatoire	21	
		1.8.5	Echantillonnage aléatoire	22	
	1.9	Maste	r theorem	22	
2	Gra	phes p	blanaires	24	
	2.1	Plongements			
	2.2	Du théorème de Jordan à la relation d'Euler			
	2.3	Graphes interdits			

	2.4	Critères de planar	ité	35			
	2.5	Plongements rectil	lignes	37			
	2.6	Algorithmes pour	la planarité	40			
3	Tria	riangulation 42					
	3.1	.1 Existence					
	3.2	Algorithmes		44			
		3.2.1 Algorithme	e diviser pour régner	44			
		3.2.2 Algorithme	e par décomposition en polygones monotones	50			
		3.2.3 Application	n au problème de la galerie d'art	55			
4	Plu	s court chemin d	ans un polygone	58			
5	Rec	herche monodim	ensionnelle	64			
	5.1	Dictionnaires		64			
		5.1.1 Arbres bina	aires de recherche	64			
	5.2	Structures random	iisées	65			
		5.2.1 Skip list .		65			
		5.2.2 Arbres bina	aires de recherche aléatoires	68			
		5.2.3 Tree + Hea	ap = Treap	70			
6	Pol	Polytopes 7					
	6.1	Notations		71			
	6.2	Convexité		72			
	6.3	Le théorème de Minkowski-Weyl		73			
	6.4	Lemmes de Farkas, 1896		76			
	6.5	5 Faces d'un polytope		78			
		6.5.1 Terminolog	;ie	80			
		6.5.2 Treillis des	faces d'un polytope	81			
		6.5.3 Polarité .		82			
	6.6	Faces d'un cône .		84			
		6.6.1 Faces des p	oolyèdres via les cônes	86			
		6.6.2 Polarité po	vur les cônes	86			
	6.7	Exemples de Polyt	copes	88			
		6.7.1 Polytopes of	cycliques	89			

	6.8	Le théorème de la borne supérieure				
		6.8.1	Bonne orientation acyclique	92		
		6.8.2	<i>h</i> -vecteur	92		
		6.8.3	Preuve du théorème de la borne supérieure	95		
	6.9	Steinit	tz,	95		
7	Pro	gramn	nation Linéaire	96		
	7.1	Dualit	é de la programmation linéaire	96		
		7.1.1	Application de la dualité	98		
	7.2	Algori	thmes du simplexe	101		
	7.3	La cor	ajecture de Hirsch	104		
	7.4	Progra	ammation linéaire en temps linéaire	106		
8	Env	veloppe	es convexes, Voronoi et Delaunay	110		
	8.1	Calcul	ls d'enveloppes convexes	110		
		8.1.1	Algorithmes naïfs	110		
		8.1.2	Calcul en 2D	111		
		8.1.3	Calcul en 3D	113		
		8.1.4	Calcul en dimension quelconque	116		
	8.2	.2 Diagramme de Voronoi				
		8.2.1	Lien avec les enveloppes supérieures	117		
		8.2.2	Calcul en 2D	118		
		8.2.3	Variantes du diagramme de Voronoi	120		
	8.3	Triang	gulation de Delaunay	121		
		8.3.1	Lien avec les enveloppes convexes	121		
		8.3.2	Dualité entre les diagrammes de Delaunay et de Voronoi \ldots .	123		
		8.3.3	Algorithme randomisé incrémental pour Delaunay	125		
9	Arr	rrangements 12				
	9.1	Introduction : problème de la discrépance				
	9.2	Préliminaire : subdivision du plan				
	9.3	Arrangement de droites				
	9.4	Arran	gement d'hyperplans	132		
		9.4.1	Dénombrement des faces et incidences	133		

	9.5	Niveaux d'un arrangement	137			
	9.6	Dualité	137			
10 Localisation						
	10.1	Localisation par découpe en bandes verticales	140			
	10.2	Localisation par décomposition trapézoïdale	142			
	10.3	Localisation dans une triangulation	149			
11	Rec	herche multidimensionnelle	152			
	11.1	Recherche orthogonale	152			
		11.1.1 Recherche unidimensionnelle	153			
		11.1.2 Kd-trees (arbres k-dimensionnels) (Bentley 1975)	154			
		11.1.3 Arbres de domaines	155			
		11.1.4 Fractionnement en cascade	157			
	11.2	Recherche simpliciale et par demi-espaces	158			
		11.2.1 Arbre de partitions	158			
		11.2.2 Arbres de cuttings et recherche par demi-plan	160			
		11.2.3 Application à la recherche simpliciale	162			
12 Cuttings et partitions simpliciales						
	12.1	Cuttings	164			
	12.2	Échantillonnage aléatoire	167			
	12.3	Partitions Simpliciales	170			
13 Algorithmes randomisés incrémentaux						
	13.1	Introduction	173			
	13.2	Le formalisme	173			
	13.3	Algorithmes statiques	176			
		13.3.1 Formalisme et analyse randomisée	176			
		13.3.2 Applications	180			
14	14 Transversaux, ε-net et dimension de Vapnik-Chervonenkis 18					
	14.1	Transversaux et couplages	184			
	14.2	Dimension de Vapnik-Chervonenkis	185			

Bibliographie

Chapitre 1

Introduction et Rappels

1.1 Qu'est-ce que la géométrie algorithmique?

La géométrie algorithmique est une discipline apparue dans les années 1970 traitant de problèmes géométriques sous l'angle algorithmique. La géométrie algorithmique tire ses références et entretien des liens étroits avec de nombreux domaines tels que : les mathématiques discrètes et combinatoires, la théorie des polytopes, la recherche opérationnelle, la théorie des graphes et l'informatique en général.

Les applications sont nombreuses :

- Infographie (détections des collisions d'objets, faces cachées, visibilité, reconstruction, ...)
- Robotique (planification de trajectoires)
- Système d'informations géographiques (meilleurs chemins, co-raffinement de cartes, ...)
- CAO/MAO (assemblage, déplacements d'outils, ...)

La géométrie algorithmique est bien représentée par un certain nombre de problèmes fondamentaux :

- Programmation linéaire, polytopes, arrangements
 - Enveloppes convexes
 - Triangulation de Delaunay et diagramme de Voronoi
- Recherche (simpliciale, orthogonale) et localisation
- Triangulations
- Plus courts chemins

Un des développements récents concerne la robustesse et l'implémentation des algorithmes (CGAL, LEDA).

Références :

- Computational Geometry. Algorithms and applications. M. de Berg, O. Cheong, M. van Kreveld, M. Overmars and Schwarzkopf. Springer 2008. (Une très bonne introduction).

- Computational Geometry : An Introduction Through Randomized Algorithms. Ketan Mulmuley. Prentice-Hall, 1994. (Couvre les aspects liés à la randomisation ou à l'analyse randomisée des algorithmes).

- Géométrie Algorithmique. Jean-Daniel Boissonnat et Mariette Yvinec. Ediscience International, 1995.

- Computational Geometry. F. Preparata and M. Shamos. Springer 1985. (Un des premiers ouvrages sur le sujet, un peu daté).

- Handbook of Discrete and Computational Geometry. Edited by Goodman and O'Rourke. CRC PRESS, 1997. (L'état de l'art. Complet et pointu. Idéal pour chercher un résultat).

- Handbook of Computational Geometry. Edited by J. R. Sack and J. Urrutia North Holland, 2000

- The Computational Geometry Impact Task Force Report, B. Chazelle and 36 co-authors, Advances in Discrete and Computational Geometry, Contemporary Mathematics, 223, AMS, Providence, 1999, pp. 407–463.

http://www.cs.princeton.edu/~chazelle/pubs.html

1.2 Algorithmes et Complexité

La complexité d'un algorithme fait référence à un modèle de calcul ou plus précisément à un type de calculateur théorique auquel est associé une fonction de coût pour chacune de ses opérations élémentaires. La *complexité* d'un algorithme exprime le rapport asymptotique entre la taille des données d'un problème et le temps (ou l'espace) nécessaire pour exécuter l'algorithme correspondant sur le type de calculateur choisi. Le modèle RAM (Random Access Machine) est un des plus courants en géométrie algorithmique. Il se compose

- d'une suite finie de registres d'entrée pouvant contenir des entiers arbitraires et ne pouvant être accédés qu'en lecture seule,
- d'une bande infinie de registres, indexés par N et pouvant contenir des entiers arbitraires; il s'agit intuitivement de la mémoire de la machine; le registre 0, appelé accumulateur, permet d'effectuer des opération arithmétiques,
- d'un programme constitué d'une suite finie d'instructions élémentaires choisies parmi un ensemble prédéfini d'instructions (READ, STORE, ADD, ...) paramétrées par des indices (adressage direct ou indirect) de registres (d'entrée ou non),
- d'un compteur d'instructions contenant l'indice de la prochaine instruction à effectuer,
- (optionnellement) d'une suite finie de registres de sortie. On peut aussi décider que le second registre de la mémoire contient la sortie.

En général on associe un coût unité à chaque instruction élémentaire bien que cette instruction accède à des registres d'indices arbitraires (d'où le nom de RAM). Mais on peut également choisir pour l'accès à un à registre donné un coût logarithmique fonction de son indice. On dira que la complexité d'un programme est f(n) si pour toute entrée de taille n, codée sur n registres, le coût total de l'exécution du programme est au plus f(n). Il s'agit donc de la complexité dans le cas le pire relativement à tous les jeux de données de taille n.

On suppose en général que la taille des registres est constante ce qui revient à définir la taille de l'entrée par le nombre total de bits utilisés pour la coder. Dans ce cas, si on restreint les opérations arithmétiques à l'addition et la soustraction, bien que le modèle RAM puisse manipuler en temps constant des entiers de tailles arbitraires et accéder à un registre de mémoire en temps constant, il est polynomialement équivalent au modèle de la machine de Turing (MT)! C'est-à-dire que toute fonction calculable par une RAM peut être calculée par une MT en temps dépendant de manière polynomiale du temps requis par la RAM, et réciproquement.

Pour se concentrer sur la partie purement combinatoire d'un problème, on peut supposer que la machine RAM peut stocker et manipuler (effectuer les opérations/comparaisons $+, -, *, /, <, =, \sqrt{, ...}$) des réels en temps constant, on parle alors de modèle Real-RAM. Ce pourra être le cas lorsqu'on doit manipuler des coordonnées de points dans \mathbb{R}^n ou d'autres objets géométriques. Les registres pouvant contenir des réels sont alors différenciés de ceux qui peuvent contenir des addresses. Dans le cas contraire le modèle de machine serait bien trop puissant. On pourra consulter [vEB90] pour une sensibilisation aux questions de complexités et de modèles de calculs. Sauf mention explicite d'un autre modèle, tous les calculs de complexités dans ce cours utilisent le modèle de la Real-RAM.

Un certain vocabulaire, plus ou moins propre à la géométrie algorithmique, permet de décrire les algorithmes en fonction de leur adaptation au flux des données, ou en fonction de leur principe de fonctionnement. Ainsi un algorithme est dit *statique* ou *hors-ligne* s'il s'applique à un jeu de données connu avant son exécution. On parle d'algorithme *en ligne* ou *semi-dynamique* si au contraire de nouvelles données peuvent être ajoutées, et que l'algorithme est capable, au cours du temps, de maintenir la solution du problème associé à toutes les données accumulées. Enfin, un algorithme *dynamique* permet d'ajouter ou de supprimer des données au cours du temps.

La conception d'un algorithme peut être *déterministe* ou *randomisée* selon que l'algorithme aura ou non toujours le même comportement face à un même jeu de données. Dans ce dernier cas, les variations de comportement sont liées à l'usage de générateurs aléatoires de nombres (ou bits) qui permettent de modifier le déroulement de l'algorithme.

On peut analyser de manière randomisée la complexité d'un l'algorithme en la considérant comme une variable aléatoire. Cette variable aléatoire peut dépendre soit des générateurs aléatoires dans le cas des algorithmes randomisés, soit d'une distribution a priori sur les données dans le cas déterministe. Dans le cas d'un algorithme randomisé, on suppose généralement que le jeu de données est le pire possible. Ainsi lorsqu'on calcule l'espérance de la complexité, encore appelée la *complexité en moyenne*, on se place dans le cas où les données maximise cette espérance. On peut également s'intéresser à la *queue de distribution*, c'est à dire à la probabilité que la complexité dépasse une certaine valeur.

Les méthodes *incrémentales, par balayage ou diviser pour régner* apparaissent de manière récurrente en géométrie algorithmique comme principe de résolution de problèmes.

Références :

- Calculabilité, complexité et approximation. J.-F. Rey. Vuibert, 2004.

- The Design an Analysis of Computer Algorithm. Chap. 1. Aho, Hopcroft et Ullman. Addison Wesley, 1974.

- Computational complexity. Christos Papadimitriou. Addison Wesley, 1995.

- Machine Models and Simulations. van Emde Boas, P. in Handbook of theoretical computer science, vol A. Elsevier, 1990.

- What is a "pointer machine"?. M. Ben-Amram. SIGACT News, 26 (1995), 88–95.

1.3 Tri

Le tri d'éléments, pris dans un univers totalement ordonné, est un des problèmes basiques et essentiels de l'algorithmique. Étant donné une séquence de N éléments le problème est de trier cette séquence, i.e. de trouver la permutation transformant cette séquence en la séquence ordonnée de ses éléments.

De très nombreux algorithmes permettent de trier. Il y en a de plus "efficaces" que d'autres. On mesure cette efficacité par le nombre (maximal, moyen, ...) d'opérations élémentaires utilisées pour trier une suite quelconque de taille N, c.a.d. par la complexité de l'algorithme considéré. L'opération élémentaire pour les tris est la comparaison.

Exemples :

- **tri naïf** : complexité?
- tri par insertion dichotomique : Insérer par dichotomie le (k + 1)-ième élément dans la liste triée des k premiers éléments. On note C(k) la complexité dans le pire des cas, en anglais on parle de worst case analysis, de l'insertion du (k + 1)-ième élément. On a

$$C(0) = 0$$

$$C(k) = 1 + C(\lceil \frac{k-1}{2} \rceil) = 1 + C(\lfloor \frac{k}{2} \rfloor)$$

On vérifie par récurrence que

$$C(k) = \lceil 1 + \log k \rceil$$

en utilisant le fait que pour k > 1

$$\lceil \log 2\lfloor \frac{k}{2} \rfloor \rceil = \lceil \log k \rceil.$$

La complexité B(N) du tri de N éléments est donc

$$B(N) = \sum_{k=0}^{N-1} C(k) = O(N \log N).$$

Attention, cette analyse ne tient pas compte de la gestion de la mémoire.

 tri fusion : On scinde en deux la liste à trier et on trie récursivement chacune des sous-listes avant de les fusionner.

$$CF(N) = CF(\lceil N/2 \rceil) + CF(\lfloor N/2 \rfloor) + N = O(N \log N).$$

 Quicksort (Hoare 1962) : Choisir aléatoirement un pivot dans la liste à trier puis scinder en deux la liste suivant les éléments plus petits ou plus grands que le pivot. Trier récursivement chaque sous-liste. Il s'agit du tri utilisé par UNIX. Très efficace en pratique.

Analyse dans le cas le pire : Chaque clé est choisie une unique fois comme pivot. Il y a au plus N-1 comparaisons à effectuer avec un pivot, d'où une complexité en $O(N^2)$. Ce sera le cas si les pivots sont pris dans l'ordre croissant des clés.

Analyse en moyenne : les clés K_1, \ldots, K_N étant indexées selon leur ordre de grandeur, on note $X_{i,j}$ la variable aléatoire qui vaut 1 si K_i et K_j sont comparées et 0 sinon. K_i et K_j sont comparées si et seulement si elles n'ont pas été séparées avant que l'une soit choisie comme pivot, c.a.d. si et seulement si l'une de ces deux clés est choisie comme pivot en premier parmi les j - i + 1 clés comprises (au sens large) entre K_i et K_j . Ceci se produit avec une probabilité 2/(j - i + 1), d'où

$$E(\sum_{i < j} X_{i,j}) = \sum_{i < j} \frac{2}{j - i + 1} = \sum_{j=2}^{N} \sum_{k=2}^{j} \frac{2}{k} \le 2NH_N = O(N \log N)$$

où $H_N = \sum_{i=1}^N 1/i$ est le N-ième nombre harmonique.

Voir également pour une preuve, l'analyse en moyenne de la hauteur d'un arbre de recherche aléatoire.

Le tri d'entiers codés en binaire peut utiliser d'autres moyens que la comparaison. En faisant des hypothèses sur la taille des nombres à trier, on obtient des complexités moindre. Pour un article récent sur le sujet, voir :

- Integer Sorting in $O(n\sqrt{\log \log n})$ Expected Time and Linear Space. Y. Han and M. Thorup. pp 135 - 144. FOCS 2002, Vancouver, Canada.

1.4 Arbres binaires

L'ensemble des arbres binaires est défini inductivement par l'équation ensembliste $\mathcal{B} = \Box + (\circ, \mathcal{B}, \mathcal{B})$. Dit autrement un arbre binaire est soit l'arbre vide soit obtenu en accrochant deux arbres binaires (sous-arbres gauche et droit) à une racine. On note sag(B) (resp. sad(B)) le sous-arbre gauche (resp. droit) d'un arbre binaire B.

Une opération importante sur les arbres binaires est la rotation :

$$(\circ, (\circ, b_1, b_2), b_3) \mapsto (\circ, b_1, (\circ, b_2, b_3))$$

Codage informatique : un noeud est représenté par un objet à trois champs : la clé du noeud, deux pointeurs sur des noeuds représentants les sous-arbres gauche et droit.

Définitions : *Racine, noeud, noeud interne, noeud externe ou feuille, arête, enfant, parent,* etc...

Un arbre binaire *complet* est un arbre binaire dont tous les noeuds internes ont deux enfants. On peut compléter un arbre binaire quelconque en ajoutant deux (resp. une) feuilles à tous ses noeuds ayant zéro (resp. un) enfant.

Propriété : Dans un arbre binaire complet on a : $#{\text{noeuds externes}} = #{\text{noeuds internes}} + 1.$

Preuve : Orienter les arêtes des noeuds parents vers les enfants. Le nombre d'arêtes sortantes est le double du nombre de noeuds internes, c'est aussi le nombre d'arêtes entrantes qui est le nombre total de noeuds moins 1 (pour la racine). \Box

On définit récursivement les *taille, hauteur, longueurs de cheminement interne et externe* d'un arbre binaire complet par

 $\begin{array}{l} \mbox{taille, hauteur, lci, lce}(\Box) = 0 \\ \mbox{taille}(b) = 1 + \mbox{taille}(\mbox{sag}(b)) + \mbox{taille}(\mbox{sad}(b)) \\ \mbox{hauteur}(b) = 1 + \mbox{max} \{ \mbox{hauteur}(\mbox{sag}(b)), \mbox{hauteur}(\mbox{sad}(b)) \} \\ \mbox{lci}(b) = \mbox{taille}(b) - 1 + \mbox{lci}(\mbox{sag}(b)) + \mbox{lci}(\mbox{sad}(b)) \\ \mbox{lce}(b) = \mbox{taille}(b) + 1 + \mbox{lce}(\mbox{sag}(b)) + \mbox{lce}(\mbox{sad}(b)) \\ \mbox{lce}(\mbox{sad}(b)) + \mbox{lce}(\mbox{sad}(b)) \\ \mbox{lce}(b) = \mbox{taille}(b) + 1 + \mbox{lce}(\mbox{sag}(b)) + \mbox{lce}(\mbox{sad}(b)) \\ \end{tabular} \end{array}$

On définit également récursivement la profondeur d'un noeud par profondeur(racine) = 0 profondeur(noeud) = 1 + profondeur(parent(noeud))C'est aussi le nombre d'arêtes du chemin simple reliant le noeud à la racine.

Dit autrement un arbre binaire a une taille égale à son nombre de noeuds internes, une hauteur égale à la profondeur maximale d'un noeud externe et une longueur de cheminement interne (resp. externe) égale à la somme des profondeurs de tous ses noeuds internes (resp. externes).

On vérifie par récurrence que lce(b) = lci(b) + 2 taille(b).

Un arbre binaire complet est dit *parfait* si tous ses noeuds externes ont la même profondeur.

Propriétés : Dans un arbre binaire parfait, b, de hauteur h on a :

taille(b) =
$$\sum_{i=0}^{h-1} 2^i = 2^h - 1$$
,
lce(b) = $h2^h$,
lci(b) = $(h-2)2^h + 2$.

Propriétés : Le nombre d'arbres binaires à *n* sommets est le *nombre de Catalan* $C_n = \frac{1}{n+1} \binom{2n}{n}$.

1.5 Borne inférieure pour la complexité moyenne des tris

Les différentes comparaisons effectuées lors du déroulement d'un algorithme déterministe de tri sur N clés données s'organisent selon un arbre binaire complet. Les feuilles de cet arbre correspondent aux N! permutations possibles des N clés. Le coût maximal (resp. moyen) de cet algorithme est la profondeur maximale (resp. moyenne) des feuilles de son arbre de comparaisons. C'est encore, d'après la section précédente, la hauteur (resp. le rapport de la lce par la taille + 1) de son arbre de comparaisons. **Propriété** : pour une taille, t, fixée la lce d'un arbre binaire complet est minimisée par l'arbre parfait.

Preuve : On note b_t l'arbre parfait de taille t et b un arbre quelconque de taille t. De la formule récursive de la lce, on tire

lce(b) - lce(b_t) = lce(sag(b)) + lce(sad(b)) - 2 lce($b_{(t-1)/2}$) En utilisant l'hypothèse de récurrence lce(b) ≥ lce(b_t) = (t + 1) log(t + 1) aux ordres convenables on a

 $lce(b) - lce(b_t) \ge (t_g + 1) \log(t_g + 1) + (t_d + 1) \log(t_d + 1) - (t + 1) \log(t + 1)$

avec $t_g + t_d = t - 1$. On vérifie, en utilisant par exemple la convexité de $x \mapsto x \log x$, que le membre de droite est positif ou nul. La formule étant triviale pour un arbre de taille 0, ceci permet de confirmer l'hypothèse de récurrence.

De la propriété précédente on déduit que la complexité moyenne d'un algorithme de tri est minorée par $N! \log(N!)/N! = O(N \log N)$

1.6 Coefficients du binôme

Il existe une quantité impressionnante de formules à propos de ces coefficients. Elles interviennent souvent en combinatoire, mathématique discrète, géométrie algorithmique, etc... dès que l'on cherche à estimer dans un ensemble le nombre de sous-ensembles vérifiant telle ou telle propriété. On pourra consulter le chapitre 5 de

Mathématiques concrètes - Fondations pour l'informatique. R.L. Graham, D. E. Knuth, O. Patashnik. Vuibert 2003, 2e édition.

Définition 1.1 Un échantillon de taille k d'un ensemble est un sous-ensemble de cardinal k. Le nombre d'échantillons de taille k dans un ensemble de taille n est noté C_n^k (lire "binomiale de n, k") ou $\binom{n}{k}$.

On note N_n un ensemble à n éléments.

1.6.1 Formulaire I - Identités

•
$$\binom{n}{k} = \binom{n}{n-k}$$
.

Preuve : L'application qui envoie un échantillon de taille k sur son complémentaire (de taille n - k) est une bijection.

• Formule de Pascal $\binom{n+1}{k+1} = \binom{n}{k+1} + \binom{n}{k}$.

Preuve : Compter séparément les échantillons contenant ou non un élément donné.

•
$$\binom{n}{k}\binom{k}{p} = \binom{n}{p}\binom{n-p}{k-p}$$
.

Preuve : Compter de deux manières différentes le nombre d'échantillons de taille k dans N_n dont p éléments sont distingués.

Pour
$$p = 1$$
 on obtient :
• $k \binom{n}{k} = n \binom{n-1}{k-1}$ puis, avec la formule de Pascal, $\binom{n}{k+1} = \frac{n-k}{k+1} \binom{n}{k}$ et $\binom{n}{k} = \frac{n}{n-k} \binom{n-1}{k}$

On en déduit par récurrence :

•
$$\binom{n}{k} = \prod_{i=0}^{k-1} \frac{n-i}{k-i}.$$

• Formule du binôme $(x+y)^n = \sum_{k=0}^n \binom{n}{k} x^k y^{n-k}$.

Preuve : Par récurrence avec la formule de Pascal ou en exprimant comment obtenir le nombre de monômes $x^k y^{n-k}$.

•
$$\sum_{k=0}^{n} \binom{n}{k} = 2^{n}.$$

Preuve : Compter de deux manières différentes le nombre total de sous-ensembles de N_n .

•
$$\sum_{k=0}^{n} \binom{n}{k}^2 = \binom{2n}{n}.$$

Preuve : Écrire $\binom{n}{k}^2 = \binom{n}{k}\binom{n}{n-k}$ et remarquer que l'ensemble des paires d'échantillons, (E_k, E'_{n-k}) , de taille respective k et n-k pris dans deux ensembles E et E' à n éléments est en bijection avec l'ensemble des échantillons de taille n de $E \cup E'$.

•
$$\sum_{i=k}^{n} \binom{i}{k} = \binom{n+1}{k+1}.$$

Preuve : Numéroter de 1 à n + 1 les éléments de N_{n+1} , puis partitionner l'ensemble de ses échantillons de taille k + 1 en fonction de leur plus grand élément.

• $\sum_{i=1}^{n} \frac{1}{i} \binom{i}{k} = \frac{1}{k} \binom{n}{k}.$

Preuve : Écrire $\frac{1}{i}\binom{i}{k} = \frac{1}{k}\binom{i-1}{k-1}$ et utiliser la formule précédente.

•
$$\sum_{i=0}^{n} i \binom{i}{k} = (k+1)\binom{n+2}{k+2} - \binom{n+1}{k+1}.$$

Preuve : Écrire $i\binom{i}{k} = (i+1)\binom{i}{k} - \binom{i}{k} = (k+1)\binom{i+1}{k+1} - \binom{i}{k}.$

1.6.2 Formulaire II - Estimations

• $\binom{n}{k} \le n^k$.

Preuve : Îl y a moins d'échantillons de taille k dans N_n que d'applications de N_k vers N_n . (Écrire $\binom{n}{k} = \binom{n}{n-k}$ si k > n/2).

•
$$(\frac{n}{k})^k \le {\binom{n}{k}} \le (\frac{en}{k})^k \text{ (et même } \sum_{i=0}^k {\binom{n}{i}} \le (\frac{en}{k})^k \text{).}$$

Preuve : Pour l'inégalité de gauche écrire $\binom{n}{k} = \prod_{i=0}^{k-1} \frac{n-i}{k-i}$ et remarquer que $\frac{n-i}{k-i} \ge \frac{n}{k}$. Pour l'inégalité de droite, écrire : $\forall x > 0$: $\exp(nx) \ge (1+x)^n = \sum_{i=0}^n \binom{n}{i} x^i \ge \binom{n}{k} x^k$. D'où $\binom{n}{k} \leq x^{-k} \exp(nx)$ et on conclut en posant x = k/n.

• $\binom{n}{\lfloor n/2 \rfloor} = \binom{n}{\lceil n/2 \rceil} = \max_{0 \le k \le n} \binom{n}{k}$. Preuve : De $\binom{n}{k} = \frac{n-k+1}{k} \binom{n}{k-1}$ on tire $\binom{n}{k} > \binom{n}{k-1}$ si $k \le n/2$, et on utilise l'égalité $\binom{n}{k} = \binom{n}{n-k}$ pour $k \ge n/2$.

 $\frac{2^{2n}}{2\sqrt{n}} \le \binom{2n}{n} \le \frac{2^{2n}}{\sqrt{2n}}.$ Preuve : $\binom{2n}{n}^2 = \frac{((2n)!)^2}{(n!)^4} = \frac{(1.2...2n)^2}{(1.2...n)^4} = 2^{4n} \frac{(1.2...2n)^2}{(2.4...2n)^4} = 2^{4n} \frac{(1.3...(2n-1))^2}{(2.4...2n)^2}.$ D'où $\binom{2n}{n}^2 = \frac{3^2}{2.4} \frac{5^2}{4.6} \dots \frac{(2n-1)^2}{(2n-2)2n} \frac{2^{4n}}{2.2n} \ge \frac{2^{4n}}{2.2n},$ et $\binom{2n}{n}^2 = \frac{1.3}{2^2} \frac{3.5}{4^2} \dots \frac{(2n-1)(2n+1)}{2n^2} \frac{2^{4n}}{2n+1} \le \frac{2^{4n}}{2n}$

Probabilité discrète élémentaire 1.7

Espace de probabilité, variable aléatoire, indépendance, espérance, inégalité de Markov, probabilité conditionnelle, inégalité de Chebyshev, inégalité de Jensen, distribution géométrique, binomiale. Bornes de Chernoff.

Je vais suivre le 'reading assignment' du 'graduate program' de Zurich :

http://www.ti.inf.ethz.ch/ew/courses/RandAlgs00/RandAlgs.html

Autres références :

- The probalistic Method. Alon and Spencer, John Wiley 2000.

- http://kam.mff.cuni.cz/~matousek/lectnotes.html

- http://cermics.enpc.fr/~delmas/enseignement.html

1.7.1Définitions et propriétés élémentaires

Définition 1.2 Un espace de probabilité est un ensemble Ω munit d'une application $P: \Omega \to \mathbb{R}^+$ telle que $\sum_{\omega \in \Omega} P(\omega) = 1$ (lorsque Ω est infini cette somme est définie comme le sup. sur les parties finies). Un élément de Ω est appelé une réalisation et une partie de Ω est appelée un événement.

Définition 1.3 Une variable aléatoire (réelle) est une application (à valeurs réelles) définie sur un espace de probabilité.

Définition 1.4 Deux variables aléatoires X et Y sont dites indépendantes si

$$\forall x \in ImX, \forall y \in ImY : P(X = x \land Y = y) = P(X = x).P(Y = y)$$

Les n variables aléatoires X_i , i = 1, ..., n sont mutuellement indépendantes si pour tout $(x_1, x_2, \ldots, x_n) \in ImX_1 \times ImX_2 \times \ldots \times ImX_n$:

$$P(X_1 = x_1 \land X_2 = x_2 \land \ldots \land X_n = x_n) = P(X_1 = x_1) \cdot P(X_2 = x_2) \cdot \ldots \cdot P(X_n = x_n)$$

Plus généralement, les variables aléatoires d'une famille $\{X_i\}_{i \in I}$ indéxée par un ensemble dénombrable I sont mutuellement indépendantes si pour toute partie finie $J \subset I$ les variables de la famille $\{X_j\}_{j \in J}$ sont mutuellement indépendantes.

Exercice 1.5 Définir la notion d'indépendance à l'aide des probabilités conditionnelles (cf. définition 1.16).

Exercice 1.6 Soient deux variables aléatoires indépendantes $X : \Omega \to F$ et $Y : \Omega \to G$ et une fonction $f : F \to H$. Montrer que f(X) et Y sont indépendantes.

Définition 1.7 L'espérance d'une variable aléatoire X est

$$E(X) = \sum_{x \in ImX} xP(X = x) = \sum_{\omega \in \Omega} X(\omega)P(\omega)$$

lorsque ces sommes existent. Sa variance est

$$var(X) = E((X - E(X))^2) = E(X^2) - E(X)^2.$$

Par la suite on considère des variables aléatoires dont l'espérance existe. Le lemme suivant est une application directe de la définition de l'espérance (sous sa seconde forme ci-dessus).

Lemme 1.8 L'espérance est linéaire.

Exercice 1.9 Soient une variable aléatoire $X : \Omega \to F$ et une fonction $f : F \to \mathbb{R}$. Montrer que

$$E(f(X)) = \sum_{x \in ImX} f(x)P(X = x).$$

Le lemme suivant est particulier aux fonctions entières non négatives mais bien utile.

Lemme 1.10 Si X est à valeurs naturelles alors

$$E(X) = \sum_{k \in \mathbb{N}} P(X > k)$$

Preuve :
$$\sum_{k \in \mathbb{N}} P(X > k) = \sum_{k \in \mathbb{N}} \sum_{i > k} P(X = i) = \sum_{i > 0} i P(X = i) = E(X).$$

Lemme 1.11 Soient X et Y deux variables aléatoires indépendantes, alors

$$E(XY) = E(X)E(Y) \ et \ var(X+Y) = var(X) + var(Y).$$

Preuve :

x

$$E(XY) = \sum_{\omega \in \Omega} X(\omega)Y(\omega)P(\omega) = \sum_{x,y \in \mathbb{R}} \sum_{X(\omega)=x \land Y(\omega)=y} xyP(\omega) = \sum_{y \in \mathbb{R}} xyP(X(\omega) = x \land Y(\omega) = y) = \sum_{x,y \in \mathbb{R}} xyP(X(\omega) = x).P(Y(\omega) = y) = E(X)E(Y).$$

I en déduit l'égalité sur les variances.

On en déduit l'égalité sur les variances.

Lemme 1.12 (Inégalité de Markov) Soit X une variable aléatoire non négative, alors

$$\forall \lambda > 0 : P(X \ge \lambda) \le \frac{E(X)}{\lambda}$$

Il y a équité si et seulement si pour tout ω de probabilité non nulle : $X(\omega) \in \{0, \lambda\}$.

Preuve :

$$E(X) = \sum_{\omega \in \Omega} X(\omega) P(\omega) \ge \sum_{X(\omega) \ge \lambda} X(\omega) P(\omega) \ge \lambda P(X \ge \lambda).$$

Dit autrement, la probabilité qu'une variable aléatoire non négative dépasse un certain nombre de fois son espérance est majorée par l'inverse de ce nombre.

Lemme 1.13 (Inégalité de Chebychev) Soit X une variable aléatoire non négative, alors

$$\forall \lambda > 0 : P(|X - E(X)| \ge \lambda) \le \frac{var(X)}{\lambda^2}$$

Preuve : Appliquer Markov à $(X - E(X))^2$.

Lemme 1.14 (Inégalité de Jensen) Soit X une variable aléatoire et f une fonction convexe, alors

$$f(E(X)) \le E(f(X)).$$

Preuve : Si Ω est fini cette inégalité traduit simplement le fait que l'image du barycentre d'un ensemble fini de points par une fonction convexe est majorée par le barycentre des images de ces points. Le cas général demande un peu plus de travail. Je note $\tau_f(x,y) =$ $\frac{f(x)-f(y)}{r-u}$ le taux d'accroissement de f. Pour tout $s \leq u$ on a $\tau_f(E(X), s) \leq \tau_f(E(X), u)$. Soit $\beta = \sup_{s < E(X)} \tau_f(E(X), s)$, on vérifie à l'aide de l'inégalité précédente que

$$\forall x, f(x) \ge f(E(X)) + \beta(x - E(X)).$$

Dit autrement, le graphe de f est au dessus de sa "tangente" au point (E(X), f(E(X))). En substituant $X(\omega)$ à x et en prenant les espérances des deux membres de l'inégalité on obtient la relation cherchée.

1.7.2 Probabilités conditionnelles

Définition 1.15 Soit (Ω, P) un espace de probabilité et soit $B \subset \Omega$ un événement de probabilité non nulle. L'espace induit (B, P_B) est l'espace de probabilité sur l'ensemble B avec $P_B(\omega) = P(\omega)/P(B)$.

Définition 1.16 Soient A et B deux événements avec P(B) > 0, la probabilité conditionnelle de A par rapport à B est définie par

$$P(A|B) = \frac{P(A \land B)}{P(B)}.$$

Soient X une variable aléatoire et B un événement de probabilité positive, alors la variable aléatoire X|B est la restriction de X à B dans l'espace induit (B, P_B) .

On vérifie que $P_B(X|B=x) = P(X=x|B)$.

Exercice 1.17 Soient deux événements $B \subset A$ avec P(B) > 0 et X une variable aléatoire. Montrer que (X|A)|B = X|B.

Lemme 1.18 Soient A un événement, X une variables aléatoire, et $(B_i)_{i \in I}$ une famille d'événements disjoints dont la réunion est l'espace Ω . En particulier, $\sum_{i \in I} P(B_i) = 1$. Alors,

$$P(A) = \sum_{i \in I} P(A|B_i) P(B_i)$$

et

$$E(X) = \sum_{i \in I} E(X|B_i)P(B_i)$$

Preuve : Par définition de la probabilité conditionnelle :

$$\sum_{i \in I} P(A|B_i)P(B_i) = \sum_{i \in I} P(A \land B_i)$$

D'où la première égalité par hypothèse sur les B_i . Pour la seconde égalité, on écrit :

$$\sum_{i \in I} E(X|B_i)P(B_i) = \sum_{i \in I} \sum_{x \in \mathbb{R}} xP_{B_i}(X|B_i = x)P(B_i) = \sum_{x \in \mathbb{R}} \sum_{i \in I} xP(X = x|B_i)P(B_i)$$

et on termine à l'aide de la première égalité.

Remarque : Ce lemme est souvent bien pratique pour évaluer ou majorer une espérance ou une probabilité. En effet, si $E(X|B_i)$ (resp. $P(A|B_i)$) est constant ou uniformément borné par rapport aux B_i alors cette constante ou borne reste valable *inconditionnellement*, i.e. pour E(X) (resp. P(A)) : il suffit de mettre en facteur la constante ou borne dans la seconde égalité et d'utiliser le fait que $\sum_{i \in I} P(B_i) = 1$.

Exercice 1.19 Si X est une variable aléatoire indépendante de l'événement B (i.e. $P_B(X|B=x) = P(X=x)$), montrer que E(X|B) = E(X).

Exercice 1.20 Soient A_1, A_2, \ldots, A_n et B des événements. Montrer que

$$P(\bigwedge_{i} A_{i}|B) = \prod_{i} P(A_{i}|\bigwedge_{j>i} A_{j} \wedge B)$$

Exercice 1.21 Soient A un événement et X une variables aléatoire. Montrer une version conditionnelle du lemme 1.18, i.e.

$$P(A|B) = \sum_{i} P(A|B_i)P(B_i|B)$$
$$E(X|B) = \sum_{i} E(X|B_i)P(B_i|B)$$

où B est la réunion disjointe des B_i .

Exercice 1.22 Soit (Ω, P) un espace de probabilité, U un ensemble fini et $A : \Omega \to \mathcal{P}(U)$. Montrer que

$$E(|A|) = \sum_{u \in U} P(u \in A)$$

Exercice 1.23 Sous les hypothèses de l'exercice précédent, on considère de plus une famille de variables aléatoires $(X_u)_{u \in U}$ telle que l'espérance conditionnelle $E(X_u | u \in A)$ est uniformément bornée par une constante c. Montrer que

$$E(\sum_{u \in A} X_u) \le cE(|A|)$$

où $\sum_{u \in A} X_u$ désigne la variable aléatoire $\omega \mapsto \sum_{u \in A(\omega)} X_u(\omega)$. On notera en particulier que la linéarité de l'espérance ne peut s'appliquer à une somme portant sur un ensemble qui dépend de la réalisation ω .

Montrer par un contre-exemple que l'inégalité ci-dessus est généralement fausse si on suppose seulement que $E(X_u)$ est uniformément bornée par c.

1.7.3 Lois classiques

Définition 1.24 (loi de Bernoulli) Supposons avoir un sac avec une proportion p de boules blanches et 1-p de boules rouges. Si on tire une boule au hasard dans le sac, alors la probabilité d'obtenir une boule blanche est p et la probabilité d'obtenir une boule rouge est 1-p. On dit que la variable aléatoire valant 1 lorsque la boule tirée est blanche et 0 sinon suit une loi de Bernoulli de paramètre p.

Définition 1.25 (loi géométrique) Avec les hypothèses précédentes la probabilité d'obtenir une boule blanche après i tirages avec remise vaut $(1-p)^{i-1}p$. Si X est la variable aléatoire valant le nombre de tirages effectués avant d'obtenir une boule blanche alors cette probabilité est précisément P(X = i) et on dit que X a une distribution (ou loi) géométrique de paramètre p.

On vérifie par calcul direct que E(X) = 1/p et $var(X) = (1-p)/p^2$. Si b_1 est la couleur de la première boule tirée, on peut aussi écrire $E(X) = E(X|b_1 = \text{blanc})p + E(X|b_1 = \text{rouge})(1-p)$, puis remarquer que $(X|b_1 = \text{blanc}) = 1$ et que $(X|b_1 = \text{rouge}) = 1 + Y$, où Y est le nombre de tirages effectués avant d'obtenir une boule blanche à partir du second tirage. Évidement, Y a la même distribution que X et on en déduit une équation simple pour E(X). Un calcul similaire permet de calculer 'directement' var(X).

Question : Quel est l'espace de probabilités de X? Votre modèle entre-t-il dans le cadre des probabilités discrètes ? (cf. notion de schéma de Bernoulli).

Définition 1.26 (loi binomiale) Toujours avec les mêmes hypothèses, on considère la variable aléatoire Y valant le nombre de boules blanches obtenues après n tirages avec remise. On dit que Y suit une distribution binomiale de paramètres n et p. On a clairement $P(Y = i) = {n \choose i} p^i (1-p)^{n-i}$.

Remarque : si Y_i est la variable aléatoire (de Bernoulli) qui vaut 1 si le *i*-éme tirage est une boule blanche et 0 sinon, alors $Y = \sum_{1 \le i \le n} Y_i$.

On en déduit que $E(Y) = \sum_{1 \le i \le n} E(Y_i) = np$. De plus les Y_i étant indépendantes, on a $var(Y) = \sum_{1 \le i \le n} var(Y_i) = np(1-p)$.

Question : Quel est l'espace de probabilité de Y?

Définition 1.27 (loi binomiale négative) Toujours avec les mêmes hypothèses on considère la variable aléatoire Z valant le nombre de tirages nécessaires pour obtenir n boules blanches. On dit que Z a une distribution binomiale négative de paramètres n et p. On a $P(Z = i) = {i-1 \choose n-1} p^n (1-p)^{i-n}$.

Remarque : si Z_i est la variable aléatoire qui vaut le nombre de tirages entre les tirages des *i*-ème et (i + 1)-ème boules blanches, alors $Z = \sum_{0 \le i \le n-1} Z_i$. Les Z_i sont indépendantes et suivent une distribution géométrique de paramètre p.

On en déduit que $E(Z) = \sum_{0 \le i \le n-1} E(Z_i) = \frac{n}{p}$. (On peut aussi faire un calcul direct en utilisant le fait que $\sum_i {\binom{i+n}{n}} (1-p)^i = \frac{1}{n!} \sum_i (i+n) \dots (i+1)(1-p)^i = \frac{1}{n!} (-1)^n (\frac{1}{p})^{(n)} = \frac{n!}{p^{n+1}}$). On a également $var(Z) = \sum_{0 \le i \le n-1} var(Z_i) = \frac{n(1-p)}{p^2}$.

1.7.4 Technique de Chernoff

Soit X une variable aléatoire égale à la somme $\sum_{1 \le i \le n} X_i$ de *n* variables aléatoires indépendantes et de lois identiques (i.i.d). La technique de Chernoff [Che52] permet en général de trouver de bons majorants pour $P(X \ge x)$ où x est choisit comme un écart à la valeur moyenne E(X) de X. Pour cela on considère un réel $\lambda > 0$ et on écrit

$$P(X \ge x) = P(e^{\lambda X} \ge e^{\lambda x}) \le E(e^{\lambda X})/e^{\lambda x} = \prod_{1 \le i \le n} E(e^{\lambda X_i})/e^{\lambda x} = E(e^{\lambda X_1})^n/e^{\lambda x}.$$

La première inégalité est celle de Markov, la seconde égalité provient de l'indépendance des variables et la dernière de l'identité des lois des X_i . Il reste à choisir convenablement λ pour obtenir une bonne majoration.

Exemples d'application de la technique de Chernoff.

Lemme 1.28 Soit une variable aléatoire $X = \sum_{1 \le i \le n} X_i$ où les X_i sont mutuellement indépendantes à valeurs dans $\{-1, 1\}$ avec $P(X_i = 1) = P(X_i = -1) = 1/2$. Alors

$$\forall x > 0 : \qquad P(X \ge x) < e^{-\frac{x^2}{2n}}$$

Preuve : Par la technique de Chernoff on a pour tout $\lambda > 0 : P(X \ge x) \le E(e^{\lambda X_1})^n / e^{\lambda x} = (\cosh \lambda)^n / e^{\lambda x}$. En développant cosh en série entière on vérifie que $\cosh \lambda < e^{\lambda^2/2}$, d'où $P(X \ge x) < e^{n\lambda^2/2 - \lambda x}$. On obtient le résultat en choisissant $\lambda = x/n$.

Lemme 1.29 Soit une variable aléatoire X de loi binomiale négative de paramètres n et 1/2. Alors

$$\forall x \ge 3 : \qquad P(X \ge (2+x)n) < \exp(-nx/4)$$

Preuve : Notons que E(X) = 2n. Par la technique de Chernoff on a pour tout $\lambda > 0$: $P(X \ge (2+x)n) \le E(e^{\lambda X_1})^n / e^{\lambda(2+x)n}$ où X_1 suit une loi géométrique de paramètre 1/2. Or

$$E(e^{\lambda X_1}) = \sum_{i=1}^{\infty} e^{\lambda i} / 2^i = \frac{e^{\lambda}}{2 - e^{\lambda}}.$$

Cette dernière égalité supposant $e^{\lambda} < 2$. On a dans ce cas $P(X \ge (2+x)n) \le (\frac{e^{-\lambda(1+x)}}{2-e^{\lambda}})^n$. On choisit λ tel que $e^{\lambda} = 1 + \frac{x}{2+x}$ (donc $e^{\lambda} < 2$), d'où, en utilisant l'inégalité $1 - u < e^{-u}$ pour u > 0:

$$\frac{e^{-\lambda(1+x)}}{2-e^{\lambda}} = \left(1 - \frac{x}{2+2x}\right)^{1+x} \left(1 + \frac{x}{2}\right) < e^{-\frac{x}{2+2x}(1+x)} \left(1 + \frac{x}{2}\right) = e^{-\frac{x}{2}(1+x)} \left(1 + \frac{x}{2}\right)$$

Or pour $x \ge 3$ on vérifie que $1 + x/2 < e^{x/4}$, ce qui permet de conclure.

Références :

- Computational Geometry. An Introduction Trough Randomized Algorithms. K. Mulmuley, Prentice Hall, 1994.

1.8 Exemples d'applications de la méthode probabiliste

1.8.1 Analyse arrière

On munit l'ensemble des permutations de [1,n] de la distribution uniforme. On considère la variable aléatoire X comptant le nombre de minima successifs en lisant une permutation (a_1, \ldots, a_n) de gauche à droite.

$$X = |\{i \in [1, n] \mid a_i = \min\{a_1, \dots, a_i\}\}|$$

On considère également la variable aléatoire X_i valant 1 si a_i est un minimum et 0 sinon. Alors $X = \sum_{1 \le i \le n} X_i$. Pour calculer $P(X_i = 1)$, i.e. $P(a_i = \min\{a_1, \ldots, a_i\})$, on fixe a_{i+1}, \ldots, a_n , d'où le nom d'analyse arrière. On remarque alors que

$$P(X_i = 1 \mid a_{i+1}, \dots, a_n \text{ fixes}) = 1/i$$

d'où $P(X_i = 1) = 1/i$ (cf. lemme 1.18). On en déduit $E(X) = \sum_{1 \le i \le n} 1/i = H_n$.

1.8.2 Nombres harmoniques

On vérifie que $\forall n \ge 1$ $\ln(n+1) \le H_n \le 1 + \ln n$.

1.8.3 Ensembles indépendants

Un sous-ensemble de sommets d'un graphe est *indépendant* si le graphe induit sur ces sommets ne contient pas d'arête.

Théorème 1.30 Tout graphe G à n sommets et m arêtes contient un sous-ensemble de sommets indépendants de taille au moins $\lfloor n/\sqrt{m} \rfloor$.

Preuve : On regarde la probabilité pour qu'un sous-ensemble de k sommets soit indépendant. On pose que chaque échantillon de taille k est équiprobable. Une arête de Grelie deux sommets d'un échantillon de taille k avec la probabilité

$$\frac{\binom{n-2}{k-2}}{\binom{n}{k}} = \frac{k(k-1)}{n(n-1)}$$

puisqu'il y a $\binom{n-2}{k-2}$ échantillons contenant les extrémités de l'arête. Un sous-ensemble de k sommets n'est pas indépendant si et seulement si au moins une des m arêtes de G relie deux de ses sommets. La probabilité de cet événement est majorée par $m\frac{k(k-1)}{n(n-1)}$. Par conséquent un échantillon est indépendant avec probabilité au moins $1 - m\frac{k(k-1)}{n(n-1)}$. Si cette valeur est positive il y a nécessairement (au moins) un sous-ensemble de k sommets indépendants, ce qui est le cas si $k = \lfloor n/\sqrt{m} \rfloor$.

On peut obtenir une autre minoration :

Théorème 1.31 (Túran) Tout graphe G à n sommets et m arêtes contient un sousensemble de sommets indépendants de taille α_G au moins égale à $n^2/(2m+n)$.

Preuve : On numérote les sommets de 1 à n et on associe à toute permutation π de [1, n] l'ensemble, E_{π} , des sommets u de G tels que $\pi(u) > \pi(v)$ pour tous les voisins v de u. On voit que E_{π} est un ensemble de sommets indépendants dans G. L'espérance de la taille de E_{π} est donc un minorant pour α_{G} . La probabilité que le sommet i soit dans E_{π} est la probabilité que $\pi(i) = \max\{\pi(j) \mid j = i \text{ ou } j \text{ est voisin de } i\}$. Si d_i est le degré de i dans G alors, en posant toutes les permutations équiprobables, la probabilité de cet événement est $1/(d_i + 1)$. Par conséquent

$$E(|E_{\pi}|) = \sum_{i=1}^{n} \frac{1}{d_i + 1}$$

Le théorème se déduit de la relation sommets/arêtes et du fait que la moyenne arithmétique majore la moyenne harmonique. $\hfill \Box$

1.8.4 Arbre binaire de recherche aléatoire

Un arbre binaire de recherche aléatoire est obtenu en insérant les valeurs successives d'une permutation aléatoire dans un arbre binaire de recherche vide au départ. On s'intéresse à la hauteur moyenne d'un arbre de recherche aléatoire

On considère la variable aléatoire $Y_n^{(i)}$ valant la profondeur de la clé de rang *i* (i.e. de *i* si on prend [1, n] pour l'ensemble des clés) dans un arbre de recherche aléatoire sur *n* clés. Si Y_n est la variable aléatoire valant la hauteur d'un arbre de recherche aléatoire alors $Y_n = \max\{Y_n^{(1)}, \ldots, Y_n^{(n)}\}.$

En utilisant l'inégalité de Jensen on peut écrire

$$E(Y_n) \le \log E(2^{Y_n}) = \log E(2^{\max\{Y_n^{(1)}, \dots, Y_n^{(n)}\}}) < \log E(\sum_{i \text{ est une feuille}} 2^{Y_n^{(i)}}).$$

On pose $Z_n = \sum_{i \text{ est une feuille}} 2^{Y_n^{(i)}}$, alors

$$E(Z_n) = \frac{1}{n} \sum_{1 \le i \le n} E(Z_n | \text{racine a rang i}) = \frac{2}{n} \sum_{1 \le i \le n} (E(Z_{i-1}) + E(Z_{n-i})).$$

En posant $z_n = E(Z_n)$, on a $z_0 = 0, z_1 = 1$ et

$$z_n = \frac{4}{n} \sum_{1 \le i \le n-1} z_i$$

Par conséquent on a pour $n \ge 3, nz_n - (n-1)z_{n-1} = 4z_{n-1}$. D'où

$$\frac{z_n}{(n+3)(n+2)(n+1)} = \frac{z_{n-1}}{(n+2)(n+1)n} = \dots = \frac{1}{30}$$

On en déduit $z_n = O(n^3)$ puis $E(Y_n) = O(\log n)$.

Théorème 1.32 La hauteur moyenne d'un arbre de recherche aléatoire sur n clés est un $O(\log n)$.

1.8.5 Echantillonnage aléatoire

Étant donné un ensemble E à n éléments, un r-échantillon aléatoire de E est un sous-(multi-)ensemble à r éléments de E tiré selon une certaine distribution. Les distributions les plus courantes sont les suivantes.

- Loi uniforme : On munit l'ensemble des sous-ensembles de taille r de E, noté $\binom{E}{r}$ de la loi uniforme. Chaque r-échantillon a donc la même probabilité d'occurence.
- **Tirage avec remise :** On munit E de la loi uniforme. Un r-échantillon est alors un échantillon de E^r munit de la loi produit. Dit autrement on obtient un r-échantillon en tirant au hasard de manière indépendante r fois un élément de E. Un même élément peut donc se retrouver plusieurs fois dans un r-échantillon.
- Échantillonage de Bernoulli : Chaque élément de E est tiré indépendamment avec une (loi de Bernoulli de) probabilité r/n. Un r-échantillon a donc r éléments en moyenne mais peut posséder plus ou moins d'éléments.

Exercice 1.33 Expliciter pour chacune des trois lois précédentes la probabilité d'occurrence d'un échantillon donné. Vérifier que la somme des probabilités sur l'ensemble des "échantillons" vaut 1.

Ces trois distributions donnent des résultats asymptotiques semblables dans la pratique pour les calculs de complexité en moyenne. L'utilisation de l'une ou l'autre des distributions se justifie souvent par la commodité des calculs (voir Mulmuley [Mul94, chap. 5]).

1.9 Master theorem

Théorème 1.34 Soit la relation de récurrence sur \mathbb{N}^*

$$\Gamma(n) = aT(n/b) + f(n)$$

avec $a \ge 1, b > 1$. Ici T(n/b) peut désigner $T(\lfloor n/b \rfloor)$ ou $T(\lceil n/b \rceil)$. Alors

$$T(n) = \begin{cases} \Theta(n^{\log_b a}) & si \ f(n) = O(n^{\log_b a - \epsilon}) \ pour \ un \ certain \ \epsilon > 0\\ \Theta(f(n)) & si \ f(n) = \Omega(n^{\log_b a + \epsilon}) \ et \ af(n/b) \le cf(n)\\ pour \ c < 1 \ et \ pour \ n \ assez \ grand\\ \Theta(n^{\log_b a} \log n) & si \ f(n) = \Theta(n^{\log_b a}). \end{cases}$$
(1.1)

Preuve : Voir par exemple [CLRS02, sec. 4.3] pour une preuve pas à pas. Sinon, en posant $U(n) = T(b^n)/a^n$ on obtient $U(n) = U(n-1) + f(b^n)/a^n$. D'où $U(n) = U(0) + \sum_{1 \le i \le n} f(b^i)/a^i$. Notons que U(0) = T(1). On obtient assez simplement

$$\sum_{1 \le i \le n} f(b^i)/a^i = \begin{cases} \Theta(1) & \text{si } f(n) = O(n^{\log_b a - \epsilon}) \text{ pour un } \epsilon > 0\\ \Theta(f(b^n)/a^n) & \text{si } af(n/b) \le cf(n) \text{ pour } c < 1 \text{ et } n \text{ assez grand}\\ \Theta(n) & \text{si } f(n) = \Theta(n^{\log_b a}). \end{cases}$$

On en déduit U(n) = O(1) dans le premier cas, $U(n) = \Theta(f(b^n)/a^n)$ dans le second cas compte tenu de $f(n) = \Omega(n^{\log_b a + \epsilon})$, et $U(n) = \Theta(n)$ dans le dernier cas. \Box

Chapitre 2

Graphes planaires

Les graphes planaires apparaissent sous la forme de réseaux routiers, de frontières dans les cartes géographiques, de circuits imprimés ou encore dans un cadre plus théorique, comme les graphes sommets/arêtes (1-squelettes) des 3-polytopes. La plupart des propriétés des graphes planaires font appel de manière plus ou moins explicite au fameux théorème de Jordan.

2.1 Plongements

Définition 2.1 Un plongement d'un graphe G = (S, A) dans un espace X est la donnée d'une injection $S \hookrightarrow X$ et pour chaque arête $a \in A$, d'un plongement $p_a : [0,1] \to X$ dont les extrémités coïncident avec l'injection de celles de a, de sorte que les plongements de deux arêtes ne s'intersectent qu'en leurs extrémités communes, le cas échéant. On appelle arc le plongement d'une arête.

Définition 2.2 Un graphe est planaire s'il peut être plongé dans le plan. Un graphe plan est un graphe plongé dans le plan. Dit autrement un graphe plan est un plongement particulier d'un graphe planaire. Un plongement d'un graphe dont tous les arcs sont polygonaux est dit polygonal - ou PL (pour piecewise linear). Une face d'un graphe plan est une composante connexe du complémentaire du plongement du graphe dans le plan.

Par la suite on utilisera la même notation pour un graphe plan et son plongement. Ainsi, si s est un sommet du graphe plan G, alors G - s désignera soit le graphe G privé de s et des arêtes incidentes à s, soit son plongement, i.e. le plongement de G privé de s et des arcs correspondant aux arêtes incidentes à s.

Lemme 2.3 Un ouvert de \mathbb{R}^2 connexe (par arcs) est connexe par arcs polygonaux simples.

Preuve : Soit $\gamma : p \rightsquigarrow q$ un chemin dans un ouvert Ω connexe. Considérer le sup des $t \in [0, 1]$ tels qu'il existe un chemin polygonal simple reliant $x \ge \gamma(t)$ dans Ω . Montrer que ce nombre vaut nécessairement 1.



FIGURE 2.1 – (a) Le plongement initial du graphe G avec les disques D_* . (b) Les arcs C_{**} . (c) L'arc C_{pq} est remplacé (d) par un arc polygonal dans le plan privé de D_r et des autres arcs puis (e) coupé aux bords des disques D_p et D_q . (f) Le plongement polygonal de G finalement obtenu.

Lemme 2.4 Tout graphe planaire admet un plongement polygonal.

Preuve : Considérons un plongement d'un graphe planaire G = (S, A). On choisit pour chaque sommet $p \in S$ du plongement un disque D_p de centre p de sorte que D_p n'intersecte que les arcs incidents à p. On rétrécit si besoin ces disques afin qu'ils soient deux à deux disjoints. Pour chaque arc joignant deux sommets p et q, on considère une composante C_{pq} de cet arc joignant D_p à D_q dans le plan privé des disques $\mathbb{R}^2 \setminus \bigcup_{s \in S} D_s$. On considère maintenant le sous-ensemble du plan formé des disques D_p et des arcs C_{pq} . Par le lemme précédent, $C_{p,q}$ peut être remplacé par un arc polygonal dans le plan privé des autres arcs et des disques D_r pour $r \neq p,q$. On peut ensuite extraire de cet arc polygonal un sous arc $C'_{p,q}$ joignant D_p à D_q sans rencontrer l'intérieur de ces disques. Finalement, en prolongeant tous les arcs polygonaux $C'_{p,q}$ ainsi obtenus par des segments de droites joignant les centres des disques D_p et D_q , on obtient un plongement polygonal. La figure 2.1 résume les différentes étapes de la preuve.

2.2 Du théorème de Jordan à la relation d'Euler

Le théorème de Jordan (du mathématicien français Camille Jordan. 1838 - 1922) établit qu'une courbe fermée simple du plan sépare le plan en deux composantes bordées par cette courbe. Ce résultat évident en apparence est singulièrement difficile à montrer. Des controverses sur la justesse de la preuve originale de Jordan continuent de tourmenter la communauté mathématique. La preuve qui suit est tirée de

A Proof of the Jordan Curve Theorem. Helge Tverberg. Bull. London Math. Soc. 12(1980),

pp. 34-38.

Elle consiste à montrer ce théorème pour les courbes polygonales et à l'attendre aux courbes continues par un processus de passage à la limite.

Une autre preuve a été donnée par Thomassen dans *The Jordan-Schönflies Theorem and the classification of surfaces*. Carsten Thomassen. American Mathematical Monthly. Feb 1992. pp 116-129.

également reprise dans

Graphs on Surfaces. Bojan Mohar et Carsten Thomassen. Johns Hopkins university Press, 2001.

Comme chez Tverberg, Thomassen commence par traiter le cas des courbes polygonales. Le cas général est ramené (de manière élégante en ce qui concerne la non-connexité du complémentaire d'une courbe) à la non-planarité de $K_{3,3}$.

Ces preuves font appel à un minimum de topologie et se limitent plus ou moins aux implications classiques de la compacité pour les applications continues tout en exploitant le plongement dans le plan euclidien. Les preuves sont de ce fait relativement accessibles bien qu'assez fastidieuses. On trouvera cependant dans les livres de topologie algébriques des preuves plus générales (où l'on traite des injections d'une (d-1)-sphère dans une *d*sphère) ne faisant pas appel à la structure euclidienne du plan mais qui utilisent les suites de Mayer-Vietoris pour le calcul de l'homologie des espaces en jeu. Voir par exemple *Elements of Algebraic Topology*. James Munkres. Perseus Books, 1984.

Le cas plus restreint des courbes différentiables est traité dans

Géométrie différentielle : variétés, courbes et surfaces. Marcel Berger et Bernard Gostiaux. PUF mathématiques, 1987.

Théorème 2.5 (Jordan - version polygonale) Soit C une courbe polygonale fermée simple. Alors $\mathbb{R}^2 \setminus C$ a deux composantes, l'une bornée, l'autre non-bornée, toutes deux bordées par C.

Preuve : Notons tout d'abord que C étant compact $\mathbb{R}^2 \setminus C$ a exactement une composante non bornée. Soit une direction \vec{d} transverse aux segments de C que l'on appellera direction horizontale. On considère les segments de C semi-ouverts supérieurement (s.o.s), i.e. privés de leur sommet supérieur. On note $\pi(z)$ la parité du nombre de segments s.o.s de C coupés par la demi-droite horizontale (z, \vec{d}) . On vérifie que π est localement constante dans $\mathbb{R}^2 \setminus C$, voir figure 2.2. Donc π est constante sur chaque composante de $\mathbb{R}^2 \setminus C$. De plus, π prend des valeurs distinctes de chaque côté d'un segment quelconque de C. Il suit que $\mathbb{R}^2 \setminus C$ a au moins deux composantes. Soit V_C un voisinage tubulaire de C, et soit D un petit disque intersectant C en un segment, de sorte que D est coupé par Cen exactement deux composantes. Considérons alors un point p de $\mathbb{R}^2 \setminus C$. Il existe un chemin dans $\mathbb{R}^2 \setminus C$ joignant p à V_C (exemple : un segment de droite). En prolongeant ce chemin dans $\mathbb{R}^2 \setminus C$ on peut s'arranger pour qu'il joigne D en restant dans $\mathbb{R}^2 \setminus C$. Il suit que p est dans la même composante que l'une des deux composantes de $D \setminus C$. Dit autrement, $\mathbb{R}^2 \setminus C$ a au plus deux composantes et donc exactement deux composantes. Par ailleurs les arguments qui précédent montrent également que tout point de C est adhérent aux deux composantes de $\mathbb{R}^2 \setminus C$.



FIGURE 2.2 – La demi-droite horizontale issue de z coupe 5 s.o.s et $\pi(z)$ est donc impaire. On vérifie que cette parité ne change pas pour toute demi-droite horizontale issue d'un point du disque D_z .

Avant de passer à la version générale du théorème, voici quelques applications.

Corollaire 2.6 (Lemme du θ) Soient C_1, C_2, C_3 trois courbes polygonales simples (non fermées) ayant en commun leurs extrémités p et q et disjoints par ailleurs. Alors le graphe $G = C_1 \cup C_2 \cup C_3$ a précisément 3 faces respectivement bordées par $C_1 \cup C_2, C_2 \cup C_3$ et $C_3 \cup C_1$.

Preuve : Par la version polygonale du théorème de Jordan, les trois courbes fermées simples $G_k = C_i \cup C_j$, $\{i, j, k\} = \{1, 2, 3\}$, séparent le plan en deux composantes et bordent ces mêmes composantes. On note X_k (resp. Y_k) la face bornée (resp. non-bornée) de G_k . On note également $\mathring{C}_i = C_i \setminus \{p, q\}$ l'intérieur relatif de C_i .

Remarquons qu'une courbe polygonale simple (ici privée de ses extrémités) ne peut séparer un ouvert connexe en plus de deux composantes (cf. la preuve du théorème de Jordan polygonal). Comme $\overset{\circ}{C_3}$ est inclus dans l'une des faces de G_3 on en déduit que $G = G_3 \cup \overset{\circ}{C_3}$ a au plus trois faces.

Par ailleurs on a $C_i \subset X_i$ pour au moins un indice $i \in \{1, 2, 3\}$. Dans le cas contraire on a $C_i \subset \mathbb{C}X_i$ et donc $G_i \subset G \subset \mathbb{C}X_i$, d'où $X_i \subset \mathbb{C}G \subset \mathbb{C}G_i$. Dit autrement X_i est une face de G. Comme les X_i sont distincts ($C_i \subset \overline{X}_j$ mais $\mathring{C}_i \not\subset \overline{X}_i$) on en conclut que G a au moins trois faces bornées et donc au moins quatre faces ce qui contredit la remarque précédente. On supposera par la suite $\mathring{C}_3 \subset X_3$.

De $G = G_1 \cup G_2$ on tire que toute face de G est une composante de l'intersection d'une face de G_1 avec une face de G_2 . De $G_3 \subset G \subset CY_3$ on tire que Y_3 est une face de G. Comme Y_3 est non bornée on a $Y_3 \subset Y_1 \cap Y_2$.

Comme $C_1 \subset \overline{Y}_3 \subset \overline{Y}_1 = \complement X_1$, on a en fait $G_1 \subset G \subset \complement X_1$ et donc X_1 est une face de G. De même X_2 est une face de G. Or, Y_3 qui est non bornée est distincte des deux faces bornées X_1 et X_2 , elles mêmes distinctes (C_1 borde X_2 mais pas X_1). On conclut que Y_3 , X_1 et X_2 sont les trois faces de G. **Définition 2.7** Un graphe G est 2-connexe s'il a trois sommets au moins et si pour chacun de ses sommets, s, le graphe G - s est connexe.

Proposition 2.8 Toute face d'un graphe plan polygonal 2-connexe est bordée par un cycle de ce graphe. De plus tout arc du graphe est incident (i.e. adhérent) a exactement deux faces.

Preuve : Soit G un tel graphe. Notons que la 2-connexité implique que tout sommet de G est de degré au moins 2. On raisonne par récurrence sur $\sum_{s \in S(G)} (d(s) - 2)$. Si cette somme est nulle alors G est un cycle et on peut appliquer le théorème de Jordan. Sinon, on considère une chaîne maximale P dans G dont les sommets intérieurs sont de degré 2. On vérifie que G - P est 2-connexe et on peut lui appliquer l'hypothèse de récurrence, puisque la somme ci-dessus diminue. Il suit que P est contenue dans une face de G - P qui est bordée par un cycle de G - P. On applique alors le lemme du θ à l'union de ce cycle et de P pour conclure.

Lemme 2.9 Soit G un graphe plan polygonal, s un sommet de degré un de G. Alors G possède le même nombre de faces que G - s.

Preuve : Toute face de G est évidemment incluse dans une face de G - s. On note a l'unique arête incidente à s. Soient f_1 , f_2 deux faces de G incluses dans une même face de G - s et soit p_i un point intérieur à f_i , i = 1, 2. Alors il existe un chemin polygonal P joignant p_1 et p_2 dont l'intersection avec G est incluse dans l'arc semi-ouvert $a \cup s$. En considérant un petit voisinage tubulaire de $a \cup s$, on peut modifier P en contournant a de manière à éviter G, ce qui montre que p_1 et p_2 sont dans la même face de G, c'est-à-dire $f_1 = f_2$.

Théorème 2.10 (Relation d'Euler) Les nombres F, A et S de faces, arêtes et sommets d'un graphe plan polygonal connexe vérifient la relation, dite d'Euler (1707-1783),

$$F - A + S = 2$$

Preuve : Par récurrence sur A. Si A est nul (et donc S = 1) la relation est trivialement vraie. Soit G un graphe avec A > 0 arêtes. Si G contient un sommet s de degré un, alors par le lemme 2.9, G - s a F faces et l'hypothèse de récurrence appliquée à G - s permet de confirmer la relation d'Euler pour G. Sinon G contient un cycle simple C. Soit a une arête de C. Montrons que G - a a une face de moins que G ce qui permettra de conclure avec l'hypothèse de récurrence. D'après la version polygonale du théorème de Jordan, C sépare le plan en deux régions bordées par C. Comme G est la réunion de C et G - a, toute face de G est incluse dans l'intersection d'une face de C et d'une face de G - a. L'arc a est inclus dans une unique face de G - a, disons f. Toute face de G - a distincte de f ne rencontre pas C et est donc une face de G. Comme f intersecte les deux faces de C, on en déduit que G a au moins une face de plus que G - a. Mais en considérant un voisinage tubulaire de a dans f, on montre par un raisonnement déjà vu que f - a a au plus deux composantes. Donc G a au plus une face de plus que G - a. Finalement G a exactement une face de plus que G - a.

Pour une série de preuves plus ou moins formelles de cette fameuse formule on pourra consulter la page :

http://www.ics.uci.edu/~eppstein/junkyard/euler/ extraite du par ailleurs très intéressant site : "The Geometry Junkyard" maintenu par David Eppstein.

Un célèbre casse-tête, parfois présenté comme le problème de l'eau, du gaz et de l'électricité [BLW98, p.142], demande de dessiner dans le plan 3 maisons et 3 distributeurs pour l'eau, le gaz et l'électricité et de relier chaque maison à chaque distributeur sans que les liens ne se coupent. Le théorème suivant affirme entre autre que c'est impossible.

Théorème 2.11 Le graphe complet K_5 et le graphe bipartite complet $K_{3,3}$ ne sont pas planaires.

Preuve : Supposons $K_{3,3}$ planaire. Par le lemme 2.4, on peut lui appliquer la relation d'Euler, ce qui fournit F = 5. Par ailleurs toute face d'un plongement de $K_{3,3}$ est bordée par un cycle (cf. proposition 2.8) de $K_{3,3}$ ayant au moins 4 arêtes (tout cycle d'un graphe bipartite est de longueur paire!). Par la relation d'incidence face/arête (cf. proposition 2.8) on en déduit $4F \leq 2A$. Une contradiction.

Le cas de K_5 se traite de manière similaire.

Passons maintenant au théorème de Jordan dans sa version générale (plane).

Théorème 2.12 (Jordan) Soit C une courbe fermée simple, i.e. l'image injective par une application continue du cercle unité S^1 dans le plan. Alors $\mathbb{R}^2 \setminus C$ a deux composantes, l'une bornée, l'autre non-bornée, toutes deux bordées par C.

Preuve : Je donne sans entrer dans tous les détails la preuve de Tverberg (1980).

Tout d'abord C peut être approximée d'aussi près que l'on veut par une courbe polygonale. Dit autrement, pour tout $\epsilon > 0$ on peut trouver un polygone de Jordan (une courbe polygonale simple fermée) $C' : S^1 \to \mathbb{R}^2$ tel que

$$|C - C'| := \sup_{x \in S^1} |C(x) - C'(x)| < \epsilon.$$

Pour obtenir C' on commence par couvrir le plan d'une grille à mailles carrées de côté δ (défini plus loin). Soient M_1, \ldots, M_n les mailles de la grille traversées par C (elles sont en nombre fini). On construit récursivement une suite de polygones $C_0 = C, C_1, \ldots, C_n$. Ayant construit C_i , on remplace l'image par C_i d'un plus petit arc de S^1 contenant $C_i^{-1}(M_i)$ par le segment de droite joignant l'images des extrémités de cet arc, obtenant ainsi C_{i+1} . Il est clair que $C' := C_n$ convient. Pour définir δ on choisit tout d'abord ϵ_1 tel que

 $|x-y| < \epsilon_1 \implies |C(x) - C(y)| < \epsilon/2,$

puis ϵ_2 tel que

$$|C(x) - C(y)| < \epsilon_2 \implies |x - y| < \min(\epsilon_1, \sqrt{3}).$$

L'existence de ϵ_1 découle de l'uniforme continuité de C sur le compact S^1 . Celle de ϵ_2 découle de l'uniforme continuité de C^{-1} , inverse d'une bijection continue sur le compact

 $C(S^1)$. On pose alors $\delta = \min(\epsilon/2, \epsilon_2)$. Il reste à vérifier que C_n ainsi construit convient. (La solution est dans l'article de Tverberg). On utilise ensuite les deux lemmes intermédiaires suivants.

Lemme 2.13 Si C' est un polygone de Jordan, alors la composante bornée de $\mathbb{R}^2 \setminus C'$ contient un disque touchant C' en deux points C'(x) et C'(y) avec $x, y \in S^1$ tels que $|x-y| > \sqrt{3}$.

PROOF. On considère un disque D dont l'intérieur est contenu dans la composante bornée de $\mathbb{R}^2 \setminus C'$ et touchant C' en deux points C'(x) et C'(y) de sorte que |x - y|est maximal. Si $|x - y| < \sqrt{3}$ alors tout point z sur le plus grand arc A de S^1 joignant x et y est à une distance de x ou de y supérieure à celle de x à y. On en déduit par l'hypothèse sur x et y que, en dehors de ses extrémités, C'(A) ne touche pas le bord de D. Une construction simple, dépendant du fait que D est tangent ou non en C'(x) et C'(y), permet alors de remplacer D par un disque D' touchant C' en C'(x') et C'(y') tels que |x' - y'| > |x - y|, contredisant ainsi l'hypothèse sur x et y. (voir les détails dans l'article de Tverberg). \Box

Lemme 2.14 Soit C' un polygone de Jordan, et deux points a et b dans la même composante de $\mathbb{R}^2 \setminus C'$. Si la distance de a et de b à C' est supérieure ou égale à 1, et si aucune corde de C' de longueur inférieure à 2, ne sépare a de b, alors il existe un chemin continue Π de a à b tel que $d(\Pi, C') \geq 1$.

PROOF. On commence par remarquer que l'implication du lemme est en fait une équivalence. Il suit que si a et b sont reliés à a' et b' par des chemins distants d'au moins 1 de C', alors les hypothèses sur a et b sont valides pour a' et b'. On peut donc supposer que a et b sont à distance exactement 1 de C'. On considère le disque de rayon 1 centré en a et l'idée est de définir Π comme le lieu du centre de ce disque roulant le long de C' jusqu'à atteindre b. Comme ce disque doit rester intérieur à C', il devra possiblement court-circuiter des arcs de C' et il s'agit de montrer qu'on peut malgré tout atteindre b dans tous les cas. On utilisera pour cela l'hypothèse sur les cordes. (voir les détails dans l'article de Tverberg). \Box

Terminons la preuve du théorème.

Premièrement, $\mathbb{R}^2 \setminus C$ a au moins deux composantes :

Montrons qu'en plus d'une composante non bornée, il en existe une bornée. On considère pour cela un disque D_0 contenant C, et une suite $(C_n)_{n>0}$ de polygones de Jordan contenus dans D_0 et convergeant vers C. Par le premier lemme, chaque C_n contient un disque D_n touchant C_n en deux points $C_n(x_n)$ et $C_n(y_n)$ tels que $|x_n - y_n| > \sqrt{3}$. Soit z_n le centre de D_n , et soit z la limite d'une sous-suite convergente de (z_n) . On va montrer que z ne peut être dans la composante non bornée de $\mathbb{R}^2 \setminus C$ en se ramenant au cas des polygones de Jordan, déjà traité. Par compacité, $|C(x_n) - C(y_n)|$ est borné inférieurement. Par convergence, il en est de même de $|C_n(x_n) - C_n(y_n)|$. Ce qui implique la même chose pour le rayon de D_n et donc pour $d(z_n, C_n)$. Comme $z_n \to z$, z est dans D_n pour n assez grand et donc intérieur à C_n . Si z était hors de D_0 , alors on pourrait trouver un chemin Π le reliant à un point hors de D_0 . Par compacité, $d(\Pi, C) > 0$, donc $d(\Pi, C_n) > 0$ pour n assez grand. Donc z est hors de C_n , une contradiction.

Deuxièmement, $\mathbb{R}^2 \setminus C$ a au plus deux composantes :

Soient p, q et r trois points de $\mathbb{R}^2 \setminus C$. On veut montrer que deux de ces points sont nécessairement dans une même composante. On considère à nouveau une suite $(C_n)_{n>0}$ de polygones de Jordan convergeant vers C. Pour n assez grand p, q et r sont dans $\mathbb{R}^2 \setminus C_n$ (i.e. à une distance non nulle de C_n). Quitte à prendre une sous-suite on peut supposer que p et q sont dans une même composante de $\mathbb{R}^2 \setminus C_n$. Si pour une infinité de n, p peut être relié à q dans $\mathbb{R}^2 \setminus C_n$ par un chemin Π_n tel que $d(\Pi_n, C_n)$ est borné inférieurement, alors il en sera de même pour $d(\Pi_n, C)$ à partir d'un certain rang, ce qui montre que p et q sont dans une même composante de $\mathbb{R}^2 \setminus C$. Sinon, par le second lemme on peut trouver une suite de segments $[C_n(x_n), C_n(y_n)]$ séparant p et q dans $\mathbb{R}^2 \setminus C_n$, et dont la longueur tend vers 0. On en déduit que $|C(x_n) - C(y_n)|$ tend vers 0 et donc que $|x_n - y_n|$ tend vers 0, par continuité uniforme de C^{-1} . Mais alors la plus petite des deux composantes de $\mathbb{R}^2 \setminus C_n \cup [C_n(x_n), C_n(y_n)]$ converge vers un point, impliquant que p ou q est sur C_n , une contradiction.

On a coutume d'appeler une courber fermée simple du plan, un *courbe de Jordan*. Notons qu'en modifiant le deuxièmement de la preuve, on montre sans trop de difficultés que

Théorème 2.15 Un arc simple du plan (i.e. l'image continue injective du segment [0,1]) ne sépare pas le plan.

Une version plus forte du théorème de Jordan stipule que l'intérieur d'une courbe de Jordan est un disque topologique :

Théorème 2.16 (de Jordan-Schönflies) Tout homéomorphisme entre deux courbes de Jordan s'étend au plan tout entier.

Une preuve est donnée dans

The Jordan-Schönflies Theorem and the classification of surfaces. Carsten Thomassen. American Mathematical Monthly. Feb 1992. pp 116-129.

Sans entrer dans les détails, Thomassen commence par considérer une famille dénombrable de points qui est dense dans la première courbe, disons C, et dont chaque point peut être joint à tout point de l'intérieur de C par un chemin polygonal. Il considère également une famille dénombrable de points qui est dense dans l'intérieur de C. On peut alors en déduire une suite où chaque point de ces deux familles apparaît une infinité de fois. On construit ensuite récursivement un homéomorphisme entre l'union de C et des n premiers points de la suite (plus d'autres points autour) et l'union de la seconde courbe, que l'on peut supposer polygonale, et de n points (plus d'autres points autour) qui vont également remplir l'intérieur de cette courbe polygonale. On obtient à la limite une application définie sur C et la suite de points. On montre que cette application se prolonge sur l'adhérence de l'intérieur de C en un homéomorphisme. Pour définir un homéomorphisme sur le plan tout entier, on commence par entourer les deux courbes d'un grand carré, T, que l'on relie aux deux courbes, ce qui permet d'étendre l'homéomorphisme sur ces courbes, en prenant l'identité sur T. La méthode précédente permet d'étendre cet homéomorphisme sur l'intérieur de T que l'on prolonge finalement par l'identité en dehors de T.

À l'aide de ce théorème on montre un analogue général au lemme du thêta et à la relation d'Euler.

2.3 Graphes interdits

Un des résultats les plus célèbres portant sur les graphes planaires est la réciproque du théorème 2.11 dûe au mathématicien Kazimierz Kuratowski (1896 - 1980), stipulant qu'un graphe ne "contenant" ni K_5 ni $K_{3,3}$ est planaire. Les graphes K_5 et $K_{3,3}$ sont ainsi appelés graphes interdits ou graphes de Kuratowski. De manière plus générale on montre, via la théorie des mineurs exclus de Robertson et Seymour, que pour chaque surface de genre g il existe un nombre fini de graphes interdits empêchant un graphe quelconque d'être plongé sur cette surface. On rappelle qu'une subdivision d'un graphe s'obtient en insérant des sommets à l'intérieur de ses arêtes.

Clairement, la présence d'arêtes boucles ou multiples ne modifie pas le caractère planaire d'un graphe. On considère donc dans la suite de ce chapitre des graphes *simples*, i.e. sans arête multiple ni boucle.

Théorème 2.17 (de Kuratowski) Un graphe est planaire si et seulement s'il ne contient pas de subdivision de K_5 ou de $K_{3,3}$ comme sous-graphe.

La preuve qui suit est dûe à Thomassen [MT01].

Rappelons qu'un graphe est 3-connexe s'il a au moins 4 sommets et si la suppression de 2 sommets quelconques ne le déconnecte pas. Par la caractérisation classique de Menger [Wil96, cor. 28.4], un graphe est 3-connexe si et seulement si deux sommets distincts quelconques peuvent être reliés par trois chemins disjoints en dehors de leurs extrémités.

Lemme 2.18 Soit G un graphe 3-connexe ayant au moins 5 sommets. Alors G admet une arête e telle que $G/\!/e$ (i.e. la suppression de e, suivie de l'identification des extrémités de e, suivie de la fusion des éventuelles arêtes multiples en arêtes simples de mêmes extrémités) est 3-connexe.

Preuve : Supposons par l'absurde que pour toute arête e = xy de G, $G/\!/e$ n'est pas 3-connexe. Alors il existe $z, t \in S(G/\!/e)$ (les sommets de $G/\!/e$) qui déconnectent $G/\!/e$, où t résulte nécessairement de l'identification de x et y. Dit autrement, pour toute arête e = xy, il existe $z \in S(G)$ tel que $G - \{x, y, z\}$ n'est pas connexe. On choisit e et zde sorte que la plus grande (en nombre de sommets) des composantes de $G - \{x, y, z\}$ soit maximale. Soit H cette composante et soit u adjacent à z dans une composante de $G - \{x, y, z\}$ autre que H (voir figure 2.3). D'après ce qui précède, il existe $v \in S(G)$ tel



FIGURE 2.3 – Il se peut que $v \in \{x, y\}$ mais dans tous les cas H' a plus de sommets que H.

que $G - \{z, u, v\}$ ne soit pas connexe. Montrons que le sous-graphe H' de G induit par $(S(H) \cup \{x, y\}) \setminus \{v\}$ est connexe. Comme ce sous-graphe est contenu dans une composante de $G - \{z, u, v\}$ et qu'il a plus de sommets que H, on aboutit à une contradiction. Pour montrer la connexité de H' il suffit de vérifier que tout sommet t de H peut être relié à x ou y (eux même reliés par l'arête e) dans H': par la 3-connexité de G, on a l'existence d'un chemin simple $p: t \rightsquigarrow x$ dans G qui évite z et v. Quitte à remplacer x par y on peut supposer que p - x ne contient ni x ni y. Donc p - x est inclus dans $G - \{x, y, z\}$, donc dans H. On en déduit que p est dans H'.

Proposition 2.19 Soit G un graphe 3-connexe ne contenant pas de subdivision de K_5 ou de $K_{3,3}$ comme sous-graphe. Alors G admet un plongement rectiligne convexe (i.e. dont les arêtes sont des segments de droites et dont les faces sont des convexes) dans le plan.

Preuve : Notons que ce lemme implique une version du théorème de Kuratowski restreinte aux graphes 3-connexes. Pour la preuve, on raisonne par récurrence sur le nombre de sommets de G. Le résultat se vérifie à la main si ce nombre vaut 4 ou 5. Par le précédent lemme on peut choisir une arête e = xy telle que G' = G//e est 3-connexe. Clairement G' ne contient pas de subdivision d'un graphe interdit (on vérifie sinon que se serait le cas pour G). Par l'hypothèse de récurrence, G' possède un plongement rectiligne convexe. Soit z le sommet de G' résultant de l'identification de x et y. G' - z est 2-connexe. Par la proposition 2.8, on considère le cycle C de G' - z bordant la face de G' - z contenant z. Soit X (resp. Y) l'ensemble des sommets de rattachement des arêtes de G incidentes à x (resp. y). Les ensembles X et Y ne peuvent se chevaucher dans C (i.e. $|X \cap Y| \leq 2$ et il n'y a pas deux sommets de X et deux sommets de Y apparaissant de manière alternée autour de C). Dans le cas contraire on met en évidence un graphe interdit dans G comme sur la figure 2.4. Ceci permet de construire un plongement convexe pour G en remplaçant z par x dans le plongement de G', en ne conservant que les arêtes $zu, u \in X$, et en insérant y et ses arêtes incidentes dans la face bordée par x et par la sous-chaîne de Ccontenant Y mais pas X. Notons que cette face peut être la face externe du plongement de G' lorsque z est sur le cycle bordant cette face.



FIGURE 2.4 – À gauche, deux sommets de X et deux sommets de Y apparaissant de manière alternée autour de C. On en déduit une subdivision de $K_{3,3}$ dans G. À droite X et Y partagent 3 sommets et on en déduit une subdivision de K_5 dans G.

Exercice 2.20 Soit e une arête de G. Montrer que si $G/\!/e$ contient une subdivision de K_5 ou de $K_{3,3}$, alors c'est également le cas pour G. (Indication : il se peut que $G/\!/e$ et G ne contiennent pas le même graphe interdit).

Le lemme suivant permet de se ramener au précédent dans tous les cas :

Lemme 2.21 Soit G un graphe ne contenant pas de subdivision de K_5 ou de $K_{3,3}$ comme sous-graphe et tel que l'ajout d'une arête entre toute paire de sommets non-adjacents crée un tel sous-graphe. Alors G est 3-connexe.

Preuve : On raisonne par récurrence sur le nombre de sommets de G. Le résultat se vérifie à la main si ce nombre vaut 4 ou 5.

G est 2-connexe :

sinon on peut écrire $G = G_1 \cup G_2$ où G_1 et G_2 n'ont qu'un sommet x en commun. Soit $y_i \in G_i, i = 1, 2$, adjacent à x. L'ajout d'une arête y_1y_2 à G crée par hypothèse une subdivision K d'un graphe interdit. Comme $K_{3,3}$ et K_5 sont 3-connexes et qu'il n'y a que les deux passages x et y_1y_2 entre G_1 et G_2 , les sommets de degré ≥ 3 de K sont tous dans G_1 ou tous dans G_2 . Mais on peut alors remplacer le chemin de K comportant l'arête y_1y_2 et le sommet x par y_1x ou y_2x pour faire apparaître une subdivision d'un graphe interdit dans G. Une contradiction.

Si $G - \{x, y\}$ n'est pas connexe alors xy est une arête de G:

sinon on peut écrire $G = G_1 \cup G_2$ où G_1 et G_2 n'ont que les sommets x et y en commun. L'ajout d'une arête xy à G crée par hypothèse une subdivision K d'un graphe interdit. Comme précédemment les sommets de degré ≥ 3 de K sont tous un même G_i , disons G_1 . Mais on peut alors remplacer l'arête xy dans K par un chemin reliant x et y dans G_2 (qui contient nécessairement un tel chemin) faisant ainsi apparaître une subdivision d'un graphe interdit dans G. Une contradiction.

Supposons que G n'est pas 3-connexe et soient x, y deux sommets qui déconnectent G. On écrit $G = G_1 \cup G_2$ où G_1 et G_2 n'ont que les sommets x et y et l'arête xy en commun. Il est facile de voir que l'ajout d'une arête à G_i (i = 1, 2) crée une subdivision d'un graphe interdit dans ce même G_i . On peut donc appliquer l'hypothèse de récurrence à G_i et par le lemme précédent choisir un plongement convexe de G_i . Soit alors z_i un autre sommet du cycle d'une face F_i bordée par x et y dans ce plongement. L'ajout d'une arête z_1z_2 à G crée une subdivision K d'un graphe interdit. Si tous les sommets de degré ≥ 3 de K sont dans un même G_i alors on peut facilement modifier K pour qu'il se trouve dans G_i , une contradiction. Par ailleurs $S(G_1) - \{x, y\}$ ou $S(G_2) - \{x, y\}$ ne contient qu'un seul sommet de degré ≥ 3 de K. Dans le cas contraire il faudrait au moins 4 chemins disjoints entre G_1 et G_2 dans $G + z_1 z_2$. Pour la même raison K ne peut être qu'une subdivision de $K_{3,3}$. Si G_i contient les 5 autres sommets de degré 3 de K alors on obtient un plongement planaire de $K_{3,3}$ en reliant un point intérieur à F_i aux sommets x, y et z_i . Une contradiction.

Corollaire 2.22 Toute triangulation du plan ayant au moins 4 sommets et dont le cycle externe est également un triangle (autrement dit une triangulation de la sphère) est 3-connexe.

Preuve : Il est facile de voir par la relation d'Euler et par double comptage des incidences face/arête que toute triangulation de la sphère a un nombre maximal possible d'arêtes parmi les graphes planaires possédant le même nombre de sommets. Une telle triangulation est donc 3-connexe par le lemme précédent. □

Preuve du théroème de Kuratowski : D'après le théorème 2.11, un graphe planaire ne peut contenir de subdivision d'un graphe interdit. Réciproquement, soit G un graphe ne contenant pas de subdivision d'un graphe interdit. Si G n'est pas 3-connexe, on peut par le lemme 2.21 lui ajouter des arêtes jusqu'à ce qu'il le devienne, tout en assurant qu'il ne contient pas de graphe interdit. Ce graphe est planaire d'après la proposition 2.19. Il reste évidemment planaire si on supprime les arêtes ajoutées.

Définition 2.23 Un graphe H est un mineur d'un graphe G s'il peut être obtenu à partir de G en supprimant ou contractant des arêtes de G et en supprimant certains sommets. La définition admet quelques variantes selon que l'on travaille avec des graphes simples ou non et connexes ou non.

Une autre formulation du théorème de Kuratowski affirme qu'un graphe est planaire si et seulement si aucun des 2 graphes interdits n'en est un mineur.

On pourra consulter également le chapitre 4 du livre de Diestel pour ce qui précède (en particulier pour la preuve du théorème de Jordan dans le cas polygonal, de la relation d'Euler et du théorème de Kuratowski) :

Graph Theory. Reinhard Diestel. Springer-Verlag, Graduate Texts in Mathematics, Volume 173 July 2005 (2000, 1997).

Une copie électronique est disponible à l'adresse : http://www.math.uni-hamburg.de/home/diestel/books/graph.theory/GraphTheoryIII.pdf

2.4 Critères de planarité

Je liste ici sans preuve un certain nombre de caractérisations classiques des graphes planaires.
Définition 2.24 Un cycle d'un graphe G est dit induit s'il est égal au sous-graphe induit par ses sommets ou, de manière équivalente, s'il n'a pas de corde dans G. Un cycle est dit séparateur si la supression de ses sommets déconnecte G.

Définition 2.25 On appelle cycle facial tout cycle bordant une face d'un plongement, comme dans la proposition 2.8.

Tutte [Tut63] montre que les cycles faciaux des plongements d'un graphe 3-connexe planaire sont en fait caractéristiques du graphe et non de ses plongements. Dit autrement si on réalise un tel graphe sous la forme d'un filet de pêche, il n'y a que deux manières d'habiller la sphère avec ce filet ; elles correspondent aux deux orientations de la sphère.

Théorème 2.26 Soit G un graphe 3-connexe planaire. Un cycle est facial pour un plongement de G si et seulement s'il est induit et non-séparateur dans G.

Définition 2.27 Une 2-base d'un graphe G est une famille de cycles de G qui engendre l'espace des cycles Z(G) et telle que toute arête de G apparaît dans au plus deux cycles de cette famille.

Théorème 2.28 (MacLane, 1936) Un graphe 2-connexe G est planaire si et seulement s'il possède une 2-base. De plus, toute 2-base est formée des cycles faciaux de Gà l'exception d'un cycle, dit externe, et correspond à un plongement planaire de G à homéomorphisme près.

On trouvera une preuve de ce résultat dans le livre de Mohar et Thomassen [MT01, p. 36].

Définition 2.29 Un ordre (partiel) sur un ensemble E est une relation binaire, notée <, sur $E \times E$ qui est transitive, anti-symétrique et anti-reflexive. Si x < y ou y < x on dit que x et y sont comparables. Un ordre est linéaire, ou total, si tous les éléments de E sont comparables. La dimension d'un ordre < est le nombre minimal d'ordres linéaires tel que < est l'intersection de ces ordres.

On peut associer à tout graphe G son *ordre de complexe* défini comme la relation d'inclusion sur la famille de ses sommets et arêtes.

Théorème 2.30 (Schnyder, 1989) Un graphe est planaire si et seulement si la dimension de son ordre de complexe est au plus 3.

Le graphe de contact d'une famille de disques du plan d'intérieurs disjoints a pour ensemble de sommets la famille de disques et pour arêtes les paires de disques en contact (donc tangents).

Théorème 2.31 (Koebe-Andreev-Thurston) Un graphe est planaire si seulement si c'est le graphe de contact d'une famille de disques.

La section 2.8 de [MT01] est consacrée à ce théoreme et ses extensions.

Un 3-polytope est une intersection bornée, et d'intérieur non-vide, de demi-espaces. Les 5 polyèdres de Platon en sont des exemples célèbres. Le graphe d'un 3-polytope est son 1-squelette, c'est à dire le graphe composé des sommets et arêtes du polytope.

Théorème 2.32 (de Steinitz) Un graphe est 3-connexe et planaire si et seulement c'est le graphe d'un 3-polytope.

On trouvera une preuve du théorème de Steinitz dans le livre de Ziegler [Zie95, chap. 4].

2.5 Plongements rectilignes

La proposition 2.19 et le lemme 2.21 montrent en particulier que tout graphe planaire admet un plongement rectiligne. Le problème du plongement rectiligne est à l'origine de nombreuses études comme l'existence de plongements dont les sommets sont à coordonnées entières, comme la recherche de la grille entière minimale pouvant contenir un tel plongement, etc...

Une des preuves les plus anciennes de l'existence d'un plongement rectiligne est attribuée à Fáry [Fár48] (ou à Wagner, 1936). On commence par ajouter des arêtes au graphe planaire G que l'on veut plonger tout en préservant sa planarité, et ce jusqu'à le rendre maximalement planaire (et donc 3-connexe). Tout plongement de G est donc une triangulation. On choisit l'un des triangles, disons t, comme face externe et l'on montre par récurrence sur le nombre de sommets que G admet un plongement rectiligne avec cette face externe fixée. On vérifie d'abord par la formule d'Euler que G a un sommet s de degré au plus 5 distinct des sommets de t. On considère le graphe H obtenu à partir de G - sen ajoutant des arêtes (au plus 2) pour trianguler le "trou" laissé par s. Par hypothèse de récurrence, H admet un plongement rectiligne avec t pour face externe. On supprime alors les (au plus 2) arêtes ajoutées et on insère s dans la face correspondante. Cette face ayant au plus 5 sommets il est facile de montrer qu'elle est étoilée, ce qui permet de positionner s et de le relier aux sommets de la face par des segments de droites intérieurs à cette face. On obtient ainsi un plongement rectiligne de G. Il ne reste plus qu'à supprimer les arêtes initialement ajoutées ayant servi à rendre G maximalement planaire.

Il existe une autre preuve de la proposition 2.19 dûe à Tutte [Tut63] et fournissant un algorithme direct de plongement rectiligne convexe. On rappelle qu'un plongement rectiligne convex est un plongement dont les arêtes sont des segments de droites et dont les faces sont strictement convexes. Par la suite on omettra l'adjectif rectiligne.

Soit G = (S, A) un graphe 3-connexe planaire et soit $\pi(G)$ un plongement polygonal de G. On note C le cycle de G bordant la face externe du plongement (cf. lemme 2.8). On associe à chaque arête $a \in A \setminus A(C)$ un poids strictement positif λ_a . On note $S_I = S \setminus S(C)$ l'ensemble des sommets *internes* du plongement $\pi(G)$.

Théorème 2.33 (Tutte, 1963) Tout plongement strictement convexe de C se prolonge en un unique plongement $\tau : S \hookrightarrow \mathbb{R}^2$ tel que pour chaque sommet $s \in S_I$, $\tau(s)$ soit le barycentre de l'image des voisins V(s) de s dans G avec les poids relatifs λ_{sv} , pour $v \in V(s)$:

$$\forall s \in S_I, \qquad \sum_{v \in V(s)} \lambda_{sv}(\tau(s) - \tau(v)) = 0.$$
(2.1)

De plus, ce plongement induit un plongement convexe de G obtenu en reliant les images de chaque paire de sommets voisins par un segment de droite.

Pour alléger les notations on numérote de 1 à k les sommets de S_I et de k + 1 à n les sommets de C (donc n = |S|). On écrit également λ_{ij} pour le poids de l'arête ij et on note V(i) les indices des voisins du sommet i. On pose enfin $\lambda_{ij} = 0$ si $j \notin V(i)$.

La preuve qui suit est tirée de [CdVPV01] et des notes de cours d'Éric Colin de Verdière http://www.di.ens.fr/~colin/cours/algo-graphs-surfaces.pdf.

Lemme 2.34 Pour tout graphe connexe G, le système (2.1) ayant pour inconnues les $\{\tau(s)\}_{s\in S_I}$ admet une unique solution.

Preuve : On note τ_i l'image du sommet *i* de *S*. Le système (2.1) s'écrit

$$\Lambda \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_k \end{bmatrix} = \begin{bmatrix} \sum_{j>k} \lambda_{1j} \tau_j \\ \vdots \\ \sum_{j>k} \lambda_{kj} \tau_j \end{bmatrix}$$

оù

$$\Lambda = \begin{bmatrix} \sum_{j=1}^{n} \lambda_{1j} & -\lambda_{12} & \dots & -\lambda_{1k} \\ -\lambda_{21} & \sum_{j=1}^{n} \lambda_{2j} & \dots & -\lambda_{2k} \\ \vdots & \vdots & \ddots & \\ -\lambda_{k1} & -\lambda_{k2} & \dots & \sum_{j=1}^{n} \lambda_{kj} \end{bmatrix}$$

Il suffit de vérifier que Λ est inversible.

Soit $x \in \mathbb{R}^k$ tel que $\Lambda x = 0$ et soit x_i une composante de x de valeur absolue maximale. On pose $x_{k+1} = x_{k+2} = \ldots = x_n = 0$. Puisque $(\Lambda x)_i = \sum_{j=1}^n \lambda_{ij}(x_i - x_j) = 0$ et que λ_{ij} est strictement positif pour j voisin de i et nul sinon, on en déduit $x_j = x_i$ pour $j \in V(i)$. Par connexité de G, il suit que tous les x_j , $j = 1, \ldots, n$ sont égaux et donc nuls. On en conclut que Λ est inversible.

On peut interpréter la solution du système (2.1) comme l'état d'équilibre d'un système masse ressort, dont les masses ponctuelles sont les sommets de G. Les sommets de C sont fixés dans le plan tandis que chaque sommet intérieur i est relié à chacun de ses voisins jpar un ressort de raideur λ_{ij} . Par la suite on se réfère à τ comme le plongement (de Tutte) relatif au système (2.1), même si τ n'a pas encore été formellement démontré injectif.

Remarque 2.35 Le plongement de Tutte de chaque sommet de S_I se trouve dans l'intérieur relatif de l'enveloppe convexe du plongement de ses voisins. **Lemme 2.36** Les sommets de S_I solutions du système (2.1) sont contenus dans l'intérieur du plongement convexe de C.

Preuve : Il suffit de montrer qu'aucune droite ne sépare un sommet de $\tau(S_I)$ de l'intérieur de C. Dans le cas contraire on peut choisir une droite ℓ passant par un sommet $\tau(s) \in \tau(S_I)$ et délimitant un demi-plan ouvert ℓ^+ tel que $\ell^+ \cup \ell$ contient $\tau(S)$. Par la remarque 2.35 ci-dessus les plongements des voisins de s sont sur ℓ . Soit H la composante connexe contenant s dans le sous-graphe de G induit par les sommets de S plongés sur ℓ . Puisque $\tau(C)$ est strictement convexe, au plus deux sommets de C se plongent dans ℓ . La suppression de ces au plus deux sommets déconnecte H du sous-graphe de G induit par les sommets de S se plongeant dans ℓ^+ . Comme G est 3-connexe, il suit que $\tau(S) \cap \ell^+$ est vide et donc $\tau(S) \subset \ell$. Mais ceci contredit le fait que $\tau(C)$ est d'intérieur non-vide. \Box

Définition 2.37 Soit h une forme linéaire non nulle. Un sommet de G dont le plongement est aligné avec celui de tous ses voisins dans la direction h = 0 est dit h-passif. Il est dit h-actif dans le cas contraire.

Lemme 2.38 Soit h une forme linéaire non nulle et soit $s \in S_I$ un sommet h-actif. Il existe des chemins M(s,h) et D(s,h) de G ne dépendant que de s et h tels que

- 1. $M(s,h) := v_0, v_1, \ldots v_b$ joint $s = v_0$ à un sommet v_b de C et h est strictement croissante le long du plongement de M(s,h), c'est-à-dire $h(\tau(v_{j+1})) > h(\tau(v_j))$ pour $1 \le j < b$.
- 2. D(s,h) joint s à un sommet de C et h est strictement décroissante le long du plongement de D(s,h).

Preuve : Puisque *s* est *h*-actif il admet par la remarque 2.35 un voisin *v* avec $h(\tau(v)) > h(\tau(s))$. Parmi ces voisins on choisit l'extrémité v_1 d'indice minimal (ce choix est arbitraire). Puisque v_1 est lui-même *h*-actif, on peut itérer le procédé et obtenir un chemin $M(s,h) := v_0, v_1, \ldots v_b$ avec les propriétés voulues, où v_b est le premier sommet de *C* rencontré. Une construction analogue permet d'obtenir D(s,h).

Lemme 2.39 Pour tout sommet intérieur $s \in S_I$ et toute forme linéaire $h \neq 0$, le sommet s est h-actif.

Preuve : Supposons par l'absurde que *s* est *h*-passif. Par le lemme 2.36, je peux choisir un sommet $v \in C$ avec $h(\tau(v)) > h(\tau(s))$. Par la caractérisation de Menger des graphes 3-connexes, je peux choisir trois chemins simples P_i , i = 1, 2, 3, de *s* à *v* et disjoints en dehors de *s* et *v*. Soit Q_i le plus long segment initial de P_i dont le plongement de Tutte est contenu dans la droite $\{h = h(\tau(s))\}$. On note w_i le dernier sommet de Q_i , i = 1, 2, 3. Remarquons que w_i est *h*-actif. Par le lemme précédent on peut choisir deux chemins $P(w_i, h)$ et $D(w_i, h)$ joignant w_i à un sommet de *C*. Un petit dessin indique que $Q_i, P(w_i, h)$ et $D(w_i, h)$ ont deux à deux disjoints en dehors de w_i et que le sous-graphe $C \cup_{i=1,2,3} (Q_i \cup P(w_i, h) \cup D(w_i, h))$ de *G* contient une subdivision de $K_{3,3}$, en contradiction avec le théorème 2.11. Rappelons que par hypothèse G admet un plongement polygonal $\pi(G)$ avec $\pi(C)$ pour face externe. On triangule de manière combinatoire les faces de $\pi(G)$ en ajoutant k-3arêtes dans chaque face interne de $\pi(G)$ de longueur k. On obtient ainsi un nouveau graphe G' ayant les mêmes sommets S que G. Il est possible de modifier les poids des sommets de S pour que la solution du système (2.1) écrit pour G' avec ces nouveaux poids soit la même que la solution du système initial pour G. On utilise pour cela la remarque 2.35 et l'exercice suivant.

Exercice 2.40 Soit s un point contenu dans l'intérieur de l'enveloppe convexe d'un ensemble fini P de points du plan. Montrer que s est barycentre des points de P avec des coefficients strictements positifs (indication : l'enveloppe convexe de $P \cup \{s\}$ est étoilée par rapport à s).

Le plongement rectiligne de G' induit par le système (2.1) fournit un plongement convexe de G solution du système correspondant. En effet, le plongement de G s'obtient en supprimant les arêtes du plongement de G' qui ne sont pas dans G. Les faces du plongement obtenu sont convexes puisque tous les angles intérieurs aux faces sont strictement inférieurs à π par la remarque 2.35 et le lemme 2.39. On peut donc supposer par la suite que G est une triangulation.

Lemme 2.41 Soit uvx et uvy deux triangles faciaux de G (relatifs à $\pi(G)$) partageant l'arête uv. Toute droite $\{h = h(\tau(u))\}$ passant par $\tau(u)$ et $\tau(v)$ sépare strictement $\tau(x)$ et $\tau(y)$. Dit autrement, les plongements de Tutte des triangles uvx et uvy sont non-plats et d'intérieurs disjoints.

Preuve : Dans le cas contraire on peut supposer que le demi-plan $\{h \ge h(\tau(u))\}$ contient $\tau(x)$ et $\tau(y)$. De manière analogue au lemme 2.39 on remarque que le sous-graphe $C \cup uvx \cup uvy \cup P(x,h) \cup P(y,h) \cup D(u,h) \cup D(v,h)$ de G contient une subdivision de $K_{3,3}$, en contradiction avec le théorème 2.11.

Preuve du théorème de Tutte : On considère l'image rectiligne de G induite par la solution du système (2.1). Soit p un point intérieur à $\tau(C)$ contenu dans l'intersection des intérieurs des plongements de deux triangles t et t' de G. On considère une demi-droite d d'origine p ne passant par aucun sommet de $\tau(S)$. la demi-droite d coupe une arête a de $\tau(C)$ incidente à un unique triangle t''. Par le lemme 2.41, on peut définir de manière unique une liste de triangles t'', \ldots traversés en suivant d de a jusque p. Ceci implique t = t'. Il suit que le image induite est bien un plongement rectiligne.

On peut se demander si le théorème de Tutte s'étend à la dimension trois. Colin de Verdière et al. [CdVPV03] proposent une telle généralisation et montrent par un contreexemple qu'elle n'est pas possible.

2.6 Algorithmes pour la planarité

Tarjan [Tar72] explique comment faire un parcours en profondeur en temps O(S + A)d'un graphe G. On obtient un *palm tree* : c'est un arbre enraciné auquel s'ajoute des arcs connectant un sommet à un ancêtre dans l'arbre (arcs retours ou back edge). Inversement tout palmier peut s'obtenir à partir d'un parcours en profondeur. On indexe les sommets dans l'ordre préfixe par leur date de première visite. Pour tout sommet s, on définit L(s)comme le sommet d'indice minimal parmi s lui-même et les têtes des arcs retours ayant pour origine des descendants de s.

On note T un arbre de parcours en profondeur de G, orienté de la racine vers les feuilles.

Lemme 2.42 Si vw est un arc de T et $v \neq 1$ alors $L(w) \geq v$ implique v est un sommet d'articulation (cut vertex) de G.

Preuve : On note T_w le sous-arbre de T de racine w. Alors tout chemin d'origine w ne passant pas par v reste dans T_w : où bien on emprunte une arête de T et on aboutit à un descendant de w (y compris w), ou bien on emprunte une arête retour et on aboutit également à un descendant de w (y compris w) par l'hypothèse $L(w) \ge v$. En particulier v déconnecte T_w des ancêtres de v.

Lemme 2.43 Tout chemin p de G contient un ancêtre (relativement à T) commun à ses extrémités.

Preuve : Soit H le plus petit sous arbre de T contenant les sommets de p et soit u la racine de H. On peut supposer que p contient au moins 2 sommets et considérer un sommet v, enfant de u dans H. On note H_v le sous-arbre de H de racine v. Si $H_v = H - u$ alors u est de degré 1 dans H et u est un sommet de p par minimalité de H. Sinon p contient un arc connectant un sommet de H_v à son complémentaire dans H. Cet arc ne peut être un arc de T. C'est donc un arc retour. Sa tête est nécessairement u qui est l'unique sommet de $H - H_v$ qui soit un ancêtre d'un sommet de H_v . Dans tous les cas, p contient u qui est un ancêtre commun des extrémités de p (et même de tous ses sommets). \Box

Lemme 2.44 Si G est 2-connexe et vw est un arc de l'arbre de recherche en profondeur alors $v \neq 1$ implique L(w) < v et v = 1 implique L(w) = 1.

À TERMINER.

Voir les articles de TGGT 2008. En particulier par Haeupler and Tarjan.

Chapitre 3

Triangulation

Une triangulation d'une région polygonale du plan est une décomposition de cette région en triangles dont les sommets sont ceux du bord de la région. Une triangulation permet souvent de résoudre plus facilement des problèmes portant sur la région qu'elle triangule. Le problème du gardiennage d'une galerie d'art en est un bel exemple.

Au début du XXe siècle N. J. Lennes montre de manière constructive que tout polygone simple admet une triangulation [Len11]. Cette construction fournit de fait un algorithme de complexité quadratique en fonction du nombre de sommets. Du temps de l'émergence de la géométrie algorithmique, Garey et al. (1978) ont proposé un algorithme de complexité $O(n \log n)$ pour trianguler un polygone à n côtés. Après diverses améliorations (cf. notes historiques du livre de [dBCvKO08]), Bernard Chazelle montre en 1991 qu'un polygone simple peut être triangulé en temps linéaire. L'algorithme de Chazelle [Cha91] est réputé très complexe. Une version plus simple et randomisée est décrite par Amato et al. [AGR01].

Le problème de la triangulation de l'intérieur d'un polyèdre dans \mathbb{R}^3 est beaucoup plus compliqué. Contrairement au cas bidimensionnel le nombre de tétraèdre d'une triangulation d'un polyèdre (même convexe) à *n* sommets peut varier suivant la triangulation. De plus, tous les polyèdres ne sont pas triangulables, à moins d'ajouter des sommets intérieurs (dits de Steiner), comme le montre le cas du polyèdre de Schönhardt sur la figure 3.1. Ce polyèdre est obtenu à partir d'un prisme de base triangulaire en tournant légèrement le triangle supérieur par rapport au triangle inférieur. Du coup, les faces verticales du prisme (des quadrilatères) ne sont plus planes et il faut ajouter une diagonale pour trianguler



FIGURE 3.1 – La face supérieure du prisme à base triangulaire de gauche est légèrement pivotée pour obtenir le polyèdre de Schönhardt, à droite.

chacun de ces quadrilatères gauches. En choisissant cette diagonale de manière à rendre les quadrilatères 'concaves', on vérifie que toute nouvelle arête entre deux sommets du polyèdre est extérieure au polyèdre. Il n'est donc pas possible de trianguler son intérieur.

Références :

- Handbook of Discrete and Computational Geometry. Edited by Goodman and O'Rourke. CRC Press 2004.

3.1 Existence

Définitions Une ligne polygonale de sommets (s_1, \ldots, s_n) est la suite de segments $(s_1s_2, \ldots, s_{n-1}s_n)$. Cette ligne polygonale est fermée si $s_1 = s_n$; elle est simple si les sommets sont deux à deux distincts et si les intérieurs de ses segments sont disjoints des sommets et disjoints entre eux. Un polygone est une ligne polygonale simple et fermée. On appelle arêtes les segments d'un polygone. Par le théorème de Jordan (version polygonale), un polygone P sépare le plan en deux régions connexes appelées intérieur et extérieur de P. Dans ce chapitre, on notera respectivement IntP et extP ces régions (ce sont des ouverts du plan).

Une diagonale d'un polygone P est un segment dont l'intérieur relatif (i.e. le segment privé de ses extrémités) est intérieur à P et dont les extrémités sont des sommets de P. Une triangulation de P est un recouvrement de son intérieur (au sens large, i.e. de \overline{IntP}) par des triangles d'intérieurs disjoints et dont les côtés sont soit des arêtes soit des diagonales de P.

Lemme 3.1 Tout polygone ayant au moins 4 sommets admet une diagonale.

Preuve : Soit P un polygone et soit s le sommet de P de coordonnées minimales pour l'ordre lexicographique (s est le sommet le plus bas parmi les sommets les plus à gauche). Soit p le sommet précédant s et q le sommet suivant s pour l'ordre circulaire dans P.

- Si le triangle spq ne contient (au sens large) aucun sommet de $P \setminus \{s, p, q\}$, alors le segment pq est une diagonale : aucune arête de P ne peut rencontrer l'intérieur ni le bord de spq car l'une de ses extrémités serait contenue dans spq. Donc toute demi-droite issue d'un point x intérieur au segment pq et passant par s ne rencontre P qu'une seule fois (en s). Par le théorème de Jordan, et puisque la droite est extérieure à P à l'infini, le point x est intérieur à P, i.e. pq est intérieur à P.
- Sinon, soit r un sommet de P intérieur au triangle spq et qui maximise la distance à la droite pq. On vérifie aisément que le segment sr est une diagonale de P. (cf. figure 3.2).

Exercice 3.2 Compléter les détails manquant de la preuve précédente en indiquant en particulier où intervient l'hypothèse sur le nombre minimal (4) de sommets.



FIGURE 3.2 – Existence d'une diagonale dans un polygone.

Théorème 3.3 Tout polygone admet une triangulation.

Preuve : Soit P un polygone et soit D un ensemble de diagonales de P d'intérieurs disjoints qui soit maximal pour l'inclusion. Toute région bornée du graphe plan $P \cup D$ est nécessairement un triangle; dans la négative le lemme précédent contredirait la maximalité de D.

On montre par récurrence sur n que toute triangulation d'un polygone à n sommets a exactement n-2 triangles et n-3 diagonales.

Exercice 3.4 La preuve de Lennes pour l'existence d'une triangulation est proche de la précédente quoique légèrement différente. Soit spq un triangle et soit R un ensemble de points intérieurs à ce triangle. Montrer qu'il existe $r \in R$ tel que $rsp \cap R = \{r\}$ (ici rsp désigne le bord et l'intérieur du triangle). Compléter la preuve de Lennes.

3.2 Algorithmes

Si un polygone P est décrit sous forme d'une liste doublement chaînée de sommets, la recherche d'une diagonale selon la preuve du lemme 3.1 prend un temps O(n), où n = |P| est le nombre de sommets de P. On en déduit aisément un algorithme de triangulation de complexité quadratique. Nous allons voir deux algorithmes plus efficaces.

Théorème 3.5 Il existe un algorithme qui triangule tout polygone à n sommets en temps $O(n \log n)$ et espace O(n).

Nous proposons ci-dessous deux preuves – c'est-à-dire deux algorithmes – pour ce théorème. Elles sont décrites dans les articles [Cha82, GJPT78].

3.2.1 Algorithme diviser pour régner

L'algorithme de Chazelle consiste à calculer en temps linéaire une diagonale du polygone P qui coupe P en deux sous-polygones de tailles approximativement égales. En appliquant

ce calcul de manière récursive à chacun des deux sous-polygones on obtient un algorithme de triangulation de complexité $O(|P| \log |P|)$.

On supposera que P est décrit sous forme d'une liste cyclique doublement chaînée de ses sommets dans l'ordre (cyclique) le long de P. Par la suite P désignera aussi bien un polygone que sa liste doublement chaînée. On supposera également disposer de la liste L doublement chaînée des sommets de P triés selon l'ordre lexicographique de leurs coordonnées ainsi que de la liste V des paires d'arêtes de P verticalement visibles, i.e. des paires d'arêtes pour lesquelles il existe un segment vertical intérieur à P et dont les extrémités sont respectivement intérieures à chacune de ces deux arêtes. En considérant la taille de la carte des trapèzes (cf. section 10.2), il est facile de voir que la liste V a une taille linéaire en fonction de |P|.

Théorème 3.6 (du polygon cutting, Chazelle 1982) Soit P un polygone à n sommets et soient L et V les listes associées (respectivement des sommets triés selon l'ordre lexicographique des coordonnées et des paires d'arêtes verticalement visibles). Il existe un algorithme de complexité O(n) pour calculer une diagonale de P qui coupe P en deux polygones P_1 et P_2 tels que

$$|P_1|, |P_2| \le \lceil \frac{2}{3}n \rceil + 1.$$

De plus, les listes P_i , L_i et V_i relatives à chacun des deux polygones pour i = 1, 2 peuvent être calculées en temps O(n) également.

On montre dans un premier temps qu'il existe un segment vertical intérieur à P et qui le coupe en deux polygones de tailles approximativement égales.

Lemme 3.7 Soit P un polygone à n sommets. Il existe un segment vertical pq (i.e. $x_p = x_q$) intérieur à P et intersectant P en ses deux extrémités p et q exactement de sorte que les deux composantes de $P \setminus \{p,q\}$ contiennent chacune au plus $\lceil \frac{2}{3}n \rceil$ sommets de P.

Preuve : On considère la décomposition trapézoïdale de P (cf. section 10.2). On suppose dans un premier temps que P est en position générique, c'est à dire que tous les sommets de P ont des abscisses distinctes. Il suit que chaque trapèze de la décomposition intérieur à P est bordé par exactement un sommet de P sur sa gauche et un sommet de P sur sa droite. Soit T le graphe plan obtenu en reliant par un segment chaque paire de sommets incidents à un même trapèze (il y a donc un segment par trapèze). T est connexe (utiliser par exemple la connexité du graphe d'adjacence des trapèzes) et acyclique (utiliser l'acyclicité du graphe d'adjacence des trapèzes), i.e. que T est un arbre. De plus, l'hypothèse de position générique montre que chaque sommet est incident à trois trapèzes au plus et donc que le degré des sommets de T est au plus trois. La figure 3.3 illustre la construction de T.

Remarquons qu'on peut associer à chaque arête a de T un segment vertical intérieur à P et intersectant P en ses extrémités. Le nombre de sommets des deux lignes polygonales de P coupées par ces extrémités est précisément le nombre de sommets de chaque sous arbre de T - a. Le lemme suivant permet donc de conclure.



FIGURE 3.3 – Le segment pq coupe le polygone en deux chaînes de tailles comparables.

Dans le cas non-générique, on perturbe les sommets en tournant de manière infinitésimale le polygone afin de distinguer toutes les abscisses des sommets de P. Cette rotation est elle-même simulée en considérant l'ordre lexicographique sur les paires (abscisses, ordonnées) des coordonnées des sommets.

Lemme 3.8 Soit T un arbre à n sommets, $n \ge 2$. On suppose que chaque sommet de T est de degré au plus 3. Alors il existe une arête a de T telle que chaque composante de T - a possède au plus $\lceil \frac{2n}{3} \rceil$ sommets.

Preuve : Soit *a* une arête de *T*. Si une composante *C* de *T* – *a* est telle que $|C| < \lfloor \frac{n}{3} \rfloor$ alors il existe une arête *b* de *T* telle que

1. ou bien une composante K de T - b vérifie

$$|C| < |K| < \lfloor \frac{n}{3} \rfloor$$

2. ou bien chaque composante de T-b est de taille au moins $\lfloor \frac{n}{3} \rfloor$.

En effet, soit C' la composante complémentaire de C dans T - a et x le sommet incident à C' et a. Par les hypothèses sur T, x est de degré $d \leq 2$ dans C'.

- On ne peut avoir d = 0 car dans ce cas on aurait $|C'| = 1 > \lceil \frac{2n}{3} \rceil$, en contradiction avec $n \ge 2$.
- Si d = 1, on choisit pour b l'unique arête incidente x dans C'. Alors T b a une composante K (de fait C + a) de taille |C| + 1. On se retrouve alors dans le cas 1 ou 2 ci-dessus selon que |C| + 1 est respectivement strictement inférieur ou égal à $\lfloor \frac{n}{3} \rfloor$.
- Sinon d = 2. Soient a_1, a_2 les deux arêtes incidentes à x dans C' et soit C_1 (resp. C_2) la composante de $C' - a_1$ (resp. de $C' - a_2$) qui n'est pas incidente à x. Si $|C_1| > \lceil \frac{2n}{3} \rceil$ alors on se retrouve dans le cas 1 en choisissant $b = a_1$ (et $K = C + a + C_2$). De même en choisissant $b = a_2$ si $|C_2| > \lceil \frac{2n}{3} \rceil$. On peut donc supposer $|C_1| \le \lceil \frac{2n}{3} \rceil$ et $|C_2| \le \lceil \frac{2n}{3} \rceil$. Puisqu'on ne peut avoir à la fois $|C_1| < \lfloor \frac{n}{3} \rfloor$ et $|C_2| < \lfloor \frac{n}{3} \rfloor$, on a $|C_1| \ge \lfloor \frac{n}{3} \rfloor$ ou $|C_2| \ge \lfloor \frac{n}{3} \rfloor$. On se retrouve dans le cas 2 en choisissant $b = a_1$ dans le premier cas et $b = a_2$ dans le second cas.

La preuve du lemme est alors terminée par récurrence sur |C|.

Preuve du théorème 3.6 : On sait d'après le lemme 3.7 qu'il existe une verticale dont les extrémités coupent P en deux lignes polygonales contenant chacune au plus $\lceil \frac{2n}{3} \rceil$ sommets. En parcourant la liste V des paires d'arêtes verticalement visibles on trouvera



FIGURE 3.4 – (Gauche) Le polygone P est coupé en deux chaînes P_c et P_d par la paire (a, b) de segments visibles. (Droite) Une triangulation de la zone Q' bordée par a, b, C'_c et C'_d .

donc nécessairement une paire (a, b) telle que tout segment vertical de visibilité entre les deux arêtes a et b coupe P comme ci-dessus. Fixons un sommet s de P et indexons les sommets de P de 0 à n-1 dans l'ordre direct le long de P à partir de s. À partir de ces indices on peut calculer en temps constant la longueur de la ligne polygonale entre deux sommets d'indices i et j: en incluant les deux sommets et en considérant la ligne de ivers j dans le sens direct, cette longueur vaut j-i+1 si $j \ge i$ et n-j+i-1 sinon. On peut donc tester en temps constant si une paire d'arêtes convient et de ce fait déterminer (a, b) en temps linéaire.

Il n'est en général pas possible de relier deux des extrémités de a et de b par une diagonale car celle-ci peut recouper P. Considérons le quadrilatère Q formé par a, b et les deux segments c et d reliant les extrémités de a et b situés d'un même côté d'une "verticale de visibilité" entre a et b. Le quadrilatère Q forme bien un polygone (simple) de par l'existence d'un segment de visibilité qui sépare c et d. Soit P_c (resp. P_d) la sous-ligne polygonale de P bordés par les extrémités de c (resp. de d) et ne contenant ni a ni b. L'objectif est de calculer une diagonale entre un sommet de P_c intérieur à Q et un sommet de P_d intérieur à Q de sorte que cette diagonale sépare P comme voulu. Pour cela on considère l'ensemble S_c et S_d des composantes de $P \cap IntQ$ qui s'appuient respectivement sur c et d. On pose $C_c = Conv(S_c)$ et $C_d = Conv(S_d)$ (voir figure 3.4, gauche).

Affirmation I : C_c et C_d sont disjointes. De plus, les sommets de C_c (resp. de C_d) sont les sommets de P_c (resp. de P_d).

Preuve de l'affirmation I : Par hypothèse, il est facile de voir – à l'aide du théorème de Jordan – que S_c et S_d sont séparées dans Q par un segment vertical. Il en est donc de même de leurs enveloppes convexes. Par ailleurs, toujours à l'aide du théorème de Jordan, on montre que toute région bordée par une composante de $S_c \cap P_d$ et un segment de c est contenue dans une région bordée par une composante de $S_c \cap P_c$ et un segment de c. On en déduit que $C_c = Conv(S_c \cap P_c)$. Donc C_c est l'enveloppe convexe des sommets de $S_c \cap P_c$ qui comprend les sommets de P_c inclus dans S_c et les extrémités des composantes de S_c . Or ces sommets sont tous sur c donc dans l'enveloppe convexe des extrémités de c (ce sont des sommets de P_c). Un raisonnement analogue montre que les sommets de C_d sont des sommets de P_d .

L'affirmation précédente permet de sélectionner en temps O(n) un sous-ensemble A des sommets de P tel que $C_c = Conv(A)$: il suffit de parcourir P_c et de retenir les sommets de P_c compris entre deux intersections successives de P_c avec c, lorsque P_c entre dans Q à la première intersection. On peut extraire de L la sous-liste L_A , triée selon l'ordre lexicographique des coordonnées, des sommets de A. On obtient finalement C_c en temps linéaire à partir de L_A par l'algorithme classique de balayage 8.1.2.¹ De manière analogue on calcule C_d en temps linéaire.

On considère maintenant le polygone Q' formé des arêtes a et b et des deux chaînes concaves $C'_c = C_c - c$ et $C'_d = C_d - d$. Notons que Q' est bien une ligne polygonale simple d'après l'affirmation I. Notre but est de montrer que dans toute triangulation \mathcal{T} de Q'l'une des arêtes de \mathcal{T} fournit une diagonale qui sépare P comme voulu. On remarque tout d'abord que tout triangle de \mathcal{T} contient nécessairement une arête de C'_c ou bien de C'_d . En effet, C'_c et C'_d étant concaves, un tel triangle ne peut avoir deux sommets non adjacents sur une même de ces deux chaînes (cf. figure 3.4, droite). Le dual de \mathcal{T} est donc une chaîne simple, ce qui permet d'ordonner les diagonales de \mathcal{T} de la première, incidente au même triangle que a, à la dernière, incidente au même triangle que b. On note $\delta_1, \delta_2, \ldots \delta_k$ ces diagonales, on pose $\delta_{k+1} = b$ et pour $i = 1, \ldots, k - 1$, on note γ_i la troisième arête du triangle de \mathcal{T} bordé par δ_i et δ_{i+1} . Donc γ_i est une arête de C'_c ou de C'_d et on note Γ_i la sous-chaîne de respectivement P_c ou P_d joignant les extrémités de γ_i . Pour chaque diagonale δ_i de \mathcal{T} , on note enfin Δ_i la sous-chaîne de P contenant a et joignant les extrémités de δ_i .

Affirmation II : Une des arêtes de \mathcal{T} (soit une diagonale soit une arête de Q') est une diagonale de P qui coupe P en deux lignes polygonales (extrémités incluses) de taille au plus $\lceil \frac{2}{3}n \rceil + 1$.

Preuve de l'affirmation II : Supposons qu'une arête γ_i de C'_c ou de C'_d ne soit pas une arête de P (et soit donc une diagonale de P) et que

$$|\Gamma_i| \ge \lfloor \frac{n}{3} \rfloor + 1.$$

On a alors, en notant Γ'_i l'autre chaîne de P joignant les extrémités de γ_i

- d'une part : $|\Gamma_i| \leq \max\{|P_c|, |P_d|\} \leq \lceil \frac{2n}{3} \rceil$,

- d'autre part : $|\Gamma_i| + |\Gamma'_i| = n + 2$, d'où $|\Gamma'_i| \leq \lfloor \frac{2}{3}n \rfloor + 1$.

L'affirmation est donc vérifiée en choisissant γ_i comme diagonale de P.

Supposons maintenant à l'inverse que pour toute arête γ_i de C'_c ou de C'_d on ait

$$|\Gamma_i| \le \lfloor \frac{n}{3} \rfloor.$$

On a en particulier

$$|\Delta_1| = |\Gamma_1 + a| \le \lfloor \frac{n}{3} \rfloor + 1.$$

^{1.} Chazelle utilise un autre argument. Il extrait de S_c une ligne polygonale simple joignant les extrémités de c et dont l'enveloppe convexe est C_c Cette ligne est constituée de composantes de S_c et de segments de c. Il utilise ensuite l'algorithme de complexité linéaire pour calculer l'enveloppe convexe d'une ligne polygonale simple.

Par ailleurs si pour un certain $i \in [1, k]$ on a $|\Delta_i| \leq \lfloor \frac{n}{3} \rfloor$ alors δ_i n'est pas la dernière diagonale de \mathcal{T} (i.e. i < k) et

$$|\Delta_i| < |\Delta_{i+1}| \le \lceil \frac{2}{3}n \rceil + 1.$$

En effet, on a

$$|\Delta_{i+1}| = |\Delta_i + \Gamma_i| = |\Delta_i| + |\Gamma_i| - 1 \le 2\lfloor \frac{n}{3} \rfloor - 1.$$

Alors que $|\Delta_k| \ge \lceil \frac{2}{3}n \rceil + 1$ (faire un raisonnement analogue à la majoration de $|\Delta_1|$). On conclut par récurrence sur k que l'une au moins des diagonales δ_i vérifie

$$\lfloor \frac{n}{3} \rfloor + 1 \le |\Delta_i| \le \lceil \frac{2}{3}n \rceil + 1$$

Ceci permet également de confirmer l'affirmation en choisissant δ_i comme diagonale de P.

Affirmation III : Q' peut être triangulé en temps linéaire.

Preuve de l'affirmation III : D'après la preuve de l'affirmation II, on peut trianguler Q' de manière incrémentale en calculant chaque diagonale δ_{i+1} en fonction de la diagonale δ_i calculée précédemment : en notant p_c et p_d les extrémités de δ_i et p_cq_c et p_dq_d les arêtes de Q' respectivement incidentes à p_c et p_d et "au dessus" de δ_i , alors on a soit $\delta_{i+1} = p_cq_d$ soit $\delta_{i+1} = p_dq_c$. Il suffit de tester si q_d (resp. p_c) est au dessous de la droite p_cq_c (resp. p_dq_d) pour savoir si p_cq_d (resp. p_dq_c) est une diagonale. Il se peut que les deux le soient, auquel cas l'une ou l'autre convient puisque dans les deux cas on se retrouve dans une configuration où la partie de Q' au dessus de δ_{i+1} est constituée de deux chaînes concaves reliées par deux segments, ce qui permet d'appliquer la récurrence.

La conjonction des affirmations II et III permet de conclure la première partie du théorème. Il reste à vérifier, en appelant P_1 et P_2 les deux polygones coupés par la diagonale δ trouvée, que les listes P_i , L_i et V_i relatives au polygone P_i pour i = 1, 2 peuvent être calculées en temps O(n) également. C'est clair pour les listes P_i et L_i (ces listes contiennent plus précisément des pointeurs bidirectionnels sur un tableau des sommets fixé une fois pour toute. On peut associer à chacun des sommets du tableau un drapeau qui permet de sélectionner dans une première passe les sommets qui nous intéressent). Pour la liste V_i il suffit de remarquer qu'elle est constituée d'une part des paires d'arêtes de V qui sont dans P_i et d'autre part des paires (a, δ) pour chaque paire (a, b) de V dont la visibilité est obstruée par δ et telle que a est une arête de P_i (et donc b n'est pas une arête de P_i). En parcourant V, on peut ainsi construire V_i de la manière suivante. Pour chaque paire (a, b) de V:

- 1. si $a \in P_i$ et $b \in P_i$, alors on place (a, b) dans V_i ,
- 2. sinon, si a et b ne sont pas dans le même polygone, disons $a \in P_i$, $b \notin P_i$, et si les projections verticales sur l'axe des abscisses des arêtes a, b et δ ont une intersection non vide, alors on place (a, δ) dans V_i après avoir vérifié que cette paire n'était pas déjà présente dans V_i . Cette dernière vérification s'obtient en temps constant en marquant au fur et à mesure les arêtes a de P telles que (a, δ) est dans V_i .

50

Preuve du théorème 3.5 : Soit P donc un polygone à n sommets. L'algorithme consiste à appliquer récursivement le théorème du polygon cutting : la triangulation de P est l'union des triangulations des polygones P_1 et P_2 obtenus par le théorème 3.6.

Pour initialiser l'algorithme il faut construire la liste L des sommets de P triés selon l'ordre lexicographique de leurs coordonnées; ce qui prend un temps $O(n \log n)$ et une place linéaire suivant tout algorithme de tri standard. Il faut construire également la liste V des paires d'arêtes de P verticalement visibles que l'on obtient en temps linéaire en parcourant la carte des trapèzes de P (cf. section 10.2). Cette carte a elle-même une taille linéaire (lemme 10.2) et peut être construite en temps $O(n \log n)$ par un algorithme randomisé (cf. section 10.2) ou non (cf. section 10.1).

Soit C(n) la complexité maximale de la triangulation de tout polygone de taille n. On peut écrire

$$C(n) \le kn + \max_{\substack{n_1+n_2=n+2\\n_1,n_2 \le \lceil \frac{2}{3}n \rceil + 1}} \{C(n_1) + C(n_2)\}$$

pour un certain k > 0. Montrons que $C(n) = O(n \log n)$. Soit α tel que $2/3 < \alpha < 1$. Choisissons N assez grand pour que $n > N \implies \lceil \frac{2}{3}n \rceil + 1 < \alpha n$ et $\frac{n}{2} \log \frac{1}{\alpha} > 2 \log n$. Choisissons ensuite K suffisamment grand pour que $n \le N \implies C(n) \le Kn \log n$ et pour que $K \log \frac{1}{\alpha} > 2k$.

Pour $n \leq N$ on a donc par hypothèse $C(n) \leq Kn \log n$. Supposons par récurrence $C(m) \leq Km \log m$ pour m inférieur à un certain n > N. Pour $n_1, n_2 \leq \lceil \frac{2}{3}n \rceil + 1$ tels que $n_1 + n_2 = n + 2$ on a

$$kn + C(n_1) + C(n_2) \leq kn + Kn_1 \log n_1 + Kn_2 \log n_2 \leq kn + K(n+2) \log(\alpha n)$$

$$\leq Kn \log n + 2K \log n + (k - K \log \frac{1}{\alpha})n$$

$$\leq Kn \log n + K(2 \log n - \frac{n}{2} \log \frac{1}{\alpha}) \leq Kn \log n$$

Ce qui permet de conclure $C(n) \leq Kn \log n$.

3.2.2 Algorithme par décomposition en polygones monotones

L'algorithme de triangulation de Garey et al. se compose de deux étapes. Dans un premier temps le polygone à trianguler est décomposé en polygones plus simples appelés polygones monotones. Cette étape prend un temps $O(n \log n)$. Ces polygones monotones sont ensuite triangulés en temps linéaire selon une technique appropriée. Au total on obtient donc une complexité équivalente à l'algorithme diviser pour régner de Chazelle.

Polygones monotones

Définition 3.9 On considère une direction du plan qu'on appelle verticale. La direction orthogonale est dite horizontale. La hauteur d'un point est sa projection horizontale sur



FIGURE 3.5 – (A) Une ligne polygonale non monotone. (B) Une ligne polygonale strictement monotone. (C) Un polygone monotone, mais pas strictement. Les sommets p q sont des extrema stricts, r et s sont des extrema non stricts, t est un minimum strict et u n'est pas un extremum. Les sommets p, r et u sont réflexes.

la verticale. Une ligne polygonale L est dite (strictement) monotone si la hauteur de la séquence des sommets de L est (strictement) monotone. Dit autrement L est monotone si toute droite horizontale intersecte L en au plus une composante, réduite à un point dans le cas strict.

Un polygone P est dit (strictement) monotone s'il est la réunion de deux lignes polygonales (strictement) monotones ayant seulement leurs extrémités en commun. Dit autrement un polygone P est monotone (resp. strictement monotone) si toute droite horizontale coupe P en au plus deux composantes (resp. au plus deux points).

Un sommet intérieur (i.e. qui n'est pas une extrémité) à une ligne polygonale ou à un polygone est dit maximum, (resp. minimum) (strict) si ces deux sommets voisins sont (strictement) en dessous (resp. au dessus) de la droite horizontale passant par ce sommet. On appelle extremum (strict) un sommet qui est soit maximum (strict) soit minimum (strict). La figure 3.5 illustre les différentes définition.

On vérifie aisément qu'un sommet n'est pas un extremum si et seulement si sa hauteur est strictement comprise entre celles de ses deux sommets voisins. Par suite :

Lemme 3.10

- une ligne polygonale ayant au moins 3 sommets est strictement monotone si et seulement si aucun de ses sommets intérieurs n'est extremum.
- Un polygone est strictement monotone si et seulement si il a exactement deux extrema.

Définition 3.11 Un sommet d'un polygone est dit réflexe si l'angle intérieur au polygone formé par les deux arêtes incidentes au sommet est strictement plus grand que π . Une sous-chaîne d'un polygone est dite concave si ses sommets intérieurs sont réflexes.

Lemme 3.12 Un polygone sans extremum réflexe est monotone.

Preuve : Soit P un polygone sans extremum réflexe. Soient p et q des sommets de P de hauteur respectivement minimale et maximale. Les sommets p et q coupent P en deux



FIGURE 3.6 – (A) Le polygone P est supposé sans extremum réflexe (B) Une composante C de $P_G \cap h^+$. (C) L'extrémité v de C sépare C de q le long de P_G ; le sommet y est un minimum réflexe.

lignes polygonales P_G et P_D telles que P_G est à gauche de P_L . Supposons par l'absurde que P n'est pas monotone. Alors par définition, P_G ou P_D n'est pas monotone. On suppose sans perte de généralité que P_G n'est pas monotone. Il existe donc une droite horizontale h coupant P_G en deux composantes au moins. On note h^+ et h^- les demi-plans ouverts respectivement au dessus et au dessous de h. Par connexité de P_G , l'une des composantes, C, de $h^+ \cap P_G$ ou de $h^- \cap P_G$ a ses deux extrémités dans h. On note u et v ces deux extrémités avec u à gauche de v. Supposons à nouveau sans perte de généralité $C \subset h^+$ comme sur la figure 3.6.A.

J'affirme que

(A) le long de C, l'intérieur de P est situé du même côté que l'intérieur du polygone délimité par C et le segment uv de h.

Par ailleurs, puisque P_G est à gauche de P_D , l'intérieur de P est à droite de P_G lorsque P_G est parcourue du bas vers le haut (i.e. de p vers q). Par conséquent, en admettant (A), le sommet u est avant v dans ce parcours. Soit D la composante de $P_G \setminus C$ joignant v à q. J'affirme que

(B) le sommet le plus à gauche parmi les sommets de hauteur minimale de D est extremum réflexe (cf. figure 3.6.C).

Cette dernière contradiction permet de conclure la monotonicité de P_G et donc de P. Il reste à montrer les affirmations (A) et (B).

Pour (A), on considère le sommet x de C le plus à droite parmi les sommets de hauteur maximale. Les directions des deux arêtes d'origine x et la direction horizontale \vec{h} vers la droite sont donc deux à deux distinctes. Soit a l'arête issue de x dont la direction suit celle de \vec{h} dans le sens indirect (le sens des aiguilles d'une montre). On note b la seconde arête issue de x. Comme x est extremum, il ne peut être réflexe, ce qui montre que l'intérieur de P est entre b et a dans le sens direct, ou encore à droite de a. Montrons que c'est également le cas pour l'intérieur de la courbe de Jordan $uv \cup C$. Pour cela, on note C_a et C_b les deux composantes de $C \setminus \{x\}$ contenant respectivement a et b et on note $w \in \{u, v\}$ l'extrémité de C_a autre que x (cf. figure 3.6.B). Il est clair qu'on ne peut avoir w = u,



FIGURE 3.7 – Le polygone P est décomposé en polygones monotones par ajout de diagonales incidentes à ses extrema réflexes.

sinon la courbe simple S formée de C_a , de la demi-droite horizontale à droite de x et de la demi-droite horizontale à gauche de w formerait une courbe de Jordan qui ne rencontre pas C_b . Or l'intérieur de b est au dessus de S et l'extrémité v de C_b est au dessous de S, ce qui contredit la connexité de C_b . Donc w = v. Comme l'intérieur de $uv \cup C$ est au dessus et donc à droite de vu, il en est de même pour C_a , i.e. l'intérieur de $uv \cup C$ est à droite de a.

Un raisonnement analogue permet de montrer que le sommet spécifié dans (B) qui est évidemment extremum est également réflexe. \Box

Décomposition en polygones monotones

Théorème 3.13 Il existe un algorithme de complexité $O(n \log n)$ pour décomposer tout polygone à n sommets, par l'ajout de diagonales au polygone, en polygones monotones.

Preuve : Considérons la décomposition trapézoïdale d'un polygone P obtenue par cloisonnement horizontal (cf. section 10.2). On suppose le polygone en position générale, i.e. deux sommets distincts ont des ordonnées distinctes. Chaque trapèze intérieur à P est donc incident à exactement deux sommets de P, un sommet supérieur sur le côté horizontal supérieur du trapèze et un sommet inférieur sur le côté horizontal inférieur du trapèze. On ajoute une diagonale joignant ces deux sommets si le sommet supérieur est un minimum réflexe ou si le sommet inférieur est un maximum réflexe. On obtient ainsi une décomposition de P en sous-polygones comme sur la figure 3.7. Aucun sous-polygone de la décomposition ne peut avoir d'extremum réflexe : un sommet p qui serait extremum réflexe dans un sous-polygone le serait nécessairement dans P; mais les règles d'ajout de diagonales font que p n'est plus extremum dans aucun des sous-polygones incidents. Il suit du lemme 3.12 que ces polygones sont tous monotones. Le cas non générique où plusieurs sommets peuvent posséder une même ordonnée se traite en simulant une rotation infinitésimale en considérant l'ordre lexicographique sur les paires (ordonnée, abscisse).

Notons que la décomposition trapézoïdale peut s'obtenir en temps $O(n \log n)$ et que l'ajout de chaque diagonale s'obtient en temps constant par diagonale (en utilisant une structure de carte planaire en demi-arêtes), ce qui achève la démonstration.

Exercice 3.14 Décrire un algorithme de complexité $O(n \log n)$ pour la décomposition d'un polygone P en polygones monotones qui n'utilise pas à proprement parler la décomposition trapézoïdale de P, mais seulement un balayage des sommets de P.

Triangulation des polygones monotones

Théorème 3.15 Il existe un algorithme de complexité linéaire pour trianguler un polygone monotone.

Preuve : Soit P un polygone strictement monotone. En temps linéaire on peut couper P en deux chaînes monotones. Les sommets sur chaque chaîne sont naturellement triés selon leur ordonnée. La fusion (en temps linéaire) de ces deux listes permet d'obtenir la liste V des sommets de P triés selon leur ordonnée. On effectue un balayage des sommets de haut en bas. Au cours du balayage une partie de l'intérieur de P est triangulée et une autre P_i forme un polygone strictement monotone que l'on doit trianguler. Soit σ_i le sommet maximum de P_i et soient G_i et D_i les deux chaînes monotones maximales de P_i . On stocke les sommets balayés dans une pile Π de manière à conserver l'invariant suivant : (i) les sommets dans Π forment une sous-chaîne concave de sommet ν_i plus bas que les sommets de Π . En particulier, si les sommets de Π sont dans G_i (resp. D_i) alors ν_i est dans D_i (resp. G_i). Au départ Π est initialisée avec les deux premiers sommets de V (i.e. le sommet maximal de $P_0 = P$ et le sommet juste au dessous).

Soit s le nouveau sommet balayé dans la liste V.

- 1. Si s forme une chaîne concave avec Π (i.e. le dernier sommet de Π est réflexe) alors on empile s. Les invariants (i) et (ii) sont maintenus.
- 2. Si $s = \nu_i$ et si s n'est pas le sommet minimal de P alors on relie s par des segments à chacun des sommets de Π , hormis σ_i (qui est déjà relié à s). Notons que ces segments sont des diagonales de P_i (et donc de P) car aucune arête entre deux sommets de Π ne peut couper une telle diagonale (par concavité de Π) ni aucune autre arête de P_i par monotonie de Π . On a ainsi triangulé une partie supérieure de P_i . Le reste constitue le polygone P_{i+1} . On vide ensuite Π et on ré-insert son dernier sommet, qui devient le sommet σ_{i+1} , puis le sommet s; ces deux sommets formant les deux plus hauts sommets de P_{i+1} . On définit également ν_{i+1} comme le sommet suivant Π le long de P_i . Clairement les invariants (i) et (ii) sont rétablis.
- 3. Si s est incident au dernier sommet s_{ℓ} de Π mais ne forme pas une chaîne concave avec Π (i.e. s_{ℓ} est convexe dans P_i) alors on relie s par des segments aux derniers sommets $s_k, s_{k+1}, \ldots, s_{\ell-1}$ de Π , hormis le tout dernier s_{ℓ} auquel il est déjà relié, de sorte que la droite ss_k est support pour Π (i.e. les sommets de Π sont d'un même côté de cette droite) mais que la droite ss_{k+1} ne l'est pas. À nouveau ces segments sont des diagonales de P_i car aucune des deux chaînes G_i et D_i ne peut recouper



FIGURE 3.8 – (a-i) Triangulation par balayage vertical des sommets de $P = P_0$. On retrouve la situation 1 aux étapes a,b,f, la situation 2 aux étapes c, h, i et la situation 3 aux étapes e et g.

le segment ss_k du fait de leur monotonie. On a ainsi triangulé une partie de P_i . Le reste constitue le polygone P_{i+1} . On dépile alors les sommets s_{k+1}, \ldots, s_ℓ de Π et insert le sommet s. Le fait que la droite ss_k soit support de Π montre que Π contient bien une chaîne concave et que les invariants (i) et (ii) sont rétablis.

Lorsqu'on balaye le sommet minimal de P on se retrouve dans la situation 2 ci-dessus et la triangulation qui suit achève la triangulation de P. La figure 3.8 résume les différentes étapes sur un exemple. Pour chaque sommet balayé les opérations effectuées ci-dessus se décomposent dans chacun des trois cas en un nombre constant d'opérations élémentaires auquel s'ajoute un nombre d'opérations proportionnel au nombre de diagonales ajoutées à la triangulation. Comme il y a un nombre linéaire de diagonales et que l'on balaye un nombre linéaire de sommets, le coût total de la triangulation est linéaire.

3.2.3 Application au problème de la galerie d'art

Le problème communément attribué à Victor Klee en 1973 est le suivant : étant donné une galerie d'art dont le sol a la forme d'un polygone, combien de gardiens (ou caméras) fixes suffisent à garder la galerie ? On sous-entend que chaque gardien peut regarder dans toutes les directions autour de lui. De manière plus géométrique, soit P un polygone du plan et $x \in \overline{IntP}$ un point intérieur (au sens large) à P. La zone de visibilité de x dans P est l'ensemble des points intérieurs à P et visibles depuis x dans P. C'est encore

$$V_P(x) = \{ y \in \overline{IntP} \, | \, xy \subset \overline{IntP} \}$$

où xy dénote le segment joignant x et y. Un ensemble $X \subset P$ couvre P si

$$P = \bigcup_{x \in X} V_P(x)$$

i.e. si l'union des zones de visibilités des points de X recouvre P. Le problème de la galerie d'art revient donc à chercher un ensemble X de taille minimale couvrant P. Le problème de Klee était plus précisément de trouver la taille minimale g(n) telle que *tout* polygone à n sommets est couvert par un ensemble de taille g(n).

Théorème 3.16 (de la galerie d'art, Chvátal 1973)

$$g(n) = \lfloor n/3 \rfloor.$$

La preuve suivante due à Fisk en 1978 utilise la notion de coloriage. Un *coloriage* d'un ensemble E par une ensemble C est une application de E dans C. Si le cardinal de C est k on parle d'un k-coloriage de E.

Preuve du théorème : $g(n) \leq \lfloor n/3 \rfloor$: soit P un polygone à n sommets et \mathcal{T} une triangulation de P. L'algorithme suivant construit un 3-coloriage des sommets de P tel que tout triangle de \mathcal{T} est tricolore (ses trois sommets ont des couleurs 2 à 2 distinctes) :

Choisir une arête a de P et colorier ses deux sommets, l'un en bleu et l'autre en blanc. Cette arête est incidente à un unique triangle de \mathcal{T} , ce qui détermine la couleur de son troisième sommet, disons rouge. Si l'arête (rouge, blanc) de ce triangle est une diagonale, colorier récursivement la partie de P coupée par cette diagonale et ne contenant pas a. Faire de même avec l'arête (rouge, bleu).

Remarquons alors que l'ensemble des sommets de P ayant la couleur la moins fréquente dans un tel 3-coloriage est de cardinal au plus $\lfloor n/3 \rfloor$ et couvre P, puisque couvre tout triangle de \mathcal{T} .

 $g(n) \geq \lfloor n/3 \rfloor$: Pour tout k, on construit un polygone en forme de peigne de taille 3k, ayant k dents, qui ne peut être couvert par moins de k points. Considérons pour cela un triangle t ayant un côté horizontal de longueur 1 et k copies t_1, t_2, \ldots, t_k de t successivement translatées horizontalement de 2 unités. Ces copies sont donc disjointes et forment les dents du peigne. On considère également un trapèze dont les bases sont horizontales : l'une joint un sommet du côté horizontal de t_1 avec un sommet du côté horizontal de t_1 avec un sommet du côté non horizontal de t_1 avec un point intérieur à un côté non horizontal de t_1 avec un point intérieur à un côté non horizontal de t_k . Notre peigne est le bord de l'union de ce trapèze avec les k triangles t_1, t_2, \ldots, t_k . Puisque les triangles sont disjoints, aucun point du peigne ne peut couvrir simultanément les deux pointes de deux dents du peigne. Il faut donc au minimum k points pour couvrir le peigne.

L'algorithme de coloriage de la preuve du théorème est clairement de complexité linéaire. Compte tenu de l'existence d'un algorithme de triangulation de complexité linéaire [Cha91], ceci permet de trouver en temps linéaire, pour un polygone donné, un ensemble couvrant de taille minimale dans le cas le pire. Par contraste, trouver la taille minimale de tout ensemble couvrant pour un polygone précis est un problème NP-difficile [Agg84].

Références :

- http://valis.cs.uiuc.edu/~sariel/teach/2004/b/webpage/lec/23_triang.pdf

http://valis.cs.uiuc.edu/~sariel/teach/2004/b/webpage/lec/24_triang_II.pdf
On pourra consulter l'État de l'art

Art Gallery and Illumination Problems. J. Urratia. Chap. 22 in "Handbook of Computational Geometry". Edited by J. R. Sack and J. Urrutia

Chapitre 4

Plus court chemin dans un polygone

On s'intéresse ici au calcul d'un plus court chemin entre deux points dans un environnement polygonal, c'est-à-dire dans le plan privé de l'intérieur d'une union finie de polygones. On note \mathcal{O} l'ensemble des polygones – ou obstacles – de l'union. On note également $S(\mathcal{O})$ l'ensemble des sommets des polygones de \mathcal{O} . L'espace libre \mathcal{E} désigne le plan privé de l'intérieur de $\cup \mathcal{O}$. On considère donc que \mathcal{E} est un fermé du plan.

Lemme 4.1 Soit x, y deux points de l'espace libre, et soit $\gamma : [0,1] \to \mathcal{E}$ un chemin rectifiable joignant x à y. Il existe un chemin polygonal p entre x et y dans \mathcal{E} dont les sommets, hormis x et y, sont dans $S(\mathcal{O})$ et de longueur inférieure à celle de γ . Si de plus γ n'est pas une courbe polygonale ou si γ n'a pas ses sommets dans $S(\mathcal{O}) \cup \{x, y\}$ alors on peut choisir p strictement plus court que γ .

Preuve : On peut supposer que l'espace libre est réduit à un rectangle contenant x, y et $\cup \mathcal{O}$. Soit \mathcal{T} une triangulation de l'espace libre. L'image réciproque par γ de tout triangle fermé $t \in \mathcal{T}$ est un compact de [0,1]. Soit u, v les bornes respectivement inférieure et supérieure de $\gamma^{-1}(t)$. Le chemin obtenu en remplaçant la restriction de γ à [u, v] par le segment $[\gamma(u), \gamma(v)]$ n'est pas plus long que γ . Par récurrence sur le nombre de triangles de \mathcal{T} , on peut donc supposer que l'intersection de γ avec chaque triangle est un segment et donc que γ est polygonale (cf. figure 4.1). Si un sommet s de γ n'est pas dans $S(\mathcal{O}) \cup \{x, y\}$ alors s est intérieur à \mathcal{E} ou à une arête de \mathcal{O} . On vérifie dans les deux cas que l'on peut court-circuiter s pour raccourcir γ strictement.

Corollaire 4.2 Deux points quelconques x, y de l'espace libre sont reliés par au moins un chemin de longueur minimale dans \mathcal{E} . Un tel chemin est nécessairement simple et polygonal et ses sommets sont dans $S(\mathcal{O}) \cup \{x, y\}$.

Preuve : Remarquons qu'un chemin qui s'auto-intersecte contient un chemin strictement plus court. Un plus court chemin est donc nécessairement simple. Il y a un nombre fini de chemins polygonaux simples dont les sommets sont dans $S(\mathcal{O}) \cup \{x, y\}$. Le lemme précédent indique que, parmi ceux-ci, ceux qui minimisent la longueur sont des plus courts chemins.



FIGURE 4.1 – (Gauche) Un chemin γ entre deux points x et y de l'espace libre \mathcal{E} . (Milieu) Triangulation \mathcal{T} de l'espace libre. On peut supposer que la restriction de γ à tout triangle $t \in \mathcal{T}$ est un segment. (Droite) Un plus court chemin et un chemin polygonal non-optimal entre x et y.



FIGURE 4.2 – (Gauche) Deux chemins polygonaux entre x et y dans l'espace libre \mathcal{E} intérieur à P. (Droite) L'un de ces chemins possède un sommet convexe qui peut être court-circuité et n'est donc pas optimal.

On se restreint désormais au cas où l'espace libre \mathcal{E} est délimité par une courbe polygonale fermée simple P. En particulier l'espace libre est homéomorphe à un disque fermé et le dual de toute triangulation (cf. chapitre 3) de l'espace libre est un arbre (exercice).

Lemme 4.3 Deux points quelconques x, y de \mathcal{E} sont reliés par un unique plus court chemin dans \mathcal{E} . Un tel chemin est caractérisé par le fait d'être polygonal et d'avoir des sommets internes qui sont des sommets de P et qui sont concaves (d'angles intérieurs au moins égaux à π).

Preuve : Remarquons qu'il existe au moins un plus court chemin par le corollaire 4.2. De plus, un plus court chemin est nécessairement simple, polygonal et ses sommets sont dans P.

Supposons par l'absurde que p et q sont deux plus courts chemins distincts entre x et y comme sur la figure 4.2. On peut donc trouver deux sous-chemins polygonaux p' et q' de p et q respectivement, ayant mêmes extrémités et disjoints par ailleurs. Leur union $p' \cup q'$ est une courbe fermée simple et, par le théorème de Jordan, l'intérieur de cette courbe est



FIGURE 4.3 – La région Q a une forme d'entonnoir (*funnel* en anglais).

contenu dans \mathcal{E} . Mais d'après l'exercice 4.4 suivant, $p' \cup q'$ a un sommet convexe distinct des extrémités communes de p' et q', disons intérieur à q'. En court-circuitant ce sommet convexe dans \mathcal{E} , on obtient un chemin reliant les extrémités de q' et strictement plus court que q', contredisant la minimalité de q.

Par ailleurs, supposons que p est un chemin polygonal de \mathcal{E} entre x et y dont les sommets internes sont des sommets de P d'angles intérieurs au moins égaux à π . Un tel chemin existe, puisqu'il existe au moins un plus court chemin dont les sommets internes sont dans P et que de tels sommets ne peuvent être convexes (sinon le chemin ne serait pas de longueur minimale). Si q est un chemin polygonal disctinct de p entre x et y, on en déduit comme ci-dessus deux sous-chemins $p' \subset p$ et $q' \subset q$ tels que $p' \cup q'$ est une courbe fermée simple, laquelle possède au moins trois sommets convexes. Par hypothèse, p' n'a pas de sommet intérieur convexe, donc q' possède un tel sommet. Il suit que q n'est pas un plus court chemin et donc que p est l'unique plus court chemin entre x et y. \Box

Exercice 4.4 Montrer que toute courbe polygonale simple fermée a au moins trois sommets convexes (d'angles intérieurs strictement inférieurs à π).

On note $\pi(x, y)$ l'unique plus court chemin dans \mathcal{E} reliant x et y. On notera $\pi_1 \cdot \pi_2$ la concaténation de deux chemins π_1 et π_2 .

Lemme 4.5 Soit x un point de \mathcal{E} et [y, z] un segment dont l'intérieur est inclus dans l'intérieur de \mathcal{E} (cf. figure 4.3). Il existe un point a tel que $\pi(x, y) = \pi(x, a) \cdot \pi(a, y)$ et $\pi(x, z) = \pi(x, a) \cdot \pi(a, z)$. De plus, si $a \neq y, z$, les deux chemins $\pi(a, y), \pi(a, z)$ et le segment [y, z] forment une courbe simple et chacun des chemins $\pi(a, y)$ et $\pi(a, z)$ est concave : les angles intérieurs aux sommets intérieurs sont plus grands que π et situés d'un même côté du chemin.

Preuve : Soit *a* le point de $\pi(x, y) \cap \pi(x, z)$ le plus éloigné de *x*. D'après l'unicité des plus courts chemins, on peut écrire $\pi(x, y) = \pi(x, a) \cdot \pi(a, y)$ et $\pi(x, z) = \pi(x, a) \cdot \pi(a, z)$ où $\pi(a, y)$ et $\pi(a, z)$ sont disjoints en dehors de *a*. De plus, l'hypothèse $\pi(y, z) = [y, z]$ impose que la courbe fermée $\pi(a, y) \cdot [y, z] \cdot \pi(z, a)$ est simple et borde un polygone $Q \subset \mathcal{E}$. Les chemins $\pi(a, y)$ et $\pi(a, z)$ sont nécessairement concaves dans Q, sinon on pourrait les raccourcir dans Q.



FIGURE 4.4 – (Dessus)Les droites supports des deux chaînes concaves de Q coupent le demi-plan opposé à Q et bordé par pq en secteurs disjoints. (Dessous) les trois configurations du lemme 4.6.

Lemme 4.6 Soit Q un polygone en forme d'entonnoir bordé par un segment [p,q] et deux chaînes polygonales concaves $(a_0, a_1, \ldots, a_r = p)$ et $(a_0, a_{-1}, \ldots, a_{-s} = q)$. On suppose que l'orientation est telle que Q est à droite de (a_{-s}, \ldots, a_r) comme sur la figure 4.4. Soit b un point séparé de Q par la droite pq. Alors,

- ou bien le segment $[a_0, b]$ est contenu dans $Q \cup [p, q, b]$,
- ou bien il existe $i \in [1, r]$ tel que la droite $a_i b$ est tangente à (a_0, a_1, \ldots, a_r) , c'est-à-dire que l'angle orienté $\angle a_{i-1}a_i b$ est au moins π ,
- ou bien il existe $i \in [-1, -s]$ tel que la droite $a_i b$ est tangente à $(a_0, a_{1-1}, \ldots, a_{-s})$, c'est-à-dire que l'angle orienté $\angle ba_i a_{i+1}$ est au moins π .

Preuve : On note H le demi plan bordé par pq et opposé à Q. Soit D_i le secteur diédral contenu dans $Q \cup H$ et bordé par les droites $a_{i-1}a_i$ et a_ia_{i+1} pour $i \in [1-s, r-1]$. On note également D_{-s} (resp. D_r) le secteur bordé par pq et $a_{1-s}q$ (resp. $a_{r-1}p$) dans H et ne contenant pas p (resp. q). La concavité des chaînes (a_0, a_1, \ldots, a_r) et $(a_0, a_{-1}, \ldots, a_{-s})$ entraîne que les D_j sont disjoints et couvrent H. Si $b \in D_i$ on est dans l'une des trois situations du lemme selon que i est positif, nul ou négatif.

On suppose désormais que P est triangulé et on note \mathcal{T} une triangulation de P.

Théorème 4.7 On peut calculer $\pi(x, y)$ en temps O(n), où n est le nombre de sommets de P.



FIGURE 4.5 – (Gauche) Les points x et y peuvent être joints par une suite de triangles $t_x = t_0, t_1, \ldots, t_7 = t_y$. (Droite) Les chemins $\pi(x, p_5) = \pi(x, a_0).(a_0, p_5)$ et $\pi(x, q_5) = \pi(x, a).(a, q_4, q_5)$ dans la bande de triangles $\cup t_i$.

Preuve : Soient t_x et t_y deux triangles de \mathcal{T} contenant respectivement x et y. Le graphe d'adjacence des triangles de \mathcal{T} étant un arbre, il existe un unique chemin reliant t_x à t_y dans cet arbre. Ce chemin correspond à une suite de triangles $t_x = t_0, t_1, \ldots, t_{k+1} =$ t_y telle que t_i et t_{i+1} partagent une arête (une diagonale de P) $p_i q_i$, $0 \le i \le k$. En particulier, ou bien $p_i = p_{i+1}$ ou bien $q_i = q_{i+1}$. Quitte à subdiviser t_y , on peut supposer que $t_y = [p_k, q_k, y]$. On a trivialement $\pi(x, p_0) = [x, p_0]$ et $\pi(x, q_0) = [x, q_0]$. Supposons avoir calculé $\pi(x, p_i)$ et $\pi(x, q_i)$. Par le lemme 4.5 appliqué à x et $[p_i, q_i]$, on a $\pi(x, p_i) =$ $\pi(x, a_0).(a_0, a_1, \dots, a_r = p_i)$ et $\pi(x, q_i) = \pi(x, a_0).(a_0, a_{-1}, \dots, a_{-s} = q_i)$ pour certains sommets a_j de $P, j \in [-s, r]$. On suppose également, sans perte de généralité, que $q_i = q_{i+1}$ comme sur la figure 4.5. Par le lemme 4.6 et la caractérisation des plus courts chemins du lemme 4.3, on sait que $\pi(x, p_{i+1})$ est de la forme $\pi(x, a_i).[a_i, p_{i+1}]$, pour un certain $j \in [-s, r]$. Pour trouver j, on part de j = r et, tant que j > 0, on teste si $\angle a_{j-1}a_jp_{i+1}$ est plus grand que π (i.e. est concave en a_j). Dans l'affirmative, on a $\pi(x, p_{i+1}) = \pi(x, a_j) \cdot [a_j, p_{i+1}]$. Sinon on décrémente j jusqu'à vérifier la condition de concavité. Si le test n'est toujours pas validé pour j = 1, on pose j = 0 et on teste si p_{i+1} est à gauche de la droite orientée $a_j a_{j-1}$ en décrémentant j jusqu'à vérifier cette condition qui nous donne à nouveau $\pi(x, p_{i+1}) = \pi(x, a_i) \cdot [a_i, p_{i+1}]$.

En pratique, plutôt que de faire une (ou deux) boucle sur j on maintient trois listes π_c , π_p et π_q représentant respectivement la partie commune à $\pi(x, p_i)$ et $\pi(x, q_i)$, et les deux branches de l'entonnoir vers p_i et q_i . Au départ on a ainsi $\pi_c = \pi(x, a)$, $\pi_p = \pi(a, p_i)$ et $\pi_q = \pi(a, q_i)$. On commence par dépiler π_p "par le haut" à partir de p_i , puis le cas échéant, on dépile π_q "par le bas" à partir de $a = a_0$. Chaque sommet dépilé dans π_q est de plus empilé au sommet de π_c afin d'allonger la partie commune en conséquence. Chaque sommet dépilé ne pouvant être rempilé dans π_p ni π_q par la suite (pourquoi?), l'algorithme nécessite au plus O(n) opérations. \Box

On appelle *algorithme de l'entonnoir* (funnel algorithm) la procédure décrite ci-dessus [LP84, GHL⁺87]. Elle peut être améliorée si l'on souhaite effectuer plusieurs requêtes de plus courts chemins pour obtenir un temps logarithmique par requête.

Proposition 4.8 Après un précalcul sur P en temps O(n), on peut calculer pour tout

couple de points x, y de \mathcal{E} , la longueur de $\pi(x, y)$ en temps $O(\log n)$. On peut également reporter $\pi(x, y)$ en temps proportionnel à son nombre d'arêtes.

Voir [GH89] et [Her91]. Principe : couper récursivement P en deux par une diagonale de la triangulation de P (supposée donnée). On peut associer à ce découpage récursif un arbre binaire T de profondeur $O(\log n)$. Chaque noeud de l'arbre correspond à un sous-polygone de P et une diagonale qui le coupe en deux. La profondeur $\log n$ résulte de l'existence dans l'arbre dual de la triangulation d'une arête qui coupe en sous-arbres ayant au moins (n+1)/3 sommets chacun (cf. lemme 3.8). Voir l'appendice de [GHL⁺87] pour l'obtention du découpage récursif en temps O(n).

Chaque polygone associé à un noeud est bordé par des diagonales de coupe de profondeurs distinctes et donc par au plus $O(\log n)$ d'entre elles. On enrichie T par des arcs issues du noeud associé à ce polygone et dirigés vers les noeuds associés aux diagonale bordantes (donc de profondeurs moindres). On remarque qu'un noeud est pointé par au plus 2 arcs par niveau plus profond. On montre que la taille du graphe enrichi est toujours en O(n).

Chaque arc de l'arbre T enrichi correspond à deux diagonales d_1, d_2 bordant un polygone. On calcule le "sablier" interne à ce polygone entre ces deux diagonales. Ce sablier permet de calculer efficacement les plus courts chemins entre un point de d_1 et un point de d_2 . Le calcul de ces sabliers peut-être fait de manière efficace en partant des triangles et en remontant dans T. Étant donné une suite de polygones adjacents par une diagonale, la concaténation des tels sabliers pour des polygones consécutifs fournit le sablier de l'union des polygones qui permet de calculer le plus court chemin entre des points dans les triangles.

En pratique, étant donnés x, y dans \mathcal{E} , on commence par calculer la séquence de diagonales traversées par $\pi(x, y)$. Elle correspond à l'unique chemin joignant les triangles contenant xet y dans l'arbre duale de la triangulation de P. Les première et dernière diagonales dans cette séquence sont les diagonales de coupe de deux noeuds v, w dans T. On considère l'unique chemin entre u et v dans T et on fusionne de manière appropriée les sabliers associés. Voir les détails (compliqués) dans [GH89] et [Her91].

Voir également A near optimal algorithm for finding Euclidean shortest path in polygonal domain, Rajasekhar Inkulu, Sanjiv Kapoor, S. N. Maheshwari. Soumis sur ArXiv http://arxiv.org/abs/1011.6481

Chapitre 5

Recherche monodimensionnelle

5.1 Dictionnaires

Un dictionnaire est une structure de données permettant de rechercher, insérer ou supprimer des données. Chaque donnée est supposée posséder une clé qui l'identifie. Cette clé doit appartenir à un univers totalement ordonné (typiquement des entiers). Pour ranger les données on se sert de leur clé. Par la suite on ne s'intéresse qu'aux clés, les "véritables" données pouvant être obtenues à partir des clés à l'aide d'un pointeur par exemple. D'autres opérations telles que la recherche de la clé minimale ou maximale, de la clé suivant ou précédant une clé donnée sont possibles. La fusion et la scission de dictionnaires sont également des opérations courantes.

Classiquement, mais pas uniquement (voir plus bas) les dictionnaires sont représentés à l'aide d'arbres. Ces arbres peuvent être binaires (comme pour les arbres AVL ou bicolores) ou non (arbres a-b).

Références :

http://www.cs.sunysb.edu/~algorith/

http://www.nist.gov/dads/

Introduction to Algorithms. Cormen, Rivest, Leiserson and Stein, M.I.T. PRESS, second edition 2001. Version française chez Dunod, 1994.

STL (http://www.sgi.com/tech/stl/)

LEDA (http://www.mpi-sb.mpg.de/LEDA/MANUAL/MANUAL.html)

5.1.1 Arbres binaires de recherche

Définition 5.1 Un arbre binaire de recherche est un arbre binaire dont les noeuds possèdent des clés rangées dans l'ordre infixe, i.e. un parcours dans l'ordre infixe de l'arbre (sag(b).racine(b).sad(b)) visite les clés dans l'ordre croissant.

Propriété : L'ensemble des arbres binaires de recherche est stable par les opérations de

rotation.

Note : parfois on ne range les clés qu'aux noeuds externes et un noeud interne contient une valeur intermédiaire entre celles de son sous-arbre gauche et de son sous-arbre droit.

5.2 Structures randomisées

Les structures classiques telles que les arbres AVL, bicolores ou a-b sont des structures déterministes. Les performances des opérations de dictionnaires ne dépendent ni des données ni de l'ordre de construction de ces structures. Les algorithmes de construction et de modification sont cependant relativement compliqués. Les structures randomisées telles que les *skip lists* ou les *treaps* utilisent des générateurs aléatoires dans leur construction et permettent d'obtenir des performances *en moyenne* équivalentes à celles des structures déterministes. Les constructions sont généralement plus simples.

5.2.1 Skip list

Une structure de *skip list* sur un ensemble M de n clés est un dictionnaire sur M construit de la manière suivante :

On tire au hasard et de manière indépendante chaque clé de M avec une probabilité 1/2. On obtient ainsi un sous-ensemble M_2 de M avec lequel on recommence la procédure de tirage. On continue ainsi jusqu'à obtenir l'ensemble vide. On obtient finalement une gradation de M, c'est à dire une suite décroissante de sous-ensembles de M:

$$M = M_1 \supseteq M_2 \supseteq \ldots \supseteq M_r \supset M_{r+1} = \emptyset$$

Une skip list sur une telle gradation s'obtient à partir des r listes triées de chaque M_i , appelé niveau, augmenté pour chaque clé de la liste M_i d'un pointeur vers l'occurrence de cette clé dans la liste M_{i-1} et d'un pointeur vers son successeur dans la liste M_i . On ajoute également un élément fictif minimal dans chaque liste que l'on relie entre-eux. On s'intéresse à la taille de cette structure et à la complexité des opérations de recherche, insertion, suppression en fonction du nombre n de clés. Puisque ces grandeurs dépendent de tirages aléatoires, on s'intéresse à leur valeur moyenne où l'on considère tous les tirages de clés indépendants. De plus, pour caractériser le fait que les grandeurs s'écartent très peu des valeurs moyennes (c.a.d. que la distribution des valeurs est bien localisée autour de la moyenne) on introduit la notation suivante :

Définition 5.2 Soient f(n) et g(n) des variables aléatoires dépendant d'un paramètre n. On écrit $f = \tilde{O}(g)$ si P(f(n) > cg(n)) < 1/p(n,c) où p(n,c) est un polynôme en n dont le degré tend vers l'infini avec c.

Intuitivement, un $\tilde{O}(g)$ est un O(g) avec très forte probabilité.

Soit h_i la variable aléatoire qui donne le nombre de niveaux auxquels appartient la clé i. h_i suit une loi géométrique de paramètre 1/2, d'où $P(h_i = k) = 1/2^k$ et $E(h_i) = 2$.

On pose $h = \max_{1 \le i \le n} h_i$ la hauteur d'une skip list.

Lemme 5.3 $E(h) = O(\log n)$ et $h = \tilde{O}(\log n)$.

Preuve :

$$P(h > k) = P((h_1 > k) \lor (h_2 > k) \lor \ldots \lor (h_n > k)) \le \sum_{i=1}^n P(h_i > k) = n/2^k.$$

On en déduit pour tout c > 0 que $P(h > c \log n) \le 1/n^{c-1}$, d'où $h = O(\log n)$. On a de plus (cf. lemme 1.10)

$$E(h) = \sum_{k=0}^{c\log n-1} P(h > k) + \sum_{k=c\log n}^{\infty} P(h > k) \le c\log n + n \sum_{k=c\log n}^{\infty} \frac{1}{2^k} = c\log n + \frac{2}{n^{c-1}}.$$

On note $t = \sum_{1 \le i \le n} h_i$ la taille d'une skip list.

Lemme 5.4 La taille t d'une skip list sur n clés vérifie E(t) = O(n) et $t = \tilde{O}(n)$.

Preuve : Par linéarité de l'espérance, on a $E(t) = \sum_i E(h_i) = 2n$. En mettant bout a bout les tirages pour chaque clé, t s'interprète comme le nombre de tirages nécessaires pour obtenir n échecs. Dit autrement t suit une loi binomiale négative de paramètres n et 1/2. Le lemme 1.29 utilisant la technique de majoration de Chernoff permet de conclure. \Box

Remarque : La construction d'une skip list sur un ensemble de n clés peut s'obtenir après tri de ses clés en temps proportionnel à sa taille, i.e. en temps $O(n \log n + t) = \tilde{O}(n \log n)$.

Soit K une clé, fixée une fois pour toute, à rechercher dans une skip list. On note K_i la plus grande clé du niveau i (i.e. de M_i) inférieure ou égale à K. On note également X_i le nombre de clés comprises, au sens large, entre K_{i+1} et K_i dans M_i . Lorsque M_i est vide, on pose $X_i = 0$ et $K_i = l'élément$ fictif minimal (voir plus haut). Pour rechercher K dans la skip list on commence par parcourir les clés du plus haut niveau M_r dans l'ordre croissant jusqu'à atteindre K_r . À l'aide du pointeur de K_r vers le niveau r - 1 on descend sur sa copie dans M_{r-1} puis on parcourt M_{r-1} dans l'ordre croissant des clés jusqu'à atteindre se poursuit récursivement jusqu'au premier niveau. La recherche est fructueuse si et seulement si $K_1 = K$.

Lemme 5.5 Le temps de recherche d'une clé fixée est un $\tilde{O}(\log n)$ et le temps moyen est un $O(\log n)$.

Preuve : Le temps de recherche est clairement proportionnel à la longueur ℓ du "chemin" de recherche, i.e. à $\ell = \sum_{i\geq 1} X_i$ (cette longueur inclut les marches horizontales et verticales). Notons que pour M_i fixé, M_{i+1} est obtenu en tirant aléatoirement et indépendamment chaque clé de M_i avec une probabilité 1/2. Par conséquent, la variable aléatoire $X_i|M_i$ suit une loi géométrique de paramètre 1/2 (plus précisément $X_i|M_i$ est majorée par une variable suivant une telle loi). On en déduit $E(X_i|M_i) \leq 2$, d'où inconditionnellement $E(X_i) \leq 2$.

On note Y_i la variable aléatoire valant 0 si M_i est vide et 1 sinon, de sorte que $X_i \leq nY_i$. On a $P(Y_i = 1) = P(h \geq i) \leq n/2^{i-1}$, soit $E(Y_i) \leq n/2^{i-1}$. On obtient¹

$$E(\sum_{i\geq 1} X_i) = \sum_{i\geq 1} E(X_i) \le \sum_{i=1}^{c\log n} E(X_i) + \sum_{i>c\log n} E(nY_i) \le 2c\log n + 2/n^{c-2}.$$

Et on conclut $E(\ell) = O(\log n)$.

Montrons que $\ell = \tilde{O}(\log n)$. Pour cela on 'coupe' ℓ en deux en écrivant $\ell = \ell_{\leq} + \ell_{>}$ avec $\ell_{\leq} = \sum_{i=1}^{c \log n} E(X_i)$ et $\ell_{>} = \sum_{i>c \log n} E(X_i)$. Alors ℓ_{\leq} est une somme de $c \log n$ variables aléatoires majorées par des variables de loi géométrique de paramètre 1/2 et est donc majorée par une variable aléatoire suivant une loi binomiale négative de paramètres $c \log n$ et 1/2. On en déduit par la technique de Chernoff (lemme 1.29) que

$$P(\ell_{\leq} > c(2+d)\log n) \le \exp(-dc\log n/4) = O(\frac{1}{n^{dc/4}})$$

dès que $d \ge 3$. D'où $\ell_{\le} = \tilde{O}(\log n)$.

Par ailleurs $\ell_{>}$ est trivialement majorée par la hauteur h de la skip list additionnée au nombre d'éléments restant au niveau $c \log n + 1$. Soit Z_i la fonction indicatrice de la présence de la *i*-ème clé au niveau $c \log n + 1$, de sorte que $\ell_{>} \leq h + \sum_{i=1}^{n} Z_i$. Les Z_i sont indépendantes et suivent une loi de Bernoulli de paramètre $1/2^{c \log n} = 1/n^c$. Donc $\sum_{i=1}^{n} Z_i$ suit une loi binomiale de paramètre n et $1/n^c$. Par l'inégalité de Markov on a

$$P(\sum_{i=1}^{n} Z_i > c) \le E(\sum_{i=1}^{n} Z_i)/c \le \frac{1}{cn^{c-1}}.$$

D'où $\sum_{i=1}^{n} Z_i = \tilde{O}(1)$. D'après le lemme 5.3, $h = \tilde{O}(\log n)$ et on conclut finalement que $\ell_{>} = \tilde{O}(\log n)$ puis que $\ell = \tilde{O}(\log n)$.

On a en fait un résultat plus fort ne dépendant pas de la clé particulière à chercher :

Lemme 5.6 Le temps maximal de recherche d'une clé quelconque dans une skip list est un $\tilde{O}(\log n)$.

Preuve :

$$P(\max_{K} \ell(K) > c \log n) \le \sum_{K} P(\ell(K) > c \log n) \le \frac{n}{p(n,c)} = \frac{1}{q(n,c)}.$$

^{1.} On peut invoquer le théorème de convergence monotone pour faire commuter l'espérance et la somme d'une série.

Lemme 5.7 Le temps maximal de suppression d'une clé quelconque dans une skip list est un $\tilde{O}(\log n)$.

Pour supprimer une clé on effectue sa recherche en la suppriment des niveaux M_i où elle apparaît. Le temps de suppression est donc de l'ordre du temps de recherche. Si on dispose d'un pointeur sur l'élément à supprimer alors la suppression est un $\tilde{O}(1)$ car $h_K = \tilde{O}(1)$.

Lemme 5.8 Le temps maximal d'insertion d'une clé dans une skip list est un $O(\log n)$.

Pour insérer une clé on effectue des tirages jusqu'à obtenir un échec. Si k est le nombre de tirages effectués on insère la clé dans les k premiers niveaux en même temps que l'on effectue une recherche dans la skip list.

Référence :

- Skip Lists : A probabilistic Alternative to Balanced Trees. W. Pugh. Communication of the ACM, 33(6), june 1990.

5.2.2 Arbres binaires de recherche aléatoires

Etant donnée une permutation σ de [1, n] on construit un arbre binaire de recherche en introduisant $\sigma(1)$ puis $\sigma(2), \ldots$, puis $\sigma(n)$ dans un arbre initialement vide. Les n!permutations de [1, n] permettent d'obtenir toutes les formes possibles d'arbres binaires de recherche sur n éléments. En considérant une loi de probabilité uniforme sur les n!permutations on en déduit une loi de probabilité sur les arbres binaires de recherche.

Proposition 5.9 La hauteur d'un arbre binaire de recherche aléatoire est un $\hat{O}(\log n)$ et sa profondeur moyenne est un $O(\log n)$.

Notons que d'après le théorème 1.32, la hauteur moyenne d'un arbre de recherche aléatoire est un $O(\log n)$, ce qui implique évidement que la profondeur moyenne des clés est du même ordre. On donne ici une preuve de ce dernier résultat, moins fort mais plus simple à montrer.

Preuve : On note X_j^i la variable aléatoire indiquant si j est un ancêtre de i dans l'arbre binaire de recherche associé à une permutation aléatoire σ . On note τ la permutation (aléatoire) inverse de σ . $X_j^i = 1$ si et seulement si $\tau(j) = \min_{k \in [i,j]} \tau(k)$, c.a.d. si et seulement si j est inséré avant i et aucun élément inséré avant j ne sépare i et j. Clairement $P(X_j^i = 1 | \tau([i, j])) = \frac{1}{|i-j|+1}$, d'où $E(X_j^i) = \frac{1}{|i-j|+1}$.

Soit $h_i = \sum_{j \neq i} X_j^i$ la hauteur de *i* dans l'arbre binaire de recherche. Par linéarité de l'espérance $E(h_i)$ vaut $\sum_{j=1}^n \frac{1}{|i-j|+1} = H_i + H_{n+1-i} - 1 = O(\log n)$.

Par ailleurs, pour *i* fixé et i < j le lemme 5.10 ci-dessous indique que les variables X_j^i sont mutuellement indépendantes. Par la technique de Chernoff on peut donc écrire pour tout *t* et tout λ positif

$$P(\sum_{j>i} X_j^i > t) \le \frac{\prod_{j>i} E(\exp(\lambda X_j^i))}{\exp(\lambda t)}.$$

Or

$$E(\exp(\lambda X_j^i)) = \frac{\exp(\lambda)}{j-i+1} + 1 - \frac{1}{j-i+1} = 1 + \frac{\exp(\lambda) - 1}{j-i+1} \le \exp(\frac{\exp(\lambda) - 1}{j-i+1})$$

d'où

$$P(\sum_{j>i} X_j^i > t) \le \exp((\exp(\lambda) - 1)H_n - \lambda t) \le \exp((\exp(\lambda) - 1)(\ln n + 1) - \lambda t)$$

En choisissant $t = c \ln n$ et $\lambda = \ln c$ on trouve

$$P(\sum_{j>i} X_j^i > c \ln n) \le \frac{\exp(c-1)}{n^{c(\ln c-1)+1}} = \frac{1}{p(n,c)}$$

En opérant de même avec la somme $\sum_{j < i} X_j^i$ on en déduit que $h_i = \tilde{O}(\log n)$. Comme pour l'analyse de la hauteur d'une skip list on conclut finalement que la hauteur $h = \max_i h_i$ d'un arbre binaire de recherche aléatoire est un $\tilde{O}(\log n)$ et que son espérance est un $O(\log n)$.

Dit autrement un arbre binaire de recherche aléatoire est presque toujours bien équilibré.

Lemme 5.10 Avec les notations de la preuve ci-dessus, soient $1 \le i < j_1 < j_2 < \ldots < j_k \le n$. Alors les variables $X_{j_1}^i, X_{j_2}^i, \ldots, X_{j_k}^i$ sont mutuellement indépendantes. On a un résultat analogue pour $1 \le j_1 < j_2 < \ldots < j_k < i \le n$.

Preuve : On raisonne par récurrence sur k. Soient $\epsilon_r \in \{0, 1\}, r \in [1, k]$. On pose $A = \{X_{j_1}^i = \epsilon_1 \land \ldots \land X_{j_{k-1}}^i = \epsilon_{k-1}\}$. Par hypothèse de récurrence à l'ordre k - 1, les variables $X_{j_1}^i, X_{j_2}^i \ldots, X_{j_{k-1}}^i$ sont indépendantes. On remarque en particulier que $P(A) = \prod_{r=1}^{k-1} P(X_{j_r}^i = \epsilon_r)$ ne dépend que des grandeurs $|j_r - i|$. On considère une partie $I \subset [1, n]$ à $j_k - i + 1$ éléments et le sous-ensemble de permutations de [1, n] :

$$B_I = \{\tau : \tau([i, j_k]) = I \land \tau(j_k) = \min I\}$$

On note que les B_I sont disjoints et que leur union, $\cup_I B_I$, est l'événement $\{X_{j_k}^i = 1\}$. J'affirme que $P(A|B_I) = P(A)$. Pour le voir on considère l'injection croissante $\iota : [1, j_k - i] \rightarrow [1, n]$ telle que Im $\iota = I$ et la surjection ϕ de B_I dans les permutations de $[1, j_k - i]$:

$$\begin{array}{cccc} B_I & \stackrel{\phi}{\to} & \mathcal{S}_{j_k-i} \\ \tau & \mapsto & \tau' : \ell \mapsto \iota^{-1}(\tau(\ell+i-1)) \end{array}$$

Puisque l'appartenance de τ à A ne dépend que des ordres relatifs de $\tau(i), \tau(i+1), \ldots, \tau(j_{k-1}),$ on a

$$\tau \in A \Leftrightarrow \phi(\tau) \in A' = \{X_{j_1-i+1}^i = \epsilon_1, X_{j_2-i+1}^i = \epsilon_2, \dots, X_{j_{k-1}-i+1}^i = \epsilon_{k-1}\}$$

De plus, le nombre $|\phi^{-1}(\tau')|$ ne dépend pas de $\tau' \in S_{j_k-i}$. On en déduit, avec la remarque ci-dessus, $P(A|B_I) = P(A') = P(A)$, et par suite $P(A|X_{j_k}^i = 1) = P(A)$ (cf. exercice 1.21). En reportant cette égalité dans la suivante

$$P(A) = P(A|X_{j_k}^i = 1)P(X_{j_k}^i = 1) + P(A|X_{j_k}^i = 0)P(X_{j_k}^i = 0)$$

on obtient également $P(A|X_{j_k}^i=0)=P(A)$. Finalement on en déduit,

$$\forall \epsilon_k \in \{0, 1\} : P(A \land X^i_{j_k} = \epsilon_k) = P(A)P(X^i_{j_k} = \epsilon_k)$$

soit encore

$$P(X_{j_1}^i = \epsilon_1 \land \ldots \land X_{j_k}^i = \epsilon_k) = \prod_{r=1}^k P(X_{j_r}^i = \epsilon_r)$$

5.2.3 Tree + Heap = Treap

Introduits sous le nom d'arbres cartésiens par J. Vuillemin, les treaps sont des arbres binaires de recherches dont les noeuds possèdent une clé et une priorité comprise entre 0 et 1. Un treap est un arbre binaire de recherche pour les clés - les noeuds dans le sous arbre gauche (droit) d'un noeud ont une clé inférieure (supérieure) à la clé de ce noeud - et un tas pour les priorités (le parent d'un noeud a une priorité plus grande que celle de ce noeud). Un treap est donc le résultat d'un arbre binaire de recherche construit en introduisant un à un les éléments dans l'ordre de leur priorité. En choisissant les priorités de manière aléatoire on peut simuler un arbre binaire de recherche aléatoire. Ceci assure d'après le lemme 5.9 que la recherche a un coût en $\tilde{O}(\log n)$. Il est possible de maintenir dynamiquement un treap, i.e. d'insérer ou de supprimer des éléments en conservant la propriété d'arbre aléatoire. Pour cela, on utilise le fait que l'opération de rotation sur un arbre binaire (cf. section 1.4) préserve l'ordre infixe, intervertit la parenté pour deux noeuds, et préserve la propriété de tas pour le reste. Ceci permet, par rotations successives de faire remonter ou descendre un noeud dans un treap pour l'ôter ou le positionner à sa bonne place, c'est-à dire comme s'il avait été introduit 'à l'instant' donné par sa priorité.

Références :

- Randomized search trees. C. Aragon and R. Seidel. Algorithmica 16. pp 464-497. 1996.

- Randomized Binary Search Trees. C. Martínez and S. Roura. Journal of the ACM. 45. pp 288-323. 1998.

Chapitre 6

Polytopes

L'étude des polytopes tire son origine au XVIIIe siècle de la mécanique et plus spécifiquement de l'analyse des points d'équilibre d'une masse ponctuelle soumise à des contraintes. Cette analyse fait apparaître des inéquations linéaires en lien avec les multiplicateurs de Lagrange (1788). Au XIXe siècle, le développement de l'étude des systèmes d'inéquations linéaires doit beaucoup à Fourier et fut motivée par différentes branches des mathématiques comme les probabilités ou la théorie des nombres ou encore par les théories politiques (élections) ou économiques. Ces dernières, ainsi que la théorie de jeux, furent également à la source de nombreux problèmes de programmation linéaire au XXesiècle. Dans le même temps l'étude de la convexité et de la théorie des polytopes en tant que telles s'est largement développée en mathématique.

Pour de plus amples références, voir les notes historiques dans

- Theory of Linear and Integer Programming. A. Schrijver. Wiley-Interscience, 1986.

- The Evolution of Methods of Convex Optimization. V. M. Tikhomirov. The American Mathematical Monthly. Jan. 1996, pp. 65-71.

6.1 Notations

Un vecteur colonne ou ligne est noté en caractère gras comme le vecteur \mathbf{x} , ses composantes x_i sont notées en caractères maigres. Les produit scalaire est noté de manière matricielle comme le produit d'un vecteur ligne par un vecteur colonne. La notation $\mathbf{x} \leq \mathbf{y}$ signifie $x_i \leq y_i$ pour tout i.

Un demi-espace $\{\mathbf{x} \mid \mathbf{cx} \leq c_0\}$ contenant un polytope P est dit valide pour P. Par extension, on dit que l'hyperplan $\{\mathbf{x} \mid \mathbf{cx} = c_0\}$ est valide pour P si le demi-space $\{\mathbf{x} \mid \mathbf{cx} \leq c_0\}$ est valide pour P.

Par concision j'écrirai { $\mathbf{cx} \leq c_0$ } pour { $\mathbf{x} \mid \mathbf{cx} \leq c_0$ }. Par abus de langage, j'identifierai un hyperplan avec son équation de sorte que je pourrai écrire $H = {\mathbf{cx} = c_0} = {H(\mathbf{x}) = 0}$ et { $\mathbf{cx} \leq c_0$ } = { $H(\mathbf{x}) \leq 0$ }.

Pour toute partie X de \mathbb{R}^n , aff(X) désigne l'enveloppe affine de X dans \mathbb{R}^n tandis que
vec(X) désigne l'espace vectoriel engendré par les éléments de X vus comme des vecteurs de \mathbb{R}^n .

6.2 Convexité

Définition 6.1 Une combinaison convexe d'une famille de n points $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}$ de \mathbb{R}^d est un point \mathbf{x} de la forme $\mathbf{x} = \sum_i t_i \mathbf{x_i}$, avec $t_i \ge 0$ et $\sum_i t_i = 1$.

Définition 6.2 Un ensemble X de \mathbb{R}^d est convexe si pour tout couple de points de X le segment qui les joint est inclus dans X.

Remarque 6.3 Par récurrence on en déduit facilement qu'un ensemble est convexe si et seulement si il est stable par combinaison convexe de familles finies de ses points.

Définition 6.4 Soit X une partie de \mathbb{R}^d . L'enveloppe convexe de X, notée Conv(X), est le plus petit convexe contenant X (ce qui a un sens puisque la propriété de convexité est stable par intersection).

Lemme 6.5 Conv(X) est l'ensemble des combinaisons convexes de familles finies de points de X.

Preuve : Je note C l'ensemble des combinaisons convexes de familles finies de X. Par la remarque précédente C est convexe (une combinaison convexe de combinaisons convexes en est une) et comme C contient X on en déduit $Conv(X) \subset C$. La même remarque implique $C \subset Conv(X)$, d'où l'identité.

Lemme 6.6 (Radon) Soit A un ensemble de d + 2 points de \mathbb{R}^d alors il existe deux parties disjointes A_1 et A_2 de A telles que $Conv(A_1) \cap Conv(A_2)$ n'est pas vide.

Preuve : Les d + 2 points $\mathbf{a_1}, \mathbf{a_2}, \dots, \mathbf{a_{d+2}}$ de A sont affinement dépendants. Il existe donc des λ_i non tous nuls tels que $\sum_i \lambda_i \mathbf{a_i} = \mathbf{0}$ et $\sum_i \lambda_i = 0$. En séparant les termes avec des λ_i strictement positifs des termes avec des λ_i strictement négatifs on conclut facilement.

Théorème 6.7 Conv(X) est l'ensemble des combinaisons convexes de familles de d + 1points de X. Autrement dit Conv(X) est l'union des d-simplexes (possiblement dégénérés) dont les sommets sont des points de X.

Preuve : Par le lemme 6.5 tout point \mathbf{x} de Conv(X) s'écrit $\sum_{i=1}^{n} \lambda_i \mathbf{x_i}$ avec $\mathbf{x_i} \in X, \lambda_i > 0$ et $\sum_{i=1}^{n} \lambda_i = 1$. Si n > d+1 alors les $\mathbf{x_i}$ sont affinement liés et il existe des μ_i non tous nuls tels que $\sum_{i=1}^{n} \mu_i \mathbf{x_i} = \mathbf{0}$ et $\sum_{i=1}^{n} \mu_i = 0$. On peut choisir un réel α tel que $\lambda_i + \alpha \mu_i$ est nul pour au moins un indice i et positif sinon. Donc $\mathbf{x} = \sum_{i=1}^{n} (\lambda_i + \alpha \mu_i) \mathbf{x_i}$ est combinaison convexe d'au plus n - 1 points de X et on termine par récurrence sur n. **Lemme 6.8 (de séparation)** Soit C un convexe compact de \mathbb{R}^d et D un convexe fermé de \mathbb{R}^d disjoint de C. Alors il existe un hyperplan H les séparant strictement, i.e. tel que C et D soient respectivement contenus dans les deux demi-espaces ouverts délimités par H.

Preuve : Supposons tout d'abord D compact. L'application distance est continue sur le compact $C \times D$ où elle atteint son minimum. Soient donc $\mathbf{x} \in C$ et $\mathbf{y} \in D$ tels que $d(\mathbf{x}, \mathbf{y}) = d(C, D) > 0$. On vérifie que l'hyperplan médiateur de \mathbf{x} et \mathbf{y} convient pour H. Si D n'est pas compact on l'intersecte avec une boule B compacte suffisamment grande pour que $d(C, D \cap B) = d(C, D)$ et on se ramène au cas précédent.

Théorème 6.9 (Helly, 1923) Soient n > d et C_1, C_2, \ldots, C_n des convexes de \mathbb{R}^d tels que l'intersection de d + 1 quelconques de ces convexes est non vide, alors $\cap_i C_i$ est non vide.

Preuve : On raisonne par récurrence sur n. Pour tout i on considère, par hypothèse de récurrence, un point a_i dans l'intersection des n-1 convexes $\bigcap_{j\neq i}C_j$. On obtient ainsi un ensemble de n points $\{a_1, \ldots, a_i, \ldots a_n\}$. Par le théorème de Radon on peut en extraire deux sous-ensembles disjoints A et B dont les enveloppes convexes s'intersectent. Tout point x de cette intersection est dans $\bigcap_i C_i$. En effet, si a_i n'est pas dans A, alors chaque point de A est dans C_i , et donc $x \in Conv(A) \subset C_i$. De même si i n'est pas dans B. \Box

Exercice 6.10 Soit P un ensemble de n points de \mathbb{R}^d . Un point central pour P est un point $x \in \mathbb{R}^d$ tel que tout demi-espace qui ne contient pas x contient au plus $n\frac{d}{d+1}$ points de P. Montrer à l'aide du théorème de Helly que tout ensemble fini de points admet un point central.

6.3 Le théorème de Minkowski-Weyl

Ce théorème établit l'équivalence entre les objets obtenus comme enveloppe convexe de familles finies de points ou comme intersection de familles finies de demi-espaces lorsque celle-ci est bornée.

Définition 6.11 Un cône polyédrique (ou polyédral) est une intersection d'une famille finie de demi-espaces vectoriels (de la forme $\{\mathbf{x} \mid \mathbf{ax} \leq 0\}$). L'enveloppe conique d'une famille finie de vecteurs est l'ensemble des combinaisons linéaires à coefficients non négatifs de ces vecteurs.

Théorème 6.12 (de Minkowski-Weyl pour les cônes) Tout cône polyédrique est une enveloppe conique et réciproquement.

Lemme 6.13 La projection sur un sous-espace d'un cône polyédrique est un cône polyédrique. **Preuve :** Une projection sur un sous-espace de codimension p s'obtient comme p projections successives sur des sous-espaces de codimension 1 (dans les espaces appropriés). Il suffit donc de se restreindre à ce dernier cas. Par changement de coordonnées on peut supposé que cette projection est la projection orthogonale sur $\{x_1 = 0\}$. Soit $C = \{A\mathbf{x} \leq \mathbf{0}\}$ un cône polyédrique où les inéquations sont exprimées dans une repère adapté à la projection. On utilise la procédure de *Fourier-Motzkin* (1927) après avoir normalisé les inéquations de C : si C s'écrit

$$\begin{cases} x_1 + \mathbf{a}'_{\mathbf{i}}\mathbf{x}' \leq 0, \ i \in I^+ \\ -x_1 + \mathbf{a}'_{\mathbf{j}}\mathbf{x}' \leq 0, \ j \in I^- \\ & \mathbf{a}'_{\mathbf{k}}\mathbf{x}' \leq 0, \ k \in I^0 \end{cases}$$

alors sa projection sur $\{x_1 = 0\}$ s'écrit

$$C_1 = \begin{cases} (\mathbf{a}'_{\mathbf{i}} + \mathbf{a}'_{\mathbf{j}})\mathbf{x}' &\leq 0, \quad (i, j) \in I^+ \times I^- \\ \mathbf{a}'_{\mathbf{k}}\mathbf{x}' &\leq 0, \quad k \in I^0 \end{cases}$$

En effet, tout point projeté de C est clairement dans C_1 . Par ailleurs si $\mathbf{x}' \in C_1$ alors le point $(-\max_{I^+} \mathbf{a}'_i \mathbf{x}', \mathbf{x}') \in C$, donc C_1 est contenu dans la projection de C.

Lemme 6.14 L'intersection d'une enveloppe conique avec un sous-espace est une enveloppe conique.

Preuve : Il suffit de se restreindre à l'intersection avec un sous-espace de codimension 1, par exemple $\{x_1 = 0\}$ et de conclure par récurrence sur la codimension. Soit une enveloppe conique $E = \{R\lambda, \lambda \ge 0\}$ avec $R = (R^+, R^-, R^0)$ et $\mathbf{r} \in R^+$ (resp. R^-, R^0) si $r_1 = 1$ (resp. -1, 0). Alors $E \cap \{x_1 = 0\}$ est l'enveloppe conique E_1 sur $(R^+ + R^-, R^0)$. En effet,

$$\mathbf{x} \in E_1 \implies \mathbf{x} = \sum_{R^+ \times R^-} \lambda_{\pm} (\mathbf{r}^+ + \mathbf{r}^-) + \sum_{R^0} \lambda_0 \mathbf{r}^0$$

avec $\lambda_{\pm}, \lambda_0 \ge 0$. Donc $\mathbf{x} \in E \cap \{x_1 = 0\}$, i.e. $E_1 \subset E \cap \{x_1 = 0\}$. Réciproquement,

$$\mathbf{x} \in E \cap \{x_1 = 0\} \implies \mathbf{x} = \sum_{R^+} \lambda_+ \mathbf{r}^+ + \sum_{R^-} \lambda_- \mathbf{r}^- + \sum_{R^0} \lambda_0 \mathbf{r}^0$$

avec $\sum_{R^+} \lambda_+ = \sum_{R^-} \lambda_-$ et $\lambda_+, \lambda_-, \lambda_0 \ge 0$. On en déduit

$$\mathbf{x} = \frac{1}{\sum_{R^+} \lambda_+} \left(\sum_{R^+} \lambda_+ (\sum_{R^-} \lambda_-) \mathbf{r}^+ + \sum_{R^-} \lambda_- (\sum_{R^+} \lambda_+) \mathbf{r}^- \right) + \sum_{R^0} \lambda_0 \mathbf{r}^0$$
$$= \sum_{R^+} \sum_{R^-} \frac{\lambda_+ \lambda_-}{\sum_{R^+} \lambda_+} (\mathbf{r}^+ + \mathbf{r}^-) + \sum_{R^0} \lambda_0 \mathbf{r}^0.$$

d'où $\mathbf{x} \in E_1$ et $E \cap \{x_1 = 0\} \subset E_1$.

Preuve du théorème 6.12 : Soit une enveloppe conique $E = {\mathbf{x} | \exists \lambda \ge \mathbf{0} : \mathbf{x} = R\lambda}$. E est la projection "sur \mathbf{x} parallèlement à λ " du cône polyédrique

$$\left\{ \begin{bmatrix} I & -R \\ -I & R \\ 0 & -I \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} \le \mathbf{0} \right\}.$$

Le lemme 6.13 assure que E est un cône polyédrique. Réciproquement soit un cône polyédrique $C = \{A\mathbf{x} \leq \mathbf{0}\} = \{(\mathbf{x}, \lambda) \mid A\mathbf{x} \leq \lambda\} \cap \{\lambda = \mathbf{0}\}.$ Montrons que $C' = \{(\mathbf{x}, \lambda) \mid A\mathbf{x} \leq \lambda\}$ est l'enveloppe conique E des vecteurs $\pm \begin{bmatrix} \mathbf{e_i} \\ A\mathbf{e_i} \end{bmatrix}$ et $\begin{bmatrix} \mathbf{0} \\ \epsilon_i \end{bmatrix}$ où les $\mathbf{e_i}$ et ϵ_j constituent des bases des espaces adéquats :

$$\begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} \in E \implies \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} = \sum_{i} (u_i^+ - u_i^-) \begin{bmatrix} \mathbf{e_i} \\ A\mathbf{e_i} \end{bmatrix} + \sum_{j} v_j \begin{bmatrix} \mathbf{0} \\ \epsilon_j \end{bmatrix} \text{ avec } u_i^+, u_i^-, v_j \ge 0.$$

d'où

$$\begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ A\mathbf{x} + \mathbf{v} \end{bmatrix}$$
 avec $\mathbf{v} \ge \mathbf{0}$

On en déduit $A\mathbf{x} \leq \lambda$ i.e. $\begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} \in C'$. Réciproquement supposons $\begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} \in C'$. Alors on peut écrire

$$\begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ A\mathbf{x} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \lambda - A\mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^+ \\ A\mathbf{x}^+ \end{bmatrix} - \begin{bmatrix} \mathbf{x}^- \\ A\mathbf{x}^- \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \lambda - A\mathbf{x} \end{bmatrix} \text{ avec } \mathbf{x}^+, \mathbf{x}^- \ge \mathbf{0}.$$

ce qui montre que $\begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} \in E$. Finalement E = C' et $C = C' \cap \{\lambda = \mathbf{0}\}$. Le lemme 6.14 permet de conclure.

Définition 6.15 Un polyèdre est l'intersection d'un famille finie de demi-espaces affines.

Théorème 6.16 (de Minkowski-Weyl pour les polyèdres) Tout polyèdre est la somme (de Minkowski) d'une enveloppe convexe d'une famille finie de points et d'un cône polyédrique et réciproquement.

Preuve : Soit un polyèdre
$$P = \{A\mathbf{x} \le \mathbf{b}\} = \{\mathbf{x} \mid \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} \in C\}$$
 où C est le cône

$$C = \{\begin{bmatrix} x_0 \\ \mathbf{x} \end{bmatrix} \mid \begin{bmatrix} -1 & \mathbf{0} \\ -\mathbf{b} & A \end{bmatrix} \begin{bmatrix} x_0 \\ \mathbf{x} \end{bmatrix} \le \mathbf{0}\}.$$

Dit autrement C est le cône de sommet **0** sur une copie de P dans l'hyperplan $x_0 = 1$. Le théorème de Minkowski-Weyl pour les cônes assure que $C = \{R\lambda, \lambda \ge 0\}$ pour un certain R vérifiant $(R\lambda)_0 \ge 0$ pour tout $\lambda \ge 0$. On en déduit

$$\begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} \in C \Leftrightarrow \exists \lambda \ge \mathbf{0} : \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \sum_i \lambda_i r_{i0} \\ \sum_i \lambda_i \mathbf{r_i} \end{bmatrix}$$

où les r_{i0} sont non négatifs. En séparant les indices pour les quels r_{i0} est soit positif soit nul en I^+ et I^0 respectivement, on obtient

$$\mathbf{x} \in P \Leftrightarrow \exists \lambda \ge \mathbf{0} : \mathbf{x} = \sum_{I^+} \lambda_i r_{i0} \frac{\mathbf{r_i}}{r_{i0}} + \sum_{I^0} \lambda_i \mathbf{r_i} \text{ et } \sum_{I^+} \lambda_i r_{i0} = 1.$$

D'où $\mathbf{x} \in Conv(\frac{\mathbf{r}_{\mathbf{i}}}{r_{i0}})_{I^+} + C\hat{o}ne(\mathbf{r}_{\mathbf{i}})_{I^0}.$

Pour la réciproque on considère la somme Q d'une enveloppe convexe d'une famille finie de points S et d'une enveloppe conique sur une famille R de vecteurs :

$$Q = \{S\lambda + R\mu \mid \lambda, \mu \ge \mathbf{0} \text{ et } \mathbb{I}\lambda = 1\}. \text{ Alors } Q = \{\mathbf{x} \mid \begin{bmatrix} 1\\ \mathbf{x} \end{bmatrix} \in C\} \text{ où } C \text{ est le cône}$$

$$C = \left\{ \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ S & R \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} \mid \lambda, \mu \ge \mathbf{0} \right\}$$

Le théorème de Minkowski-Weyl pour les cônes assure que C est de la forme $\left\{ \begin{bmatrix} x_0 \\ \mathbf{x} \end{bmatrix} \mid (-\mathbf{b} \quad A) \begin{bmatrix} x_0 \\ \mathbf{x} \end{bmatrix} \leq \mathbf{0} \right\}$ d'où $P = \{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}\}$

Exercice 6.17 Le cône de récession du polyèdre $P = \{A\mathbf{x} \leq \mathbf{b}\}$ est l'ensemble des directions \mathbf{y} telles que $\mathbf{x} + \lambda \mathbf{y} \in P$ pour un certain $\mathbf{x} \in P$ et tout $\lambda \geq 0$. Montrer que le cône de récession de P est le cône $\{A\mathbf{x} \leq \mathbf{0}\}$. Montrer que dans toute décomposition de P en somme d'une enveloppe convexe de points et d'un cône polyédrique, ce cône est le cône de récession de P.

L'espace de linéalité de P est l'ensemble des directions \mathbf{y} telles que $\mathbf{x} + \lambda \mathbf{y} \in P$ pour un certain $\mathbf{x} \in P$ et tout λ . Montrer que cet espace a pour équation $\{A\mathbf{x} = \mathbf{0}\}$.

Définition 6.18 Un polytope est un polyèdre borné.

Corollaire 6.19 (théorème de Minkowski-Weyl pour les polytopes) Tout polytope est une enveloppe convexe d'une famille finie de points et réciproquement.

6.4 Lemmes de Farkas, 1896

Lemme 6.20 Soit un polyèdre $P = \{A\mathbf{x} \leq \mathbf{b}\}$ et soit p_i la projection sur $\{x_i = 0\}$ parallèlement à $\mathbf{e_i}$. Alors il existe une matrice C à coefficients non négatifs telle que

$$Elim_i(P) = p_i^{-1}(p_i(P)) = \{CA\mathbf{x} \le C\mathbf{b}\}.$$

Preuve : appliquer la procédure de Fourier-Motzkin aux inéquations de *P*.

Lemme 6.21 Les deux assertions suivantes sont équivalentes
(i) {Ax ≤ b} est vide,

(ii) $\exists \mathbf{c} \geq \mathbf{0}$ tel que $\mathbf{c}A = \mathbf{0}$ et $\mathbf{c}\mathbf{b} < 0$.

Dit autrement si le polyèdre $\{A\mathbf{x} \leq \mathbf{b}\}$ est vide alors on peut déduire de son système d'inéquations une inéquation absurde : $0 = \mathbf{c}A\mathbf{x} \leq \mathbf{cb} < 0$.

Preuve : (i) équivaut à dire que les projections de P sont vides. En appliquant d fois le lemme 6.20, où d est la dimension de \mathbf{x} , on a l'existence d'une matrice C à coefficients non négatifs telle que $Elim_1(Elim_2(\ldots Elim_d(P)\ldots)) = \{CA\mathbf{x} \leq C\mathbf{b}\}$ est vide. Comme $CA = \mathbf{0}$ (on a projeté sur un espace de dimension 0), l'un des vecteurs lignes \mathbf{c} de C vérifie (ii).

Lemme 6.22 Les deux assertions suivantes sont équivalentes (i) $\nexists \mathbf{x} \ge \mathbf{0}$ tel que $A\mathbf{x} = \mathbf{b}$,

(ii) $\exists \mathbf{c} \ tel \ que \ \mathbf{c}A \ge \mathbf{0} \ et \ \mathbf{cb} < 0.$

Preuve : (i) se réécrit $\left\{ \begin{bmatrix} -I \\ A \\ -A \end{bmatrix} \mathbf{x} \le \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \\ -\mathbf{b} \end{bmatrix} \right\}$ est vide. Appliquer alors le lemme 6.21.

Preuve plus géométrique : (i) signifie que **b** n'est pas dans le cône des vecteurs colonnes \mathbf{a}_i de A. Par le lemme de séparation 6.8 appliqué au convexe compact $\{\mathbf{b}\}$ et au convexe fermé $C \hat{o} ne(\{\mathbf{a}_i\})$ ceci entraîne l'existence d'un hyperplan séparateur $H = \{\mathbf{cy} = c_0\}$ tel que

$$\forall \mathbf{x} \geq \mathbf{0} : \mathbf{c} A \mathbf{x} > c_0 \text{ et } \mathbf{c} \mathbf{b} < c_0.$$

En particulier $\mathbf{x} = \mathbf{0}$ implique $\mathbf{cb} < c_0 < 0$. De plus on ne peut avoir $\mathbf{ca_i} < 0$ car pour \mathbf{x} ayant la *i*-ème coordonnée positive assez grande on aurait $\mathbf{cAx} < c_0$. D'où $\mathbf{cA} \ge \mathbf{0}$. \Box

Noter que cette proposition exprime qu'un vecteur est soit dans un cône engendré par une famille finie donnée de vecteurs soit séparé de cette famille par un hyperplan. Pour une démonstration directe voir :

Theory of linear and integer programming. Alexander Schrijver. Wiley, 1986. page 85.

Lemme 6.23 Les deux assertions suivantes sont équivalentes

(i) {ax ≤ b₀} est valide pour {Ax ≤ b},
(ii) ∃c ≥ 0 tel que cA = a et cb ≤ b₀ ou tel que cA = 0 et cb < 0.

Dit autrement, si une inéquation est valide pour un polyèdre non vide, alors cette inéquation est impliquée par une combinaison à coefficients positifs des inéquations du polyèdre.

Preuve : (ii) \Rightarrow (i) : facile. non (ii) \Rightarrow $\nexists (c_0 \mathbf{c}) \ge \mathbf{0}$ tel que $(c_0 \mathbf{c}) \begin{bmatrix} -\mathbf{a} \\ A \end{bmatrix} = \mathbf{0}$ et $(c_0 \mathbf{c}) \begin{bmatrix} -b_0 \\ \mathbf{b} \end{bmatrix} < 0.$ On en déduit par la proposition 6.21 :

$$\exists \mathbf{w} \text{ tel que } A\mathbf{w} \leq \mathbf{b} \text{ et } \mathbf{aw} \geq b_0.$$

On a encore non (ii) \Rightarrow

$$\nexists (c_0 \mathbf{c}) \ge \mathbf{0} \text{ tel que } (c_0 \mathbf{c}) \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{b} & A \end{bmatrix} = (b_0 \mathbf{a})$$

Et on en déduit par la proposition 6.22:

$$\exists \begin{bmatrix} y_0 \\ \mathbf{y} \end{bmatrix} \text{ tel que } \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{b} & A \end{bmatrix} \begin{bmatrix} y_0 \\ \mathbf{y} \end{bmatrix} \ge \mathbf{0} \text{ et } (b_0 \mathbf{a}) \begin{bmatrix} y_0 \\ \mathbf{y} \end{bmatrix} < 0.$$

Selon que y_0 est positif ou nul on en déduit \mathbf{y} tel que $A\mathbf{y} \leq \mathbf{b}$ et $\mathbf{ay} > b_0$ ou bien $A\mathbf{y} \leq \mathbf{0}$ et $\mathbf{ay} > 0$. Dans le premier cas c'est terminé, dans le second $\mathbf{w} + \mathbf{y}$ permet également de contredire (i).

Références :

- A Simple Proof of Farka's Lemma. V. Komornik. The American Mathematical Monthly. Dec. 1998, pp. 949-950.

6.5 Faces d'un polytope

Je suis fidèlement le chapitre 2 de Lectures on Polytopes. Günter Ziegler. Springer GTM 152, 1994.

Soit P un polytope.

Définition 6.24 Une face de P est soit P lui-même soit l'intersection de P avec un hyperplan valide. Un tel hyperplan est dit support de la face. La dimension d'une face est celle de son enveloppe affine. Une face de dimension 0 (resp. 1, resp. k, resp. $\dim(P)-1$) est appelée un sommet (resp. une arête, resp. une k-face, resp. une facette).

On note V(P) les sommets de P.

Définition 6.25 Un point \mathbf{v} de P est extrême s'il n'est pas combinaison convexe d'autres points de P. Si P = Conv(S), pour un ensemble S de points, cela équivaut d'après le lemme 6.5 à $\mathbf{v} \notin Conv(S \setminus \mathbf{v})$.

Lemme 6.26 Un point de P est extrême si et seulement si c'est un sommet de P.

Preuve : Soit P = Conv(S) et soit $\mathbf{v} = P \cap \{\mathbf{cx} = c_0\}$ un sommet de P. Tout point \mathbf{x} de P distinct de \mathbf{v} vérifie $\mathbf{cx} < c_0$, donc \mathbf{v} ne peut être combinaison convexe d'autres points de P. Réciproquement, supposons que \mathbf{v} est un point extrême de P, i.e. que $\mathbf{v} \notin Conv(S \setminus \mathbf{v})$. Par le lemme de séparation on en déduit un hyperplan $\{\mathbf{cx} = c_0\}$ tel que $\mathbf{cv} > c_0$ et $\mathbf{cs} < c_0$ pour $\mathbf{s} \in S \setminus \mathbf{v}$. L'hyperplan $\{\mathbf{cx} = \mathbf{cv}\}$ est un hyperplan valide définissant le sommet \mathbf{v} .

Proposition 6.27

- (i) Si P est l'enveloppe convexe d'un ensemble fini de points, alors cet ensemble contient V(P).
- (ii) P est l'enveloppe convexe de ses sommets.

Preuve : (i) est une conséquence directe du lemme 6.26. Soit P = Conv(S). Si $\mathbf{v} \in S \setminus V(P)$, alors $P = Conv(S \setminus \mathbf{v})$ d'après la remarque 6.3 et on en déduit (ii) par récurrence sur $|\mathbf{S}|$.

Puisqu'un polytope est une intersection bornée de demi-espaces, il est clair que toute face d'un polytope est elle-même un polytope.

Proposition 6.28 Soit P un polytope et F une face de P.

- (i) l'intersection de deux faces de P est une face de P,
- (ii) les faces de F sont les faces de P incluses dans F, en particulier, $V(F) = V(P) \cap F$, et
- (iii) $F = P \cap aff(F)$.

Preuve : (i) : Soit H un hyperplan support de F et soit F' une autre face de P d'hyperplan support H'. Alors toute combinaison positive de H et H' définit un hyperplan support pour $F \cap F'$.

(ii) : Soit F' une face de F d'hyperplan support H' (pour F). Il est facile de choisir un hyperplan support de F' pour P de la forme $H + \lambda H'$ avec λ tel que l'inégalité associée soit stricte pour les sommets de $V(P) \setminus V(F)$: si H' n'est pas support de P alors $\lambda = -\nu/\mu$ convient, où $\nu = \max_{\mathbf{v} \in V(P) \setminus V(F)} H(\mathbf{v})$ et $\mu = \max_{\mathbf{v} \in V(P)} H'(\mathbf{v})$. Si H' est support de P, cf. (i).

(iii):
$$F \subset P \cap aff(F) \subset P \cap H = F$$
.

Définition 6.29 Soit \mathbf{v} un sommet de P d'hyperplan support $H_0 = \{\mathbf{cx} = c_0\}$. Soit $c_1 < c_0$ tel que $\{\mathbf{cx} < c_1\}$ pour les sommets $V(P) \setminus \mathbf{v}$. Je note $H_1 = \{\mathbf{cx} = c_1\}$. L'étoile de \mathbf{v} est par définition

$$P/\mathbf{v} = P \cap H_1$$

Proposition 6.30 L'application qui associe à toute k-face, F, de P contenant \mathbf{v} la (k-1)-face $F \cap H_1$ de P/\mathbf{v} est une bijection d'inverse

$$F' \mapsto P \cap aff(F' \cup \{\mathbf{v}\}).$$

Preuve : Remarquons d'abord que $F \cap H_1$ est bien une (k-1)-face de $P/\mathbf{v} : F \cap H_1 = P \cap H \cap H_1 = P/\mathbf{v} \cap H$ où H est un hyperplan support de F.

L'application inverse est également bien définie : soit H' un hyperplan support d'une face F' de P/\mathbf{v} . Soit λ tel que $\mathbf{v} \in H' + \lambda H_1$ (ce qui est possible puisque $H_1(v) > 0$). Alors

 $H' + \lambda H_1$ est un hyperplan valide pour P. Pour le voir on prend \mathbf{v}' dans $V(P) \setminus \mathbf{v}$ et on pose

$$\mathbf{v}'' = t\mathbf{v}' + (1-t)\mathbf{v} \text{ avec } 1 \ge t = \frac{c_0 - c_1}{c_0 - \mathbf{cv}'} > 0.$$

Alors $\mathbf{v}'' \in P \cap H_1 = P/\mathbf{v}$ et $(H' + \lambda H_1)(\mathbf{v}'') \leq 0$ donc $(H' + \lambda H_1)(\mathbf{v}') \leq 0$. De plus, si $\mathbf{v}' \in P \cap (H' + \lambda H_1)$ alors $(H' + \lambda H_1)(\mathbf{v}'') = 0$ et comme $H_1(\mathbf{v}'') = 0$, on en déduit $H'(\mathbf{v}'') = 0$. Autrement dit $\mathbf{v}'' \in F'$, d'où $\mathbf{v}' \in aff(F' \cup \{\mathbf{v}\})$ et $P \cap aff(F' \cup \{\mathbf{v}\}) =$ $P \cap (H' + \lambda H_1)$ est bien une face de P.

Vérifions que les deux applications sont bien inverses l'une de l'autre :

$$P \cap aff((F \cap H_1) \cup \{\mathbf{v}\}) = P \cap aff(F) = F.$$

La première égalité provient du fait que tout point de F est combinaison affine de \mathbf{v} et d'un point de $F \cap H_1$. On remarque en passant que $dim(F) = dim(F \cap H_1) + 1$ puisque $\mathbf{v} \notin aff(F \cap H_1)$.

6.5.1 Terminologie

Un ensemble partiellement ordonné, ou poset, est une relation antisymétrique, transitive et réflexive sur un ensemble fini. Deux éléments en relation sont dits comparables. On appelle ordre la relation d'un poset. Si le couple (a, b) appartient à la relation d'un poset on dit que l'élément a est plus petit que l'élément b. Le poset opposé est défini par l'ordre inverse. Un ordre est total ou linéaire si deux éléments quelconques sont comparables. On considère ci-dessous les relations plus grand et plus petit au sens large.

Définition 6.31 Une chaîne d'un poset est un sous-poset totalement ordonné. La longueur d'une chaîne est son nombre d'éléments moins un. Un intervalle [a, b] entre deux éléments a et b est l'ensemble des éléments plus grands que a et plus petits que b. Une borne inférieure (resp. supérieure) d'une partie X d'un poset est un élément plus petit (resp. plus grand) que tout élément de X et plus grand (resp. plus petit) que tout élément ayant cette propriété. Un poset est borné s'il admet un plus petit élément et un plus grand élément. Un treillis est un poset borné tel que toute paire d'éléments admet une borne inférieure appelée meet (\wedge) et une borne supérieure appelée join (\vee). Un poset est gradué si la longueur de toute chaîne maximale dont le plus grand élément est fixé ne dépend que de cet élément. Cette longueur est alors appelée le rang de cet élément. Dans un treillis gradué les éléments de rang 1 sont appelés atomes et ceux de rang un de moins que l'élément maximal sont appelés coatomes. Un treillis est atomique (coatomique) si tout élément est un join (meet) d'atomes (de coatomes). Un élément b est dit successeur d'un élément a si l'intervalle [a, b] est précisément la paire $\{a, b\}$. Le diagramme de Hasse d'un poset est un dessin dans le plan de sa relation successeur où les ordonnées des points représentant les éléments sont dans un ordre compatible avec l'ordre du poset.

Exercice 6.32 Montrer que pour tout treillis X, le meet et le join sont associatifs. En déduire que (X, \vee) et (X, \wedge) sont des semi-groupes commutatifs.

Exemples de treillis : les entiers de 0 à N avec la relation d'ordre usuelle. Les treillis booléens, B_k , i.e. les parties d'un ensemble à k éléments ordonnées par l'inclusion ($\lor = \cup$, $\land = \cap$). L'ensemble des diviseurs d'un entier pour la relation de divisibilité ($\lor = ppcm$, $\land = pgcd$).

6.5.2 Treillis des faces d'un polytope

On considère l'ensemble $\mathcal{F}(P)$ des faces de P partiellement ordonnées par l'inclusion.

Proposition 6.33

- (i) $\mathcal{F}(P)$ est un treillis gradué de longueur dim(P)+1 et de fonction rang $(F) = \dim(F)+1$, atomique et coatomique. En particulier $F \wedge F' = F \cap F'$.
- (ii) Tout intervalle [G, F] est le treillis d'un polytope de dimension $\dim(F) \dim(G) 1$.
- (iii) (propriété du carreau) Tout intervalle [G, F] de longueur 2, avec $G \subset F$, a exactement 4 éléments et son treillis est isomorphe à B_2 .
- (iv) Le treillis opposé de $\mathcal{F}(P)$ est le treillis des faces d'un polytope.

Preuve : (i) : vide (resp. P) est un plus petit (resp. plus grand) élément pour $\mathcal{F}(P)$. D'après la proposition 6.28(i) l'intersection de deux faces est un minorant de ces faces dans $\mathcal{F}(P)$ et c'est évidemment le plus grand, ce qui définit le meet. On vérifie aisément (exercice) qu'un poset borné possédant un meet est un treillis. Vérifions que ce treillis est gradué : si $G \subsetneq F$ alors par la proposition 6.28(iii) on a $G = P \cap aff(G) \subsetneq P \cap aff(F) =$ F. Donc $aff(G) \subsetneq aff(F)$ d'où dim(G) < dim(F). Il suffit alors de vérifier que si $G \subset F$ avec dim(G) < dim(F) - 1, alors il existe une face H telle que $G \subsetneq H \subsetneq F$. Cela découle de la propriété (ii), prouvée ci-après, car [G, F] est le treillis d'un polytope de dimension au moins 1, qui contient au moins un sommet, fournissant ainsi H. Les propositions 6.27(ii) et 6.28(ii) montrent que $\mathcal{F}(P)$ est atomique. Enfin (iv) permet de conclure que $\mathcal{F}(P)$ est également coatomique.

(ii) : D'après la proposition 6.28(iii), on peut supposer F = P. La propriété est vraie si $G = \emptyset$. Supposons $G \neq \emptyset$. Alors G a un sommet **v** par la propriété 6.27(ii) qui est un sommet de P par 6.28(ii). De plus le treillis de P/\mathbf{v} est isomorphe à l'intervalle $[\mathbf{v}, P]$ par la proposition 6.30 ce qui permet de conclure par récurrence sur dim(G) (ou dim(P)).

(iii) : Appliquer (ii), en remarquant qu'un 1-polytope est un segment.

(iv) : Le treillis opposé est le treillis du polytope polaire introduit dans la section 6.5.3 suivante. $\hfill\square$

Définition 6.34 Deux polytopes sont dits combinatoirement équivalents si les treillis de leurs faces sont isomorphes.

Lemme 6.35 Deux polytopes P et Q sont combinatoirement équivalents si et seulement si il existe une bijection ϕ entre V(P) et V(Q) qui envoie les sommets de chaque facette de P sur les sommets d'une facette de Q et réciproquement. **Preuve :** Puisque $\mathcal{F}(P)$ est atomique les faces de P s'identifient à des sous-ensembles de V(P) et d'après la proposition 6.28(i) $V(F \wedge F') = V(F) \cap V(F')$. La bijection ϕ se prolonge donc en un isomorphisme entre $\mathcal{F}(P)$ et $\mathcal{F}(Q)$ en définissant $\phi(F)$ par la face de Q de sommets $\phi(V(F))$. En effet, puisque $\mathcal{F}(P)$ est coatomique, on a $F = \bigwedge_{i=1}^{k} F_i$ pour des facettes F_i de P. D'où $V(F) = \bigcap_{i=1}^{k} V(F_i)$ et $\phi(V(F)) = \bigcap_{i=1}^{k} \phi(V(F_i))$ soit encore $\phi(F) = \bigwedge_{i=1}^{k} \phi(F_i)$. Mais ceci montre que ϕ préserve l'ordre car $F' \leq F \Leftrightarrow F \wedge F' = F$. \Box

Lemme 6.36 Soit $P \in \mathbb{R}^d$ un polytope de dimension d, et soit $\mathbf{y} \in P$. On a les équivalences :

(i) y n'est contenu dans aucune face propre de P,

(ii) aucun hyperplan valide pour P ne contient y,

(iii) y est l'isobarycentre de d+1 points de P affinement indépendants.

Preuve : (i) \Leftrightarrow (ii) car $\mathbf{y} \in H$ et H valide pour $P \Leftrightarrow \mathbf{y} \in F = P \cap H$. (iii) \Rightarrow (ii) : Si $\mathbf{y} = \frac{1}{d+1} \sum_{i=1}^{d+1} \mathbf{x}_i$, où les \mathbf{x}_i sont indépendants, et si H est valide pour P, alors $H(\mathbf{y}) = \frac{1}{d+1} \sum_{i=1}^{d+1} H(\mathbf{x}_i) < 0$ car (d+1) points indépendants ne peuvent être dans le même hyperplan.

(ii) \Rightarrow (iii) : $\forall \mathbf{u} \in \mathbb{R}^d, \exists \alpha > 0$ tel que $\mathbf{y} + \alpha \mathbf{u} \in P$. En effet, si $P = \{A\mathbf{x} \leq \mathbf{z}\}$ alors (ii) implique $A\mathbf{y} < \mathbf{z}$ d'où $A(\mathbf{y} + \alpha \mathbf{u}) \leq \mathbf{z}$ pour α suffisamment petit. En choisissant $\mathbf{u} = \mathbf{e_1}, \mathbf{e_2}, \dots, \mathbf{e_d}$ ou $\mathbf{u} = -\sum_i \mathbf{e_i}$ on obtient :

$$\mathbf{y} = \frac{1}{d+1} \left(\sum_{i=1}^{d} (\mathbf{y} + \alpha \mathbf{e_i}) + \left((\mathbf{y} + \alpha (-\sum_{i} \mathbf{e_i}) \right) \right)$$

pour α suffisamment petit.

Les **y** vérifiant le lemme 6.36 sont dits *intérieurs* à P. On note int(P) l'ensemble des points intérieurs à P. Si P est de dimension inférieure à d, on note relint(P) les points intérieurs à P dans l'espace aff(P). Dans ce cadre le lemme reste valide en remplaçant d par dim(P) et (ii) par "un hyperplan valide pour P et contenant **y** contient nécessairement P".

Remarque 6.37 Si P est non vide alors l'isobarycentre de ses sommets est dans son intérieur (relatif).

Remarque 6.38 D'après le lemme 6.36(i) deux faces distinctes de P ont des intérieurs relatifs disjoints. Donc P est l'union disjointe des intérieurs relatifs de ses faces.

6.5.3 Polarité

Cette section peut être sautée si l'on utilise la polarité pour les cônes comme décrit à la section 6.6.2, et si on associe les faces d'un polyèdre à celle d'un cône sur ce polyèdre comme dans la section 6.6.1.

Définition 6.39 Soit H un hyperplan ne contenant pas **0**. On appelle polaire de H la forme linéaire **c** telle que $H = \{\mathbf{cx} = 1\}$. Le polaire (ou dual), P^{Δ} , d'un ensemble $P \subset \mathbb{R}^d$ est l'ensemble des polaires des hyperplans valides pour $P \cup \mathbf{0}$, et ne contenant pas **0**, soit

$$P^{\Delta} = \{ \mathbf{c} \, | \, \forall \mathbf{x} \in P, \mathbf{c}\mathbf{x} \le 1 \}$$

On définit de manière analogue le polaire d'un ensemble de formes linéaires puis le bipolaire par

$$P^{\Delta\Delta} := (P^{\Delta})^{\Delta} = \{ \mathbf{y} \, | \, \mathbf{cx} \le 1 \text{ pour tout } \mathbf{x} \in P \Rightarrow \mathbf{cy} \le 1 \}$$

Dit autrement, le bidual est l'intersection de tous les demi-espaces valides pour $P \cup \mathbf{0}$ (de la forme $\{\mathbf{cx} \leq 1\}$).

Par la suite \mathbb{I} (resp. 1) désigne un vecteur ligne (resp. colonne) de composantes 1. Si A est une matrice de taille $d \times n$, alors Conv(A) désigne l'enveloppe convexe des n vecteurs colonnes de A, vus comme des points de \mathbb{R}^d .

Proposition 6.40 Soient P et Q inclus dans \mathbb{R}^d , alors

- $1. \ P \subset Q \Rightarrow Q^{\Delta} \subset P^{\Delta} \ et \ P^{\Delta\Delta} \subset Q^{\Delta\Delta}.$
- 2. $P \subset P^{\Delta \Delta}$.
- 3. P^{Δ} et $P^{\Delta\Delta}$ sont convexes.
- 4. $\mathbf{0} \in int(P) \Rightarrow P^{\Delta}$ est borné, et P borné $\Rightarrow \mathbf{0} \in int(P^{\Delta})$.
- 5. Si P est convexe, fermé et contient **0**, alors $P = P^{\Delta\Delta}$.
- 6. Si P = Conv(V) est un polytope alors $P^{\Delta} = \{ \mathbf{c} \mid \mathbf{c}V \leq \mathbb{I} \}.$
- 7. Si $P = \{A\mathbf{x} \leq \mathbf{1}\}$ est borné, alors $P^{\Delta} = Conv(A^t)$.

Preuve : 1, 2 et 3 sont faciles.

 $4: B(\mathbf{0}, r) \subset P \Rightarrow P^{\Delta} \subset B(\mathbf{0}, r)^{\Delta} = B(\mathbf{0}, 1/r).$

5. Il suffit de montrer P^{ΔΔ} ⊂ P. Soit x ∉ P, alors puisque P est convexe et fermé, il existe un hyperplan séparateur entre x et P. Mais cela signifie précisément x ∉ P^{ΔΔ}.
6. Clairement un hyperplan est valide pour P ∪ 0 si et seulement s'il l'est pour V ∪ 0. Donc P^Δ = V^Δ = {c | cV ≤ 1}.

7. Si $\mathbf{0} \in Q := Conv(A^t)$ alors, par le point 6, $Q^{\Delta} = P$ puis, par le point 5, $P^{\Delta} = Q^{\Delta \Delta} = Q$. Il suffit donc de vérifier que $\mathbf{0} \in Conv(A^t)$. Mais ceci découle du lemme de Farkas 6.21 car $\{A\mathbf{x} \leq -\mathbf{1}\}$ est vide, puisque P est borné.

Définition 6.41 On définit le dual d'une face F de P par

 $F^{\diamond} = \{ \mathbf{c} \mid \mathbf{cx} \leq 1 \text{ pour tout } \mathbf{x} \in P, \text{ et } \mathbf{cx} = 1 \text{ pour tout } \mathbf{x} \in F \} = \{ \mathbf{c} \mid \mathbf{x} \in F \Rightarrow \mathbf{cx} = 1 \} \cap P^{\Delta}.$

C'est donc l'ensemble des polaires des hyperplans valides pour P qui sont supports de F.

Proposition 6.42 Soit un polytope $P = Conv(V) = \{Ax \leq 1\}$. Supposons que

$$F = conv(V') = \{A''\mathbf{x} \le \mathbf{1} \ et \ A'\mathbf{x} = \mathbf{1}\}$$

soit une face de P avec¹ $V = V' \biguplus V''$ et $A = A' \biguplus A''$. Alors

$$F^{\diamond} = Conv(A'^{t}) = \{ \mathbf{a} \, | \, \mathbf{a}V'' \leq \mathbb{1} \, et \, \mathbf{a}V' = \mathbb{1} \}.$$

Preuve :

$$F^{\diamond} = \{ \mathbf{a} \mid \mathbf{a}\mathbf{x} \leq 1 \text{ pour tout } \mathbf{x} \in P, \text{ et } \mathbf{a}\mathbf{x} = 1 \text{ pour tout } \mathbf{x} \in F \}$$
$$= \{ \mathbf{a} \mid \mathbf{a}V \leq \mathbb{I} \text{ et } \mathbf{a}V' = \mathbb{I} \}$$
$$= \{ \mathbf{a} \mid \mathbf{a}V'' \leq \mathbb{I} \text{ et } \mathbf{a}V' = \mathbb{I} \}$$

On a également

$$F^{\diamond} = \{ \mathbf{a} \mid \mathbf{a}\mathbf{x} \leq 1 \text{ pour } \mathbf{x} \in P, \text{ et } \mathbf{a}\mathbf{x} = 1 \text{ pour tout } \mathbf{x} \in F \}$$

= $\{ \mathbf{c}A \mid \mathbf{c} \geq 0, \mathbf{c}\mathbf{1} = 1 \text{ et } \mathbf{c}A\mathbf{x} = 1 \text{ pour tout } \mathbf{x} \in F \}$ par la prop. 6.40, point 7.
= $\{ \mathbf{c}'A' \mid \mathbf{c}' \geq 0, \mathbf{c'}\mathbf{1} = 1 \}$

Pour la dernière égalité, \supset est facile. Vérifions \subset : soit $\mathbf{x} \in relint(F)$ avec $A'\mathbf{x} = 1$ et $A''\mathbf{x} < 1$. Alors en écrivant $\mathbf{c}A = \mathbf{c}'A' + \mathbf{c}''A''$ on trouve

$$1 = cAx = c'A'x + c''A''x \le c'1 + c''1 = c1 = 1$$

Donc $\mathbf{c}'' A'' \mathbf{x} = \mathbf{c}'' \mathbf{1}$, et comme $A'' \mathbf{x} < 1$ on a $\mathbf{c}'' = 0$.

Corollaire 6.43 Soit P un polytope contenant **0** en son intérieur, et soient F et G deux faces de P, alors

1. F^{\diamond} est une face de P^{Δ} ,

2.
$$F^{\diamond\diamond} = F$$
, et

3. $F \subset G$ si et seulement si $G^{\diamond} \subset F^{\diamond}$.

Corollaire 6.44 Le treillis des faces du polaire d'un polytope est l'opposé du treillis de ses faces.

6.6 Faces d'un cône

On rappelle qu'un cône polyédrique est décrit de manière équivalente par une intersection d'un nombre fini de demi-espaces vectoriels ou par une enveloppe cônique d'un nombre fini de vecteurs (cf. théorème 6.12).

Un demi-espace vectoriel $\{\mathbf{x} \mid \mathbf{cx} \leq 0\}$ contenant un cône C est dit valide pour C. Par extension, on dit que l'hyperplan $\{\mathbf{x} \mid \mathbf{cx} = 0\}$ est valide pour C si le demi-space $\{\mathbf{x} \mid \mathbf{cx} \leq 0\}$ est valide pour C.

La dimension d'un cône est la dimension de l'espace vectoriel engendré, i.e. du plus petit espace vectoriel le contenant. Un cône de \mathbb{R}^d est d'intérieur non vide si et seulement si sa dimension est d.

^{1.} La notation $X = X' \biguplus X''$ indique que les lignes ou les colonnes de X (selon le cas) sont l'union des lignes ou des colonnes de X' et X''.

Exercice 6.45 Montrer que le cône $\{A\mathbf{x} \leq \mathbf{0}\}$ est d'intérieur non vide si et seulement si $\{A\mathbf{x} < \mathbf{0}\}$ est non vide.

Lemme 6.46 Soit $C = \{A\mathbf{x} \leq \mathbf{0}\}$ un cône de dimension k < d dans \mathbb{R}^d . On peut extraire une sous-famille A' de d-k vecteurs de A telle que $\{A'\mathbf{x} = \mathbf{0}\}$ soit l'espace engendré par C.

Preuve : Il existe $\mathbf{a} \in A$ tel que $C \subset {\mathbf{ax} = 0}$. Sinon, on pourrait choisir pour chaque $\mathbf{a} \in A$, un $\mathbf{x}_{\mathbf{a}} \in C$ tel que $\mathbf{ax}_{\mathbf{a}} < 0$, mais alors $\mathbf{x} = \sum_{a \in A} \mathbf{x}_{\mathbf{a}} \in C$ et $\mathbf{x} \in {A\mathbf{x} < \mathbf{0}}$, ce qui contredit l'exercice précédent. On raisonne ensuite par récurrence sur d avec la trace des demi-espaces de ${A\mathbf{x} \leq \mathbf{0}}$ dans l'hyperplan ${\mathbf{ax} = 0}$.

Une face d'un cône C est l'intersection de C avec un hyperplan valide. En particulier, une face d'un cône est un cône. Une face propre de C est une face de C non triviale (qui contient un vecteur non nul) et distincte de C. L'intérieur relatif d'une face est l'intérieur de cette face dans l'espace vectoriel qu'elle engendre. Une face est non triviale si et seulement si son intérieur est non trivial.

Proposition 6.47 Soit C un cône polyédral. L'intersection de deux faces de C est une face de C. Les faces d'une face F de C sont les faces de C incluses dans F.

Preuve : Adapter la preuve de la proposition 6.28.

Exercice 6.48 Montrer que si $\{A'\mathbf{x} = \mathbf{0} \text{ et } A''\mathbf{x} < \mathbf{0}\}\$ est non vide, alors c'est l'intérieur relatif de $\{A'\mathbf{x} = \mathbf{0} \text{ et } A''\mathbf{x} \leq \mathbf{0}\}\$ dans le sous-espace $\{A'\mathbf{x} = \mathbf{0}\}.$

Lemme 6.49 Si { $\mathbf{ax} \leq \mathbf{0}$ } est valide pour $C = \{A\mathbf{x} \leq \mathbf{0}\}$, alors il existe $\lambda \geq \mathbf{0}$ tel que $a = \lambda A$.

Preuve : Appliquer le lemme 6.23.

Proposition 6.50 Soit $C = \{A\mathbf{x} \leq \mathbf{0}\}$ un cône polyédral. L'intérieur relatif de toute face non triviale de C est de la forme

 $\{A'\mathbf{x} = \mathbf{0} \text{ et } A''\mathbf{x} < \mathbf{0}\}, \text{ où } A = A' \biguplus A'' \text{ désigne une partition des lignes de A. Récipro$ $quement, tout ensemble non vide de la forme <math>\{A'\mathbf{x} = \mathbf{0} \text{ et } A''\mathbf{x} < \mathbf{0}\}$ avec $A = A' \biguplus A''$ est l'intérieur relatif d'une face non triviale de C.

Preuve : Soit $F = {\mathbf{ax} = \mathbf{0}} \cap C$ une face non triviale de C. Par le lemme 6.49, il existe $\lambda > \mathbf{0}$ tel que $\mathbf{a} = \lambda A_a$, pour un sous-ensemble A_a des lignes de A. Soit A' l'ensemble des lignes de A telles que $A'F = \mathbf{0}$, et soit $A'' = A \setminus A'$. Puisque $A_aF \leq \mathbf{0}$ et $\mathbf{a}F = \lambda A_aF = \mathbf{0}$ il suit que $A_a \subset A'$. On a donc

$$\{A'\mathbf{x} = \mathbf{0} \text{ et } A''\mathbf{x} < \mathbf{0}\} \subset F \subset \{A'\mathbf{x} = \mathbf{0} \text{ et } A''\mathbf{x} \le \mathbf{0}\}$$

Or $I := \{A'\mathbf{x} = \mathbf{0} \text{ et } A''\mathbf{x} < \mathbf{0}\}$ n'est pas vide. En effet, pour tout $\mathbf{a}'' \in A''$ il existe $\mathbf{x}_{\mathbf{a}''} \in F$ tel que $\mathbf{a}''\mathbf{x}_{\mathbf{a}''} < 0$. Soit $\mathbf{x} = \sum_{a''} \mathbf{x}_{\mathbf{a}''}$. On a $A''\mathbf{x} < 0$ et par convexité, $\mathbf{x} \in F$, d'où $A'\mathbf{x} = \mathbf{0}$ et finalement $\mathbf{x} \in I$. On en déduit par l'exercice 6.48 que I est l'intérieur (relatif) de F.

La réciproque est laissée en exercice.

Proposition 6.51 Soit $C = \{R\lambda \mid \lambda \ge 0\}$ un cône polyédral. Un sous-ensemble $F \subset C$ est une face de C si et seulement s'il existe une partition $R = R' \biguplus R''$ des colonnes de R et un vecteur **a** tels que

$$\mathbf{a}R' = \mathbf{0}, \mathbf{a}R'' < \mathbf{0} \ et \ F = \{R'\lambda' \mid \lambda' \ge \mathbf{0}\}$$

Preuve : Soit *F* une face de *C*, alors il existe un hyperplan valide $H := \{\mathbf{x} \mid \mathbf{ax} = 0\}$ tel que $F = C \cap H$. Soient $R' = R \cap H$ et $R'' = R \setminus R'$, de sorte que $\mathbf{a}R' = \mathbf{0}$ et $\mathbf{a}R'' < \mathbf{0}$. Tout $\mathbf{x} \in C$ étant de la forme $\mathbf{x} = R'\lambda' + R''\lambda''$ avec $\lambda', \lambda'' \ge \mathbf{0}$, l'équation $\mathbf{ax} = 0$ équivaut à $\mathbf{x} = R'\lambda'$ avec $\lambda' \ge \mathbf{0}$. On en déduit $F = C \cap H = \{R'\lambda' \mid \lambda' \ge \mathbf{0}\}$. Réciproquement, soit R', R'' et \mathbf{a} comme dans la proposition, alors il est immédiat que H est valide pour C et que $\{R'\lambda' \mid \lambda' \ge \mathbf{0}\} = C \cap H$ est une face de C.

6.6.1 Faces des polyèdres via les cônes

On définit une *face* d'un polyèdre de manière analogue à une face d'un cône ou d'un polytope, comme l'intersection d'un hyperplan valide avec le polyèdre. Sa dimension et celle de son enveloppe affine.

Soit C un cône et H un hyperplan ne contenant pas **0**. On considère l'application ϕ qui associe à toute face F du polyèdre $C \cap H$ le cône sur F de sommet **0**. On considère également l'application ψ qui associe à toute face F de C qui intersecte H l'intersection $F \cap H$.

Lemme 6.52 Les applications ϕ et ψ sont des bijections inverses l'une de l'autre. De plus ϕ augmente la dimension de 1 et préserve la relation d'inclusion : pour toutes faces F, F' de $C \cap H$

$$\dim \phi(F) = \dim F + 1 \ et \ F \subset F' \implies \phi(F) \subset \phi(F')$$

Preuve : Soit D un hyperplan de H support de F pour $C \cap H$. Alors l'enveloppe affine $D' = aff(\mathbf{0} \cup D)$ de $\mathbf{0}$ et D est un hyperplan support de $\phi(F)$ pour C. Réciproquement, soit F une face de C qui intersecte H et soit D un hyperplan support pour F. Alors $D \cap H$ est un hyperplan de H, support de $\psi(F)$ pour $C \cap H$.

6.6.2 Polarité pour les cônes

On considère la dualité sur \mathbb{R}^d , induite par le produit scalaire, qui associe le vecteur **a** à la forme linéaire $\mathbf{x} \mapsto \mathbf{ax}$ et réciproquement. Soit C un cône polyédrique de dimension d, ou de manière équivalente d'intérieur non vide.

Définition 6.53 Le polaire, ou dual, C^* de C est l'ensemble des vecteurs duaux aux formes linéaires négatives sur C. Soit encore,

$$C^* = \{ \mathbf{a} \mid \forall \mathbf{x} \in C, \mathbf{a}\mathbf{x} \le 0 \}$$

La face duale $F^{\#}$ d'une face F de C est l'ensemble des vecteurs duaux aux formes négatives sur C et nulles sur F :

$$F^{\#} = \{ \mathbf{a} \mid \forall \mathbf{x} \in C, \mathbf{ax} \le 0 \ et \ \forall \mathbf{y} \in F, \mathbf{ay} = 0 \}$$

Proposition 6.54 Soit C un cône polyédral d'intérieur non vide.

- 1. Si $C = \{A\mathbf{x} \le \mathbf{0}\}$ alors $C^* = \{\lambda A \mid \lambda \ge \mathbf{0}\}$
- 2. Si $C = \{R\lambda \mid \lambda \ge \mathbf{0}\}$ alors $C^* = \{\mathbf{a}R \le \mathbf{0}\}.$

Preuve : 1) $\lambda \ge \mathbf{0} \implies \forall \mathbf{x} \in C : \lambda A \mathbf{x} \le 0 \implies \lambda A \in C^*$, i.e. $\{\lambda A \mid \lambda \ge \mathbf{0}\} \subset C^*$. Réciproquement, soit $\mathbf{a} \in C^*$. Par le lemme de Farkas 6.23 il existe $\lambda \ge \mathbf{0}$ tel que $\mathbf{a} = \lambda A$. D'où $C^* \subset \{\lambda A \mid \lambda \ge \mathbf{0}\}$.

2)
$$\mathbf{a}R \leq \mathbf{0} \Leftrightarrow \forall \lambda \geq \mathbf{0} : \mathbf{a}R\lambda \leq 0 \Leftrightarrow \forall \mathbf{x} \in C : \mathbf{a}\mathbf{x} \leq 0 \Leftrightarrow \mathbf{a} \in C^*.$$

En particulier le dual d'un cône polyédral est un cône polyédral.

Exercice 6.55 On définit de manière naturelle le bidual C^{**} de C comme le dual du dual de C:

$$C^{**} = \{ \mathbf{z} \mid \forall \mathbf{a} \in C^*, \mathbf{az} \le 0 \}$$

Montrer que $C^{**} = C$.

Lemme 6.56 Soit $C = \{A\mathbf{x} \leq \mathbf{0}\}$ un cône d'intérieur non vide. Si l'intérieur relatif d'une face F est de la forme $\{A'\mathbf{x} = \mathbf{0} \text{ et } A''\mathbf{x} < \mathbf{0}\}$ avec $A = A' \biguplus A''$, alors $F^{\#} = \{\lambda'A' \mid \lambda' \geq \mathbf{0}\}.$

Preuve : $\{\lambda'A' \mid \lambda' \ge \mathbf{0}\} \subset F^{\#}$. En effet, si $\mathbf{a} = \lambda'A'$ avec $\lambda' \ge \mathbf{0}$, on a pour tout $\mathbf{x} \in C$ et tout $\mathbf{y} \in F$:

$$\mathbf{a}\mathbf{x} = \lambda' A' \mathbf{x} \le 0 \text{ et } \mathbf{a}\mathbf{y} = \lambda' A' \mathbf{y} = 0.$$

D'où $\mathbf{a} \in F^{\#}$.

 $F^{\#} \subset \{\lambda'A' \mid \lambda' \geq \mathbf{0}\}$. En effet, pour tout $\mathbf{a} \in F^{\#}$ l'hyperplan $\{\mathbf{x} \mid \mathbf{ax} = 0\}$ est valide pour C. On a donc par le lemme 6.49 $\mathbf{a} = \lambda A = \lambda'A' + \lambda''A''$ pour un certain $\lambda = [\lambda' \lambda''] \geq \mathbf{0}$. Soit \mathbf{y} un point intérieur à F. La condition $\lambda'A'\mathbf{y} + \lambda''A''\mathbf{y} = \mathbf{ay} = 0$ impose $\lambda'' = \mathbf{0}$ puisque $A'\mathbf{y} = \mathbf{0}$ et $A''\mathbf{y} < \mathbf{0}$. On en déduit $\mathbf{a} = \lambda'A'$ avec $\lambda' \geq \mathbf{0}$.

Lemme 6.57 Soit $C = \{R\lambda \mid \lambda \ge \mathbf{0}\}$ un cône d'intérieur non vide. Si F est une face de C de la forme $F = \{R'\lambda' \mid \lambda' \ge \mathbf{0}\}$ avec $R = R' \biguplus R''$, alors $F^{\#} = \{\mathbf{a}R' = \mathbf{0} \text{ et } \mathbf{a}R'' \le \mathbf{0}\}.$ **Preuve :** $\{\mathbf{a}R' = \mathbf{0} \text{ et } \mathbf{a}R'' \leq \mathbf{0}\} \subset F^{\#}$: Soit \mathbf{a} tel que $\mathbf{a}R' = \mathbf{0}$ et $\mathbf{a}R'' \leq \mathbf{0}$. Pour tout $\mathbf{x} \in C$ et $\mathbf{y} \in F$, on a $\mathbf{x} = R'\lambda' + R''\lambda''$ et $\mathbf{y} = R'\mu'$ pour certains $\lambda', \lambda'', \mu' \geq \mathbf{0}$. On en déduit $\mathbf{a}\mathbf{x} = \mathbf{a}R'\lambda' + \mathbf{a}R''\lambda'' \leq 0$ et $\mathbf{a}\mathbf{y} = \mathbf{a}R'\mu' = 0$, d'où $\mathbf{a} \in F^{\#}$.

 $F^{\#} \subset \{\mathbf{a}R' = \mathbf{0} \text{ et } \mathbf{a}R'' \leq \mathbf{0}\}: \text{Si } \mathbf{a} \in F^{\#}, \text{ alors pour tout } \lambda, \lambda' \geq \mathbf{0} \text{ on doit avoir } \mathbf{a}R\lambda \leq 0$ et $\mathbf{a}R'\lambda' = 0$. On en déduit $\mathbf{a}R' = \mathbf{0}$ et $\mathbf{a}R'' \leq \mathbf{0}$.

Proposition 6.58 Soit C un cône polyédral de dimension d dans \mathbb{R}^d . La correspondance qui associe à une face F de C sa face duale $F^{\#}$ établit une bijection entre les faces de dimension k de C et de codimension k de C^* . Cette dualité renverse l'inclusion.

Preuve : On peut écrire $C = \{A\mathbf{x} \leq \mathbf{0}\}$. Par la proposition 6.54, $C^* = \{\lambda A \mid \lambda \geq \mathbf{0}\}$.

Par la proposition 6.50, l'intérieur relatif de toute face F non triviale de C est de la forme (unique)

 $Int_{rel}(F) = \{A'\mathbf{x} = \mathbf{0} \text{ et } A''\mathbf{x} < \mathbf{0}\}, \text{ où } A = A' \biguplus A''.$ Par le lemme 6.56, on a $F^{\#} = \{\lambda'A' \mid \lambda' \geq \mathbf{0}\}$. C'est une face de C^* d'après la proposition 6.51, et cette forme est unique de par l'existence d'un \mathbf{x} avec $A'\mathbf{x} = \mathbf{0}$ et $A''\mathbf{x} < \mathbf{0}$. On en déduit que la correspondance $F \mapsto F^{\#}$ est une bijection. De plus,

$$\dim F = \dim Int_{rel}(F) = \dim \ker A'$$

(car $\{A''\mathbf{x} < \mathbf{0}\}$ est un ouvert non vide), et dim $F^{\#} = rang(A') = d - \dim \ker A'$. Par ailleurs, si F_1, F_2 sont deux faces de C associées à des décomposition respectives $A'_1 \biguplus A''_1 = A'_2 \biguplus A''_2 = A$, alors clairement $F_1 \subset F_2 \implies A'_2 \subset A'_1 \implies F_2^{\#} \subset F_1^{\#}$. \Box

Exercice 6.59 Soit P un polytope de \mathbb{R}^d contenant **0**, et soit C le cône de \mathbb{R}^{d+1} de sommet **0** sur le polyèdre translaté $\mathbf{e}_{d+1} + P$. Montrer que

$$P^{\Delta} = \mathbf{e_{d+1}} + (C^*)^t \cap \{x_{d+1} = -1\}$$

où $(C^*)^t$ est l'ensemble de vecteurs colonnes correspondants aux vecteurs lignes de C^* .

6.7 Exemples de Polytopes

- Simplexes : ce sont les seuls polytopes à la fois simples et simpliciaux.
- Cubes et cocubes : Le cube $C_n = \{\mathbf{x} \mid -1 \leq x_i \leq 1\}$ est l'intersection des demi-espaces $\{\mathbf{x}(\pm \mathbf{e_i}) \leq 1\}$. Par dualité, on obtient le cocube (ou hyperoctaèdre ou encore polytope croisé) $C_n^{\Delta} = Conv(\pm \mathbf{e_i})$. Mais on a aussi $C_n = Conv(\{-1,1\}^d)$ d'où par dualité encore $C_n^{\Delta} = \{\mathbf{x} \mid \sum |x_i| \leq 1\}$.
- Polytopes simples : Chaque sommet est de degré minimal d, i.e. l'étoile d'un sommet est un simplexe. Exemples : tétraèdre, cube, dodécaèdre. L'intersection bornée d'une famille de demi-espaces en position générale, i.e. telle que par tout point il passe au plus d hyperplans bordant les demi-espaces de la famille, est un polytope simple.
- Polytopes simpliciaux : chaque facette (et donc chaque face propre) est un simplexe. Exemples : tétraèdre, octaèdre, icosaèdre. L'enveloppe convexe d'une famille de points en position générale, i.e. telle que tout sous-ensemble de d + 1 points soit affinement indépendant, est un polytope simplicial. Le dual d'un polytope simple est simplicial et réciproquement.

- Produits, pyramides, bipyramides.
- Permutaèdres : Enveloppe convexe des points dont les coordonnées sont les d! permutations de $(1, 2, \ldots, d)$.
- Polytopes des couplages d'un graphe G = (V, E): c'est l'enveloppe convexe des vecteurs d'incidences des couplages de G. Il est défini par le système { $\forall e \in E : x_e \ge 0, \forall v \in V : \sum_{x \in e} x_e \le 1$ }.

6.7.1 Polytopes cycliques

Définition 6.60 La courbe $\gamma: t \Rightarrow (t, t^2, \dots, t^d)$ de \mathbb{R}^d est appelée courbe des moments.

Lemme 6.61 Pour tout entier n et tous réels t_1, t_2, \ldots, t_n , les points $\gamma(t_1), \gamma(t_2), \ldots, \gamma(t_n)$ sont en position générale.

Preuve : L'appartenance d'un point $\gamma(t)$ à un hyperplan quelconque s'exprime par la nullité d'un polynôme de degré au plus d en t qui a au plus d racines. Donc d + 1 points de la courbe des moments ne peuvent être affinement liés.

Définition 6.62 On considère n > d réels $t_1 < t_2 < \ldots < t_n$. On note $C_d(t_1, t_2, \ldots, t_n)$, et appelle polytope cyclique d'ordre n, l'enveloppe convexe des points $\gamma(t_1)$, $\gamma(t_2)$, \ldots , $\gamma(t_n)$ de la courbe des moments dans \mathbb{R}^d . D'après le lemme précédent un polytope cyclique est simplicial.

La proposition ci-après montre que la combinatoire du polytope cyclique d'ordre n dans \mathbb{R}^d ne dépend pas des n points choisis sur la courbe des moments. On note $C_d(n)$ ce polytope.

Proposition 6.63 (Condition de parité de Gale, 1963) Soient $t_1 < t_2 < \ldots < t_n$. On pose $v_i = \gamma(t_i)$ et $V = \{v_1, v_2, \ldots, v_n\} \subset \mathbb{R}^d$. Un sous-ensemble de d points, $F \subset V$, détermine une facette du polytope cyclique $C_d(t_1, t_2, \ldots, t_n)$ si et seulement si pour tout $v_i, v_j \in V \setminus F$ le nombre de sommets de F entre v_i et v_j est pair.

Preuve : L'hyperplan H déterminé par F est tel que $H(\gamma(t)) = \alpha \prod_{i=1}^{d} (t - t_i)$ où α est une constante et les t_i sont les paramètres des points de F. La condition de Gale exprime précisément que $H(v_i)$ et $H(v_j)$ ont même signe i.e. sont du même côté de H. \Box

Corollaire 6.64 Avec les notations de la proposition précédente, l'enveloppe convexe de tout sous-ensemble de $k \leq \lfloor d/2 \rfloor$ sommets de V est une face de dimension k-1 de $C_d(t_1, t_2, \ldots, t_n)$.

Preuve : On peut déduire cette propriété de la proposition précédente. En voici une preuve directe. Soit $T_k := \{t_{i_1}, t_{i_2}, \ldots, t_{i_k}\} \subset T := \{t_1, t_2, \ldots, t_n\}$ et $V_k \subset V$ le sousensemble de sommets correspondant. Alors le polynôme $\prod_{j=1}^k (t - t_{i_k})^2$ est nul sur T_k et strictement positif sur $T \setminus T_k$. Comme ce polynôme a un degré au plus d, ses coefficients déterminent une forme linéaire qui est nulle sur V_k et strictement positive sur $V \setminus V_k$. Cette forme correspond donc à un hyperplan support de $C_d(T)$ qui a son tour détermine la face $Conv(V_k)$. Le lemme 6.61 de position générale indique que cette face a dimension k-1.

En particulier, cette propriété implique que les points d'un polytope cyclique sont en position convexe, i.e. que ses points coïncident avec ses sommets.

Exercice 6.65 Déduire directement le corollaire 6.64 de la proposition 6.63.

Exercice 6.66 Déduire de la proposition 6.63 que la combinatoire d'un polytope cyclique d'ordre n ne dépend pas des valeurs des paramètres des points de γ choisis mais seulement de leur nombre n. Utiliser pour cela le fait que le treillis des faces est coatomique.

Exercice 6.67 Montrer que le nombre de facettes d'un polytope cyclique d'ordre n dans \mathbb{R}^d est

$$\binom{n-d/2}{d/2} + \binom{n-d/2-1}{d/2-1}$$
$$2\binom{n-\lfloor d/2 \rfloor - 1}{\lfloor d/2 \rfloor}$$

si d est pair, et

si d est impair.

6.8 Le théorème de la borne supérieure

Par un argument de perturbation, il n'est pas très difficile de montrer que pour tout polytope P à n sommets de dimension d, il existe un polytope simplicial à n sommets de dimension d qui possède au moins autant de k-faces que P pour tout k. Dit autrement les polytopes simpliciaux maximisent le nombre de k-faces pour un nombre de sommets fixé n. Puisque toutes les faces propres d'un polytope simplicial sont des simplexes, il est clair qu'un tel polytope a au plus $\binom{n}{k+1}$ faces de dimension k. En particulier, ceci fournit un majorant du nombre de facettes de l'ordre de n^d pour n grand devant d. Le théorème de la borne supérieure indique que cette estimation est largement surévaluée et que le nombre de facettes (en fait le nombre total de faces) est de l'ordre $n^{\lfloor d/2 \rfloor}$.

Théorème 6.68 (de la borne supérieure, Mac Mullen, 1970) Tout polytope à n sommets de dimension d a un nombre de k-faces, pour $0 \le k \le d$, majoré par le nombre f_k de k-faces du polytope cyclique d'ordre n dans \mathbb{R}^d .

Définition 6.69 On note $f_k(P)$ le nombre de k-faces d'un polytope P. La liste $(f_0(P), f_1(P), \ldots, f_d(P))$ est appelée le f-vecteur de P.

Voici une version plus faible et beaucoup plus facile à démontrer que le théorème de la borne supérieure.

Proposition 6.70 (version asymptotique, Seidel [Sei95]) Soit P un polytope à n sommets de dimension d, alors

$$f_{d-1}(P) \le 2 \binom{n}{\lfloor d/2 \rfloor},$$
$$\sum_{k=0}^{d} f_k(P) \le 2^{d+1} \binom{n}{\lfloor d/2 \rfloor}$$

À dimension d fixée ces deux dernières quantités sont donc des $O(n^{\lfloor d/2 \rfloor})$.

Proposition 6.71 Soit P un polytope simplicial à n sommets de dimension d, alors

$$f_{d-1}(P) \le 2f_{\lfloor d/2 \rfloor - 1}(P),$$

 $\sum_{k=0}^{d} f_k(P) \le 2^d f_{d-1}(P).$

Preuve : Pour la seconde inégalité remarquer que chaque facette de P est un d-1 simplexe qui a $2^d - 1$ faces propres au total; or toute face propre de P est face d'au moins une facette de P. Pour la première inégalité on passe au dual P^{Δ} de P qui est simple. On considère une direction pour laquelle les sommets de P^{Δ} ont des hauteurs toutes distinctes. Chaque sommet x de P^{Δ} a d voisins dont au moins la moitié est soit plus haute soit plus basse. x est donc le sommet de hauteur minimum ou maximum d'au moins une $\lceil d/2 \rceil$ -face (sur un convexe un extremum local est un extremum). En considérant la relation "être extremum de" entre les sommets et les $\lceil d/2 \rceil$ -faces, on en déduit par double comptage du nombre de relations : $f_0(P^{\Delta}) \leq 2f_{\lceil d/2 \rceil}(P^{\Delta})$. Ceci permet de conclure puisque $f_k(P^{\Delta}) = f_{d-1-k}(P)$.

Preuve de la proposition 6.70 : Si P est simplicial alors chaque $\lfloor d/2 \rfloor - 1$ -face a exactement $\lfloor d/2 \rfloor$ sommets, d'où $f_{\lfloor d/2 \rfloor - 1}(P) \leq \binom{n}{\lfloor d/2 \rfloor}$. La proposition précédente permet de conclure dans le cas simplicial. Si P n'est pas simplicial alors on peut perturber les sommets de P (en position strictement convexe) de manière à obtenir un polyèdre simplicial ayant au moins autant de faces que P dans chaque dimension. Intuitivement, cela correspond à trianguler P sans ajouter de sommet et à perturber les sommets de sorte que leur enveloppe convexe soit combinatoirement équivalente à cette triangulation. \Box

La démonstration du théorème de la borne supérieure nécessite la notion de bonne orientation acyclique que nous introduisons ci-dessous. Par la dualité des polytopes (cf. corollaire 6.44) une borne sur le nombre de k-faces d'un polytope simplicial à n sommets équivant à une borne sur le nombre de (d - k)-faces d'un polytope simple à n facettes. Il s'avère plus simple de travailler avec des polytopes simples (sic). Le théorème 6.68 devient alors

Théorème 6.72 Pour tout polytope P de dimension d à n facettes, et pour tout $k \in [0,d]$:

$$f_k(P) \le f_k(C_d^{\Delta}(n))$$

Il est à noter que par le corollaire 6.64

$$\forall k \ge \lceil d/2 \rceil : f_k(C_d^{\Delta}(n)) = \binom{n}{d-k}$$
(6.1)

6.8.1 Bonne orientation acyclique

Définition 6.73 Une orientation acyclique d'un polytope P est une orientation de ses arêtes telle que son 1-squelette (i.e. le graphe formé de ses arêtes et sommets) ne contienne pas de cycle orienté. Une orientation acyclique est bonne si sa restriction à toute face non vide F de P (y compris P) contient un unique maximum local, c'est-à-dire un unique sommet dont les arêtes incidentes dans F sont toutes entrantes.

On dira qu'une forme linéaire ϕ est *non dégénérée* sur P si sa restriction aux sommets de P est injective. Une telle forme induit une orientation acyclique de P: il suffit d'orienter chaque arête de P de son sommet de plus petite valeur vers son sommet de plus grande valeur pour ϕ .

Lemme 6.74 Toute forme linéaire non dégénérée sur P induit une bonne orientation acyclique de P.

Preuve : Il suffit de montrer que ϕ a un unique maximum local sur P : les faces de Pétant également des polytopes, l'unicité du maximum local s'appliquera directement à la restriction de ϕ à ces faces. Soit \mathbf{v} un sommet de P qui est maximum local pour ϕ . On considère le cône polyédrique C_v défini par les demi-espaces supports des facettes de Pcontenant v. En particulier $P \subset C_v$. Il suit de la propriété 6.30 de l'étoile d'un sommet que C_v est l'enveloppe conique des arêtes incidentes à v. Puisque ϕ est maximale en vsur toute ces arêtes, ϕ est également maximale en v sur C_v et donc sur P. Donc v est l'unique maximum de ϕ sur P par hypothèse de non-dégénérescence.

6.8.2 *h*-vecteur

Dans ce qui suit on suppose que P est un polytope simple de dimension d possédant n facettes. En particulier le degré de chaque sommet dans le 1-squelette de P est exactement d, qui est le minimum possible. On peut le voir en remarquant que l'étoile de chaque sommet de P est un (d - 1)-simplexe. De plus, chaque face de P est également un polytope simple.

Définition 6.75 Si o est une bonne orientation acyclique de P, on note $V_i(o)$ les sommets de P de degré entrant égal à i et on pose $h_i(o) = |V_i(o)|$. **Théorème 6.76** Soit o une bonne orientation acyclique de P. Pour tout $i \in [0, d]$, on a avec la convention usuelle $\binom{k}{i} = 0$ si k < i:

$$f_i(P) = \sum_{0 \le k \le d} \binom{k}{i} h_k(o), \ et$$
(6.2)

$$h_i(o) = \sum_{0 \le k \le d} (-1)^{i+k} \binom{k}{i} f_k(P)$$
 (6.3)

En particulier, $h_i(o)$ est indépendant de o.

On note désormais $h_i(P)$ la valeur commune des $h_i(o)$. La liste $(h_0(P), h_1(P), \ldots, h_d(P))$ est appelée le *h*-vecteur de *P*.

Preuve : Pour $v \in V(P)$ on note $f_i(v)$ le nombre de *i*-faces de P dont v est le maximum (pour o). D'où

$$f_i(P) = \sum_{v \in V(P)} f_i(v) = \sum_{0 \le k \le d} \sum_{v \in V_k(o)} f_i(v)$$

Or une *i*-face incidente à v est déterminée par i arêtes incidentes à v. Pour que v soit maximum dans cette face il faut que ces i arêtes soient entrantes en v ce qui laisse $\binom{k}{i}$ choix possibles si $v \in V_k(o)$. On en déduit (6.2) compte tenu de $h_k(o) = |V_k(o)|$.

L'équation (6.2) implique que la série génératrice du f-vecteur $f(x) = \sum_i f_i(P)x^i$ et la série $h(x) = \sum_i h_i(o)x^i$ sont reliées par

$$f(x) = h(x+1)$$

D'où h(x) = f(x-1), ce qui après développement et identification des termes fournit la relation (6.3)

Théorème 6.77 (Relations de Dehn-Sommerville) Pour tout $i \in [0, d]$:

$$h_i(P) = h_{d-i}(P) \tag{6.4}$$

Preuve : Soit *o* une bonne orientation acyclique de *P* induite par une forme linéaire ϕ . Alors $-\phi$ induit une orientation \bar{o} inverse de *o* d'où $h_i(\bar{o}) = h_{d-i}(o)$ (rappelons que tout sommet est de degré *d* dans un polytope simple). On conclut avec l'indépendance du *h*-vecteur relativement aux orientations.

Lemme 6.78 *Pour toute face* F *de* P *et pour tout* $i \in [0, d]$

$$h_i(F) \le h_i(P)$$

Preuve : Soit $\{\mathbf{x} \mid \phi(\mathbf{x}) = x_0\}$ un hyperplan support de F tel que $\phi(P) \geq x_0$. En particulier $x_0 < \min_{V(P)\setminus V(F)} \phi$. Par perturbation infinitésimale de ϕ on obtient une forme ψ non dégénérée sur P telle que $\max_{V(F)} \psi < \min_{V(P)\setminus V(F)} \psi$. Soit o l'orientation acyclique induite par ψ . Tout sommet $v \in V(F)$ de degré entrant i dans F pour o est également de degré i dans P pour o puisque l'origine w d'une arête entrante de v vérifie $\psi(w) < \psi(v)$ et est donc dans F. On conclut en utilisant à nouveau l'indépendance du h-vecteur par rapport à o.

Corollaire 6.79 (Relation d'Euler)

$$\sum_{0 \le k \le d} (-1)^k f_k(P) = 1$$

Preuve : Par les relations de Dehn-Sommerville $h_0(P) = h_d(P) = 1$. Notons au passage que cela indique que toute bonne orientation acyclique possède non seulement un unique maximum local dans toute face de P (par définition) mais également un unique minimum local. La relation d'Euler se résume alors à la relation (6.3) avec i = 0.

Lemme 6.80 *Pour tout* $i \in [0, d-1]$:

$$\sum_{F \in F_{d-1}(P)} h_i(F) = (d-i)h_i(P) + (i+1)h_{i+1}(P)$$

où $F_{d-1}(P)$ désigne l'ensemble des n facettes de P.

Preuve : On choisit une bonne orientation acyclique o de P et on note $V_i(P)$ l'ensemble des sommets de degré entrant i dans P pour o. De même, si F est une facette de P, $V_i(F)$ désigne l'ensemble des sommets de degré entrant i dans F pour o (restreinte à F).

Pour $v \in V(P)$, on définit $g_i(v)$ comme le nombre de facettes F de P telles que $v \in V_i(F)$. Par double comptage du nombre d'incidences de la relation $\{(v, F) \in V(P) \times F_{d-1}(P) \mid v \in V_i(F)\}$ on obtient

$$\sum_{v \in V(P)} g_i(v) = \sum_{F \in F_{d-1}(P)} h_i(F)$$

Le membre de gauche se décompose comme suit

$$\sum_{0 \le j \le d} \sum_{v \in V_j(P)} g_i(v) = \sum_{v \in V_i(P)} g_i(v) + \sum_{v \in V_{i+1}(P)} g_i(v)$$

En effet, tout sommet v d'une facette F étant de degré d-1 dans cette facette, une seule arête de P en v n'est pas dans F. Selon que cette arête est sortante ou entrante en v on déduit que le degré entrant de v dans P est respectivement le même ou un de plus que dans F.

Mais pour tout $v \in V_i(o)$ on a $g_i(v) = d - i$ car toute facette de degré entrant i en v est déterminée en suppriment une des d - i arêtes sortantes de P en v. Par un raisonnement analogue, pour tout $v \in V_{i+1}(o)$ on a $g_i(v) = i + 1$. On conclut en rappelant que $h_j(P) = |V_j(P)|$ par définition. \Box

Théorème 6.81 (de la borne supérieur pour le *h*-vecteur)

$$h_i(P) \le \binom{n-1 - \max\{i, d-i\}}{\min\{i, d-i\}}$$

Preuve : L'inégalité du lemme 6.78 reportée dans le lemme 6.80 donne $(d-i)h_i(P) + (i+1)h_{i+1}(P) \le nh_i(P)$, soit

$$h_{i+1}(P) \le \frac{n-d+i}{i+1}h_i(P)$$

On en conclut $h_i(P) \leq \binom{n-1-d+i}{i}$ par récurrence sur *i* compte tenu de $h_0(P) = 1$ et on termine pour la preuve à l'aide des relations de Dehn-Sommerville (6.4).

6.8.3 Preuve du théorème de la borne supérieure

Preuve du théorème 6.72 : Par la relation (6.2), il suffit de montrer que la majoration du *h*-vecteur dans le théorème 6.81 est une égalité pour le dual du polytope cyclique d'ordre *n*. En reportant les égalités (6.1) dans (6.3), on a pour $i \ge \lfloor d/2 \rfloor$

$$h_i(C_d^{\Delta}(n)) = \sum_{k=i}^d (-1)^{i+k} \binom{k}{i} \binom{n}{d-k}$$

Ce qui, après manipulation des coefficients binomiaux, donne bien l'égalité du théorème 6.81. Le cas $i \leq \lfloor d/2 \rfloor$ se déduit des relations de Dehn-Sommerville (6.4).

La relation (6.2) donne plus précisément

$$f_i(P) \le \sum_{0 \le k \le d} \binom{k}{i} \binom{n-1 - \max\{k, d-k\}}{\min\{k, d-k\}}$$

d'où $f_i(P) = O(n^{\lfloor d/2 \rfloor})$ pour *d* constant.

6.9 Steinitz,...

Théorème 6.82 (Steinitz, 1922) Tout graphe planaire simple et 3-connexe est le graphe (1-squelette) d'un 3-polytope et réciproquement.

Théorème 6.83 (Balinski, 1961) Le graphe d'un d-polytope est d-connexe.

Chapitre 7

Programmation Linéaire

Un problème de programmation linéaire (PPL) consiste en l'optimisation d'une forme linéaire sur un sous-ensemble de \mathbb{R}^d défini par un ensemble d'équations et inéquations (au sens large : \leq ou \geq) affines. Toute solution de cet ensemble d'(in)équations est dite *admissible* pour le PPL. La *valeur* d'un PPL admettant une solution admissible est l'optimum de la forme linéaire associée sur l'ensemble des solutions admissibles. Une solution est dite *optimale* si elle est admissible et optimise la forme linéaire associée. Tout PPL peut se ramener de manière équivalente à une forme *canonique*

$$\begin{array}{rll} \min \mathbf{c} \mathbf{x} \\ A \mathbf{x} &\geq \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0}. \end{array}$$

ou à une forme standard

$$\begin{array}{rcl} \min \mathbf{c} \mathbf{x} & \\ A \mathbf{x} & = & \mathbf{b} & (E) \\ \mathbf{x} & \geq & \mathbf{0}. \end{array}$$

(Bien sûr les matrices A, \mathbf{c} et \mathbf{b} ne sont pas les mêmes dans les deux formes!). Il suffit de remarquer pour cela que toute inéquation $\mathbf{ax} \ge \mathbf{b}$ peut s'écrire $\mathbf{ax} - x_0 = \mathbf{b}$ où x_0 est une variable supplémentaire non négative et réciproquement que toute équation $\mathbf{ax} = \mathbf{b}$ est équivalente aux deux inéquations $\mathbf{ax} \ge \mathbf{b}$ et $-\mathbf{ax} \ge -\mathbf{b}$. De plus, on peut écrire $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-$ où \mathbf{x}^+ et \mathbf{x}^- sont deux vecteurs non négatifs.

7.1 Dualité de la programmation linéaire

Théorème 7.1 (de dualité de la programmation linéaire) Pour des matrices de dimensions appropriées, on a

$$\max\{\mathbf{cx} \mid A\mathbf{x} \le \mathbf{b}\} = \min\{\mathbf{yb} \mid \mathbf{y} \ge \mathbf{0}, \mathbf{y}A = \mathbf{c}\}$$

pourvu que le min et le max soient pris sur des ensembles non vides.

Preuve : On pose $X = \{A\mathbf{x} \le \mathbf{b}\}$ et $Y = \{\mathbf{y} \ge \mathbf{0}, \mathbf{y}A = \mathbf{c}\}$. On a

 $\forall \mathbf{x} \in X, \forall \mathbf{y} \in Y \, : \, A\mathbf{x} \leq \mathbf{b} \implies \mathbf{y}A\mathbf{x} \leq \mathbf{y}\mathbf{b} \implies \mathbf{c}\mathbf{x} \leq \mathbf{y}\mathbf{b}$

D'où max $\mathbf{c}X \leq \min Y\mathbf{b}$. Il suffit donc de montrer l'existence de $\mathbf{x} \in X$ et $\mathbf{y} \in Y$ tels que $\mathbf{cx} \geq \mathbf{yb}$. Ce qui s'écrit encore

$$\exists \mathbf{x}, \mathbf{y} \text{ tel que} \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & -I \\ \mathbf{0} & A^t \\ \mathbf{0} & -A^t \\ -\mathbf{c} & \mathbf{b}^t \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y}^t \end{bmatrix} \leq \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{c}^t \\ -\mathbf{c}^t \\ \mathbf{0} \end{bmatrix}$$

Par le lemme de Farkas 6.21, la non-existence de tels \mathbf{x}, \mathbf{y} implique

$$\exists [\mathbf{u_1}, \mathbf{u_2}, \mathbf{u_3}, \mathbf{u_4}, u_5] \ge \mathbf{0} \text{ tel que } [\mathbf{u_1}, \mathbf{u_2}, \mathbf{u_3}, \mathbf{u_4}, u_5] \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & -I \\ \mathbf{0} & A^t \\ \mathbf{0} & -A^t \\ -\mathbf{c} & \mathbf{b}^t \end{bmatrix} = \mathbf{0} \text{ et}$$
$$[\mathbf{u_1}, \mathbf{u_2}, \mathbf{u_3}, \mathbf{u_4}, u_5] \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{c}^t \\ -\mathbf{c}^t \\ \mathbf{0} \end{bmatrix} < 0$$

ce qui s'écrit encore $\mathbf{u}_1 A = u_5 \mathbf{c}, (\mathbf{u}_4 - \mathbf{u}_3) A^t = u_5 \mathbf{b}^t - \mathbf{u}_2$ et $\mathbf{u}_1 \mathbf{b} < (\mathbf{u}_4 - \mathbf{u}_3) \mathbf{c}^t$.

- Ou bien $u_5 = 0$ et on en déduit l'existence de $\mathbf{u_1} \ge \mathbf{0}$ et de $\mathbf{v}(=\mathbf{u_4} - \mathbf{u_3})$ tels que

$$\mathbf{u_1}A = \mathbf{0}, \, \mathbf{v}A^t \leq \mathbf{0}, \, \mathbf{u_1}\mathbf{b} < \mathbf{vc}^t$$

Comme X est non vide, on a pour un certain $\mathbf{x} : \mathbf{u_1}\mathbf{b} \ge \mathbf{u_1}A\mathbf{x} = 0$. De même Y est non vide et pour un certain $\mathbf{y} \ge \mathbf{0} : \mathbf{vc}^t = \mathbf{v}A^t\mathbf{y}^t \le 0$. D'où $\mathbf{u_1}\mathbf{b} \ge \mathbf{vc}^t$, une contradiction. - Ou bien $u_5 > 0$ et on en déduit l'existence de $\mathbf{y}(=\mathbf{u_1}/u_5) \ge \mathbf{0}$ et de $\mathbf{x}(=(\mathbf{u_4}-\mathbf{u_3})^t/u_5)$ tels que

$$\mathbf{y}A = \mathbf{c}, A\mathbf{x} \leq \mathbf{b}, \mathbf{y}\mathbf{b} < \mathbf{c}\mathbf{x}$$

ce qui contredit l'hypothèse de non-existence ci-dessus!

Le théorème de dualité de la programmation linéaire a de nombreuses formulations équivalente dont :

Corollaire 7.2 Pour des matrices de dimensions appropriées, on a

$$\max\{\mathbf{cx} \mid \mathbf{x} \ge \mathbf{0}, A\mathbf{x} \le \mathbf{b}\} = \min\{\mathbf{yb} \mid \mathbf{y} \ge \mathbf{0}, \mathbf{y}A \ge \mathbf{c}\}$$
(7.1)

$$\max\{\mathbf{c}\mathbf{x} \mid \mathbf{x} \ge \mathbf{0}, A\mathbf{x} = \mathbf{b}\} = \min\{\mathbf{y}\mathbf{b} \mid \mathbf{y}A \ge \mathbf{c}\}$$
(7.2)

pourvu dans chaque égalité que le min et le max soient pris sur des ensembles non vides.

7.1.1 Application de la dualité

Comme application de la dualité de la programmation linéaire nous donnons une preuve du classique théorème de flot maximum - coupe minimale.

Soit G = (S, A) un graphe orienté. On note respectivement o(a) et t(a) le sommet origine et le sommet terminaison de l'arc a. On se donne pour chaque arc $a \in A$, une *capacité* $c_a \geq 0$, et on considère deux sommets distingués $s, p \in S$, appelés respectivement la source et le puits. Un flot $\mathbf{f} : A \to \mathbb{R}, a \mapsto f_a$ est admissible pour (G, \mathbf{c}, s, p) s'il vérifie les deux conditions suivantes :

1.
$$\forall a \in A : 0 \leq f_a \leq c_a$$

2. $\forall v \in S \setminus \{s, p\} : \sum_{a:o(a)=v} f_a = \sum_{a:t(a)=v} f_a$ (condition de conservation du flot, dite loi des noeuds)

La valeur $v(\mathbf{f})$ d'un flot \mathbf{f} est la différence entre le flot sortant de s et le flot entrant en s:

$$v(\mathbf{f}) = \sum_{a:o(a)=s} f_a - \sum_{a:t(a)=s} f_a$$

Une coupe pour (G, \mathbf{c}, s, p) est une partition $(V, S \setminus V)$ de S telle que $s \in V$ et $p \notin V$. La capacité c(V, W) d'une coupe (V, W) est la somme des capacités des arcs sortant de V:

$$c(V,W) = \sum_{\substack{a:o(a) \in V, \\ t(a) \in W}} c_a$$

On définit le flot d'une coupe (V, W) par

$$f(V,W) = \sum_{\substack{a:o(a)\in V,\\t(a)\in W}} f_a - \sum_{\substack{a:o(a)\in W,\\t(a)\in V}} f_a$$

et le flot d'un sommet v par

$$f(v) = \sum_{a:o(a)=v} f_a - \sum_{a:t(a)=v} f_a$$

Ainsi pour tout flot admissible $f(s) = v(\mathbf{f})$ et f(u) = 0 si $u \neq s, p$. On vérifie aisément que pour toute coupe (V, W):

$$f(V,W) = \sum_{v \in V} f(v) = f(s) = v(\mathbf{f})$$
(7.3)

Théorème 7.3 (flot max - coupe min, Ford - Fulkerson, 1956) La valeur maximale des flots admissibles est égale à la capacité minimale des coupes.

Nous donnons ci-dessous une preuve consistant à appliquer la dualité à un programme linéaire équivalent au calcul du flot maximal¹, puis à montrer que l'on peut se restreindre

^{1.} Il existe des preuves directes de ce résultat sans utiliser la dualité de la programmation linéaire.

aux solutions entières du problème dual. Ce dernier programme entier s'interprète alors comme le calcul de la coupe minimale.

En interprétant \mathbf{f} et \mathbf{c} comme des vecteurs de \mathbb{R}^A de composantes respectives f_a et c_a , le calcul du flot maximal est modélisé par le PPL suivant :

$$\max v(\mathbf{f}) \quad \mathbf{f} \leq \mathbf{c} \\ \forall v \in S \setminus \{s, p\} : \sum_{a:o(a)=v} f_a = \sum_{a:t(a)=v} f_a \quad (7.4) \\ \mathbf{f} \geq \mathbf{0}$$

Soit $\Pi_{s,p}$ l'ensemble des chemins simples de G joignant s à p. On note x un vecteur de composantes indexées par les chemins de $\Pi_{s,p}$ et on considère le PPL

$$\max \sum_{\pi \in \Pi_{s,p}} x_{\pi}$$
$$\forall a \in A : \sum_{\pi: a \in \pi} x_{\pi} \le c_{a}$$
$$\mathbf{x} \ge \mathbf{0}$$
(7.5)

Lemme 7.4 La valeur du PPL (7.4) est égale à la valeur du PPL (7.5).

Preuve : Notons que les deux PPLs admettent **0** comme solution admissible et sont trivialement bornés. Soit $\mathbf{x} \in \mathbb{R}^{\prod_{s,p}}$ une solution admissible pour (7.5). On vérifie aisément que le flot $f_a = \sum_{\pi:a \in \pi} x_{\pi}$ est admissible pour (7.4) et que $v(\mathbf{f}) = \mathbf{I}\mathbf{x}$. On en déduit valeur(7.5) \leq valeur(7.4). Inversement, soit \mathbf{f} un flot admissible pour (7.4) avec $v(\mathbf{f}) > 0$. Soit G_f le sous-graphe de G restreint aux arcs a tels que $f_a > 0$, et soit S_f l'ensemble des sommets atteignables depuis s dans G_f . Si $p \notin S_f$, alors S_f induit une coupe de flot négatif ou nul (les arcs sortants ont un flot nul). D'après l'équation (7.3), ceci est en contradiction avec l'hypothèse $v(\mathbf{f}) > 0$. Donc $p \in S_f$, i.e. il existe un chemin $\pi \in \prod_{s,p}$ dont tous les arcs ont un flot strictement positif. Soit x_{π} la valeur minimale du flot sur les arcs de π . On considère le flot \mathbf{f}' :

$$f'_a = \begin{cases} f_a - x_\pi & \text{si } a \in \pi \\ f_a & \text{sinon} \end{cases}$$

Clairement, \mathbf{f}' est un flot admissible et $v(\mathbf{f}') < v(\mathbf{f})$. Puisque π a un unique arc sortant de s at aucun arc entrant dans s on a

$$v(\mathbf{f}) = v(\mathbf{f}') + x_{\pi}$$

On peut itérer ce procédé en partant de \mathbf{f}' au lieu de \mathbf{f} . Puisque le graphe $G_{f'}$ contient au moins un arc de moins que G_f , la valeur du flot obtenu par itération du procédé doit s'annuler au bout d'un nombre fini k d'étapes. On a alors obtenu une famille de chemins distincts $\pi, \pi', \ldots, \pi^{(k)}$ de $\Pi_{s,p}$ et des valeurs $x_{\pi}, x_{\pi'}, \ldots, x_{\pi^{(k)}}$ correspondantes. En posant $x_{\gamma} = 0$ pour les chemins γ hors de cette famille, on obtient une solution admissible **x** pour (7.5) telle que

$$v(\mathbf{f}) = v(\mathbf{f}^{(\mathbf{k})}) + \sum_{i=0}^{k} x_{\pi^{(i)}} = \mathbb{I}\mathbf{x}$$

(car $v(\mathbf{f}^{(\mathbf{k})}) = 0$). On en déduit valeur(7.4) \leq valeur(7.5), et finalement valeur(7.5) = valeur(7.4).

Par l'équation (7.1) de dualité de la programmation linéaire, le PPL (7.5) a même valeur que le PPL suivant, où l'on a posé $v(\mathbf{y}) = \sum_{a \in A} c_a y_a$:

$$\min v(\mathbf{y}) \quad \forall \pi \in \Pi_{s,p} : \sum_{a:a \in \pi} y_a \ge 1 \quad (7.6)$$
$$\mathbf{y} \ge \mathbf{0}$$

Lemme 7.5 Le PPL (7.6) a même valeur que sa restriction $(7.6)_{\mathbb{Z}}$ à des vecteurs y entiers.

On en déduit une

Preuve du théorème 7.3 : Par les lemmes 7.4 et 7.5, il suffit de montrer que la valeur de $(7.6)_{\mathbb{Z}}$ est la capacité minimale de toute coupe. Soit (V, W) une coupe. On pose $\mathbf{y} = (y_a)_{a \in A}$ avec

$$y_a = \begin{cases} 1 & \text{si } o(a) \in V \text{ et } t(a) \in W \\ 0 & \text{sinon} \end{cases}$$

Alors \mathbf{y} est une solution admissible pour $(7.6)_{\mathbb{Z}}$ et $v(\mathbf{y}) = c(V, W)$. On en déduit que la valeur de $(7.6)_{\mathbb{Z}}$ est inférieure à la capacité minimale des coupes. Inversement, soit \mathbf{y} une solution admissible pour $(7.6)_{\mathbb{Z}}$. On définit V comme l'ensemble des sommets atteignables depuis s en utilisant des arcs tels que $y_a = 0$. La condition $\sum_{a:a\in\pi} y_a \ge 1$ indique que $p \notin V$, donc V induit une coupe. De plus,

$$c(V, S \setminus V) = \sum_{\substack{a:o(a) \in V, \\ t(a) \notin V}} c_a \le \sum_{\substack{a:o(a) \in V, \\ t(a) \notin V}} c_a y_a \le v(y)$$

Il suit que la capacité minimale des coupes est majorée par la valeur de $(7.6)_{\mathbb{Z}}$, d'où l'égalité de ces deux grandeurs. \Box

Il ne reste plus qu'à donner une

Preuve du lemme 7.5 : Une première preuve utilise le fait que la matrice des inéquations du PPL (7.6) est totalement unimodulaire (cf [MG07, p.144-145]). On en donne une preuve directe. Notons que $\mathbf{y} = \mathbf{1}$ est une solution admissible pour les PPLs (7.6) et (7.6)_Z. Par restriction de l'espace des solutions, valeur(7.6) est majorée par valeur(7.6)_Z. Inversement, soit \mathbf{y} une solution optimale pour (7.6). On définit V comme l'ensemble des sommets atteignables depuis s en utilisant des arcs tels que $y_a = 0$. Comme dans la preuve précédente, V induit une coupe et on pose $\mathbf{y}^{\mathbb{Z}} = (y_a^{\mathbb{Z}})_{a \in A}$ avec

$$y_a^{\mathbb{Z}} = \begin{cases} 1 & \text{si } o(a) \in V \text{ et } t(a) \notin V \\ 0 & \text{sinon} \end{cases}$$

 $\mathbf{y}^{\mathbb{Z}}$ est clairement une solution admissible pour $(7.6)_{\mathbb{Z}}$. Soit $\alpha = \min\{y_a \mid o(a) \in V, t(a) \notin V\}$. Si $\alpha = 1$ alors $v(\mathbf{y}) \geq v(\mathbf{y}^{\mathbb{Z}})$ et $\mathbf{y}^{\mathbb{Z}}$ est optimale. Sinon, on pose $\mathbf{y}' = (y'_a)_{a \in A}$ avec $y'_a = \frac{y_a - \alpha y_a^{\mathbb{Z}}}{1 - \alpha}$. Clairement $\mathbf{y}' \geq \mathbf{0}$. Tout chemin joignant s à p contient au moins un arc de la coupe (entre V et son complémentaire). On écrit $\pi = \pi_{su} \cdot \pi_{up}$ où u est le dernier sommet dans V le long de π . On pose $\pi' = \pi'_{su} \cdot \pi_{up}$ où π'_{su} est un chemin joignant s à u par des arcs tels que $y_a = 0$. Puisque $y_a = 0$ implique $y'_a = 0$, on a

$$\sum_{a\in\pi}y'_a\geq \sum_{a\in\pi'}y'_a$$

Or π' contient un unique arc entre V et son complémentaire, d'où $\sum_{a \in \pi'} y_a^{\mathbb{Z}} = 1$. Par ailleurs $\sum_{a \in \pi'} y_a \ge 1$ et on en déduit aisément $\sum_{a \in \pi'} y_a' \ge 1$, ce qui montre que \mathbf{y}' est une solution admissible pour (7.6).

Donc $\mathbf{y} = \alpha \mathbf{y}^{\mathbb{Z}} + (1 - \alpha) \mathbf{y}'$ est une combinaison convexe de deux solution admissibles. Ces deux solutions sont donc optimales, d'où $v(\mathbf{y}^{\mathbb{Z}}) = v(\mathbf{y})$. Il suit que valeur $(7.6)_{\mathbb{Z}}$ est majorée par valeur(7.6), d'où l'égalité entre ces deux grandeurs.

7.2 Algorithmes du simplexe

On considére ici un PPL sous forme standard

$$\begin{array}{rcl} \min \mathbf{c} \mathbf{x} & \\ A \mathbf{x} &= \mathbf{b} & (E) \\ \mathbf{x} &\geq \mathbf{0}. \end{array}$$

dont la matrice A est de dimension $m \times n$. On supposera que A est de rang m (il y a donc plus d'inconnues que d'équations) c'est-à-dire qu'aucune des m équations n'est impliquée par les autres. Il est facile de supprimer de telles équations en temps polynomial ou de vérifier que le système n'a pas de solution. On appelle *solution ou point admissible* de (E) tout \mathbf{x} vérifiant les contraintes $A\mathbf{x} = \mathbf{b}$ et $\mathbf{x} \ge \mathbf{0}$. L'ensemble des solutions de (E) est un polyèdre noté P.

Soit $B \subset [1, n]$ un sous-ensemble de *m* indices tels que les vecteurs colonnes de *A* correspondants soient indépendants, i.e. tels que la matrice A_B des colonnes de *A* indexées par *B* soit inversible. On note $N = [1, n] \setminus B$ les indices complémentaires et on décompose chaque vecteur de \mathbb{R}^n en deux morceaux selon les indices de ses composantes $\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix}$. L'ensemble *B* est appelée une base de (E) et le vecteur $\begin{bmatrix} A_B^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix}$, la solution basique associée. On parle de solution basique admissible (ou réalisable) (s.b.a.) lorsque $A_B^{-1}\mathbf{b} \ge \mathbf{0}$.

La base associée est dite admissible.

Lemme 7.6 Pour toute base admissible B, il existe un vecteur de c' tel que la s.b.a. associée à B soit l'unique point qui minimise la forme linéaire $\mathbf{x} \mapsto \mathbf{c'x}$ sur (E).

Preuve : Considérer le vecteur $\mathbf{c}' = [\mathbf{c}'_B \mathbf{c}'_N] = [\mathbf{0} \ \mathbb{1}].$

Théorème 7.7 Toute solution basique admissible est un sommet de P. Réciproquement, tout sommet de P est la solution basique associée à une base admissible de (E).

Preuve : Si B est une base admissible alors le lemme 7.6 fournit un hyperplan support de P dont l'intersection avec P est la s.b.a. associée à B. Cette intersection est donc un sommet de P.

Réciproquement, soit \mathbf{y} un sommet de P. Je note B' l'ensemble des indices des coordonnées strictement positives de \mathbf{y} . En particulier, $A_{B'}\mathbf{y}_{B'} = \mathbf{b}$. Les colonnes de $A_{B'}$ forment une famille libre. En effet, dans le cas contraire il existe un jeux de coefficients \mathbf{d} non tous nuls tel que $A_{B'}\mathbf{d} = \mathbf{0}$. En choisissant un $\epsilon > 0$ assez petit, on a alors $\mathbf{y}_{B'} \pm \epsilon \mathbf{d} \ge \mathbf{0}$ et $A_{B'}(\mathbf{y}_{B'} \pm \epsilon \mathbf{d}) = \mathbf{b}$. On en déduit que \mathbf{y} est le milieu de deux points distincts dans P, en contradiction avec le lemme 6.26 sur l'extrémalité des sommets d'un polyèdre. On peut maintenant compléter B' en une base B de A de sorte que \mathbf{y} est la s.b.a. associée à B. \Box

Si *B* est une base, la projection $\begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix}$ \mapsto \mathbf{x}_N réalise une bijection, d'inverse $\mathbf{x}_N \mapsto \begin{bmatrix} A_B^{-1}(\mathbf{b} - A_N \mathbf{x}_N) \\ \mathbf{x}_N \end{bmatrix}$, entre *P* et le polyèdre P_N de \mathbb{R}^N : { $\begin{bmatrix} -A_B^{-1}A_N \\ I_N \end{bmatrix}$ $\mathbf{x}_N \ge \begin{bmatrix} -A_B^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix}$ }. Notons que *P* et P_N ont la même structure combinatoire. De plus, minimiser $\mathbf{cx} = \mathbf{c}_B \mathbf{x}_B + \mathbf{c}_N \mathbf{x}_N$ sur *P* est équivalent à minimiser $\mathbf{c'}_N \mathbf{x}_N$ sur P_N avec $\mathbf{c'}_N = \mathbf{c}_N - \mathbf{c}_B A_B^{-1} A_N$. Remarquons également que *B* est admissible si et seulement si $\mathbf{0} \in P_N$.

Définition 7.8 Un pivot consiste à remplacer un indice d'une base admissible par un indice non-base, de manière à ce que le nouvel ensemble d'indices soit une base admissible. Deux bases qui se déduisent d'un pivot sont dites adjacentes. Un pivot est dit diminuant si la valeur de la forme linéaire à minimiser diminue lorsqu'on l'évalue en la s.b.a. associée à la base avant et après le pivot.

Soit B et B' deux bases admissibles adjacentes avec $B' = B - s \cup \{e\}$ où $e \in N$ et $s \in B$ sont les indices *entrant* et *sortant* à échanger lors du pivot correspondant. Le système (E) équivalent au système

$$\mathbf{x}_B = A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N \qquad (E')$$
$$\mathbf{x} \ge \mathbf{0}.$$

On pose $\mathbf{p} = A_B^{-1}\mathbf{b}$ et $Q_N = -A_B^{-1}A_N$ de sorte que \mathbf{p} est la s.b.a. associée à B. Chaque équation dans (E') s'écrit ainsi

$$x_i = p_i + q_{ie}x_e + Q_{i,N-e}\mathbf{x}_{N-e} \tag{7.7}$$

avec $i \in B$. En séparant l'équation pour x_s , les équations de (E') peuvent encore s'écrire

$$\mathbf{x}_{B-s} = \mathbf{p}_{B-s} + Q_{B-s,e} x_e + Q_{B-s,N-e} \mathbf{x}_{N-e}$$
$$x_s = p_s + q_{se} x_e + Q_{s,N-e} \mathbf{x}_{N-e}$$

Soit encore sous forme matricielle

$$\begin{bmatrix} I & -Q_{B-s,e} \\ \mathbf{0} & -q_{se} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{B-s} \\ x_e \end{bmatrix} + \begin{bmatrix} -Q_{B-s,N-e} & 0 \\ -Q_{s,N-e} & -1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{N-e} \\ x_s \end{bmatrix} = \mathbf{p}$$

Comme les équations de (E) s'obtiennent en multipliant l'équation ci-dessus par la matrice inversible A_B (après réorganisation de ses colonnes), il suit que B' est une base si et seulement si la matrice $\begin{bmatrix} I & -Q_{B-s,e} \\ \mathbf{0} & -q_{se} \end{bmatrix}$ est inversible, c'est-à-dire $q_{se} \neq 0$. Par ailleurs, la s.b.a. \mathbf{p}' associée à B' est par définition telle que $\mathbf{p}'_{N'} = \mathbf{0}$, soit encore $\mathbf{p}'_{N-e} = \mathbf{0}$ et $p'_s = 0$. Ceci implique avec les équations (7.7) que $p'_e = -\frac{p_s}{q_{se}}$ et $p'_i = p_i - q_{ie}\frac{p_s}{q_{se}}$ pour $i \in B - e$. Les conditions $\mathbf{p}, \mathbf{p}' \ge \mathbf{0}$ d'admissibilité des bases B et B' se traduisent donc par les conditions

$$q_{se} < 0 \quad \text{et} \quad \frac{p_s}{q_{se}} = \min_{i \in B: q_{ie} < 0} \frac{p_i}{q_{ie}}$$
(7.8)

On remarque finalement que les s.b.a. \mathbf{p} , \mathbf{p}' sont distinctes si et seulement si p_s est non nul.

Théorème 7.9 Deux bases adjacentes correspondent à deux sommets identiques ou adjacents de P.

Preuve : Avec les notations de la discussion ci-dessus, on sait par les théorème 7.7 que **p** et **p'** sont des sommets de *P*. Par ailleurs, en notant π_N la projection de *P* sur le polyèdre P_N on a $\pi_N(\mathbf{p}) = \mathbf{0}$ et $\pi_N(\mathbf{p}') = \mathbf{p}'_N$ avec $\mathbf{p}'_{N-e} = \mathbf{0}$ et $p'_e = -p_s/q_{se}$. Si p_s est non nul, le segment $\pi_N(\mathbf{p})\pi_N(\mathbf{p}')$ n'est pas réduit à un point et est inclus dans l'arête $\{\mathbf{x}_{N-e} = \mathbf{0}, x_e \ge 0\}$ de l'hyperoctant $\mathbf{x}_N \ge \mathbf{0}$ de \mathbb{R}^N . Comme P_N est inclus dans cet hyperoctant, il suit que $\pi_N(\mathbf{p})\pi_N(\mathbf{p}')$ est une arête de P_N , ou encore que \mathbf{pp}' est une arête de P.

L'algorithme du simplexe est dû à George B. Dantzig [Dan51]. Il consiste à partir d'une base admissible à effectuer une suite de pivots diminuants jusqu'à atteindre l'optimum. Dit autrement, l'algorithme du simplexe consiste à se déplacer le long des arêtes de P en descendant toujours relativement à la forme linéaire associée à minimiser. Si l'on effectue un pivot à partir d'une base admissible B et que l'on écrit (E) sous la forme équivalente (E') ci-dessus avec $f(\mathbf{x}) = \mathbf{c}'_N \mathbf{x}_N$ comme forme à minimiser, l'indice e entrant lors du pivot doit être choisi tel que $c'_e > 0$. En effet, en notant \mathbf{p} et \mathbf{p}' les s.b.a. respectivement avant et après le pivot, on a $f(\mathbf{p}) = 0$ et $f(\mathbf{p}') = c'_e p'_e$ avec $p'_e > 0$ dès que \mathbf{p} et \mathbf{p}' sont distincts.

Cette dernière condition sur e forme avec les conditions (7.8) le principe fondamental de l'algorithme du simplexe. Si on ne peut trouver d'indices entrant et sortant avec ces

conditions alors ou bien tous les coefficients de $\mathbf{c'}_N$ sont non positifs et on a atteint le minimum et l'algorithme s'arête. Ou bien, pour un $c'_e > 0$ donné on ne peut trouver $q_{se} < 0$ (i.e. $q_{se} \ge 0$ pour tout $s \in B$) et le programme est non borné. En effet, on peut augmenter x_e à l'infini avec $\mathbf{x}_{N-e} = \mathbf{0}$ en restant dans (E) et faire ainsi tendre f vers l'infini.

Les conditions ci-dessus laissent tout de même un certain choix pour les indices entrant et sortant lors de chaque pivot. Des règles explicites pour effectuer ce choix ont été étudiées depuis les travaux précurseurs de Dantzig. Parmi celles-ci les plus connues sont le choix de l'indice e qui maximise c_e , le choix d'un pivot qui maximise la diminution de f, le choix d'un pivot qui minimise l'angle de l'arête \mathbf{pp}' avec \mathbf{c}' , ou encore la règle de Bland. Cette dernière consiste à choisir l'indice minimal parmi les indices entrants possibles et de même pour l'indice sortant. On peut montrer que la règle de Bland assure que l'algorithme du simplexe termine en un nombre fini d'étapes. Il se peut en effet que l'application d'une règle donne $\mathbf{p}' = \mathbf{p}$ pour plusieurs pivots de suite et que l'on retombe sur une même base induisant une boucle infinie dans l'algorithme.

Détermination d'une base admissible Pour initialiser l'algorithme du simplexe, il faut connaître une base admissible, c'est-à-dire un sommet de l'espace des solutions ou montrer que cet espace est vide. Pour cela on résoud le PPL suivant :

$$\min \mathbf{I} \mathbf{y} A \mathbf{x} + \mathbf{y} = \mathbf{b}$$
(E0)
$$\mathbf{x}, \mathbf{y} \geq \mathbf{0}.$$

En ayant pris soin de multiplier chaque ligne de $A\mathbf{x} = \mathbf{b}$ par ± 1 pour obtenir $\mathbf{b} \ge \mathbf{0}$. Le point $\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix}$ est triviallement solution de (E0) et (E) est non vide si et seulement min $\mathbf{I}\mathbf{y} = 0$. Il reste à appliquer l'algorithme du simplexe à (E0) pour vérifier cette denière condition et en déduire une base admissible de (E) le cas échéant.

On ne connaît pas à l'heure actuelle de règle permettant de rendre l'algorithme du simplexe polynomial. Un contre-exemple célèbre, le cube de Klee-Minsky montre que la plupart des règles, dont celle de Bland, conduise à un algorithme de complexité exponentielle dans le cas le pire. Récemment un algorithme du simplexe randomisé [KS06] de complexité moyenne polynomiale (mais pas fortement polynomial) a été proposé. Pour en savoir plus sur la programmation linéaire en général, on pourra consulter l'excellente introduction au domaine de J. Matoušek et B. Gärtner [MG07]. La vidéo du cours de Craig Tovey

www.youtube.com/watch?v=Ci1vBGn9yRc explique de maniere imagée la raison des termes simplexes et pivot dans la méthode de Dantzig.

7.3 La conjecture de Hirsch

Le *diamètre* d'un polytope est le diamètre de son 1-squelette, c'est-à-dire encore la distance maximale qui sépare deux sommets quelconques dans ce 1-squelette. Puisque la méthode du simplexe consiste essentiellement à se déplacer le long des arêtes du polytope des contraintes, une borne sur le diamètre des d-polytopes à n facettes fournie une borne sur le nombre d'étapes nécessaires pour résoudre un PPL à d inconnues et n inéquations par la méthode du simplexe. En 1957 Warren Hirsch a proposé le résultat suivant.

Conjecture de Hirsch Le diamètre d'un d-polytope à n facettes est majoré par n - d.

En référence à Hirsch, on note H(n,d) le diamètre maximal de tout d-polytope à n facettes.

La conjecture de Hirsch, longtemps considérée vraie, a été récemment infirmée par Francisco Santos Leal [San12] (cf. http://personales.unican.es/santosf/Hirsch/) à l'aide d'un contre-exemple. Il s'agit d'un 43-polytope à 86 facettes dont le diamètre est au moins 44 > 86 - 43. En pratique F. Santos travaille avec le polytope dual et obtient donc un 43-polytope à 86 sommets dont le diamètre du dual est au moins 44. Sa construction repose sur la notion de prismatoide.

Définition 7.10 Un prismatoide est un polytope dont les sommets sont contenus dans deux facettes ayant des hyperplans supports parallèles. Ces deux facettes sont dites de base.



FIGURE 7.1 – Un quadrilatère P (à gauche) et sa suspension par le point s (à droite).

Santos introduit ensuite la supsension par un point d'un polytope. Si s est un sommet d'un d-polytope P, sa suspension par le point s, notée $S_s(P)$ est le (d + 1)-polytope obtenu en plaçant P dans l'hyperplan $x_{d+1} = 0$ de \mathbb{R}^{d+1} et en ajoutant les deux sommets r := (s, -1) et $t := (s, 1) \in \mathbb{R}^{d+1}$, c'est-à-dire en posant $S_s(P) = Conv(P \cup \{r, s\})$. Voir la figure 7.3. Les facettes de $S_s(P)$ sont de la forme $S_s(F)$ pour toute facette F de Pcontenant s ou de la forme F * r et F * t pour toute autre face de P.

On note $w(P^*)$ le diamètre du dual de P.

Lemme 7.11 Soit P un d-prismatoide à n sommets. Il existe un (n-d)-prismatoide Q à 2(n-d) sommets tel que $w(Q^*) \ge w(P^*) + n - 2d$.

Preuve : On raisonne par récurrence sur k := n - 2d. Notons que P ayant ces deux facettes de base P^+ et P^- disjointes, on a $n \ge 2d$ et donc $k \ge 0$. Pour k = 0, le lemme

est une tautologie. On suppose donc k > 0. Il suit que l'une des deux facettes de base, disons P^+ , a au moins d + 1 sommets. Soit s un sommet de P^- . Le polytope $S_s(P)$ a ses sommets contenu dans les faces $S_s(P^-)$ et P^+ . En perturbant un sommet de P^+ dans la direction de l'hyperplan support de $S_s(P^-)$, on obtient un d-polytope P_1^+ contenu dans un hyperplan parallèle à $S_s(P^-)$. Le polytope $P_1 := Conv(S_s(P^-) \cup P_1^+)$ est donc un (d + 1)-prismatoide à n + 1 sommets. On vérifie que s'il faut traverser $w(P^*)$ facettes pour passer de P^+ à P^- dans P, il faut en traverser au moins $w(P^*) + 1$ pour passer de $S_s(P^-)$ à P_1^+ dans P_1 (cf. [San12] pour les détails). Par récurrence, on obtient ainsi des polytopes $P_1, P_2, \ldots, P_{n-2d}$ avec $w(P_i^*) \ge w(P^*) + i$. On remarque finalement que $Q := P_{n-2d}$ convient.

Santos décrit la liste des 48 sommets d'un certain 5-prismatoide P. Les nombreuses symétries de P et sa taille relativement petite (322 facettes) permettent de vérifier à la main que $w(P^*) = 6$. Le lemme précédent implique alors l'existence d'un 43-polytope Qà 86 sommets tel que $w(Q^*) \ge 44 > 86 - 43$, ce qui achève la construction du contreexemple. Des contre-exemples plus petits ont été exhibés depuis (voir la page ouaib de F. Santos à ce sujet).

7.4 Programmation linéaire en temps linéaire

Dans les années 1980 N. Megiddo [Meg84] a proposé un algorithme probabiliste de complexité linéaire pour résoudre un PPL en dimension d fixée. Cet algorithme a ensuite grandement été simplifié par Seidel [Sei91]. Nous présentons la méthode de Seidel.

Soit \mathcal{H} une famille finie de demi-espaces de \mathbb{R}^d , et soit h une forme linéaire, possiblement nulle. On fixe également un point O de \mathbb{R}^d . Rappelons que par le théorème de Minkowski-Weil le polyèdre $P = \cap \mathcal{H}$ est la somme de Minkowski de son cône de récession \vec{P} et d'un polytope. Si P est non vide, il y a deux cas possible.

- 1. Ou bien h atteint une valeur maximale finie h_{max} sur P et donc sur une face $F = P \cap \{h = h_{\text{max}}\}$. On a par exemple F = P si h = 0.
- 2. Ou bien h est non majorée sur P, ce qui équivaut à l'existence d'une direction \mathbf{v} de \vec{P} telle que $h(\mathbf{v}) > 0$. De manière encore équivalente, le vecteur de \vec{P} à distance minimale du vecteur dual de h, vu comme point de \mathbb{R}^d , est non nul.

Dans le premier cas on définit la *solution* du problème de programmation linéaire donné par (\mathcal{H}, h, O) comme le point de F à distance (euclidienne) minimale de O. Dans le second cas, on définit la solution comme le vecteur de \vec{P} à distance minimale du vecteur dual de h. Il est facile de voir que la solution est définie de manière unique dans les deux cas (cf. exercice ci-après).

Si $P = \cap \mathcal{H}$ est non vide et si la solution du problème associé à (\mathcal{H}, h, O) est un point p, on peut ainsi conclure que h est maximale en p sur P. Si la solution est un vecteur \mathbf{v} on conclut que h est non majorée sur P dans la direction \mathbf{v} , i.e. que $h(\mathbf{v}) > 0$.

Exercice 7.12 On considère dans \mathbb{R}^d un point O et un polyèdre non vide P. Montrer qu'il existe un unique point de P qui minimise la distance euclidienne à O.

Lemme 7.13 Soit \mathcal{H} une famille finie de demi-espaces vectoriels dans \mathbb{R}^d . Soit E un hyperplan vectoriel valide pour le cône $C = \cap \mathcal{H}$ et soit $F = E \bigcap \cap \mathcal{H}$, la face de C correspondante. Il existe une sous famille $\mathcal{H}' \subset \mathcal{H}$ d'au plus $d - \dim F$ demi-espaces telle que E est valide pour le cône $\cap \mathcal{H}'$.

Preuve : Soit **v** la normale à *E* opposée à *C*. Pour tout demi-espace $H \in \mathcal{H}$, on note $H = \{\mathbf{x} \mid \mathbf{v}_{\mathbf{H}}\mathbf{x} \leq 0\}$ son équation et $H^0 := \{\mathbf{x} \mid \mathbf{v}_{\mathbf{H}}\mathbf{x} = 0\}$ son hyperplan bordant.

On suppose dans un premier temps que C est de dimension d. Par la proposition 6.58 de dualité des cônes, il existe une sous-famille de $\mathcal{K} \subset \mathcal{H}$ telle que

$$\mathbf{v} = \sum_{H \in \mathcal{K}} \lambda_H \mathbf{v}_\mathbf{H} \tag{7.9}$$

où les λ_H sont strictement positifs et où $\{\mathbf{v}_H\}_{H\in\mathcal{K}}$ est de rang au plus $d - \dim F$. Si cette famille contient plus d'éléments que son rang, alors elle est liée et on peut écrire $\sum_{H\in\mathcal{K}} \mu_H \mathbf{v}_H = \mathbf{0}$ pour des coefficients μ_H non tous nuls. On peut choisir un réel α tel que les coefficients $\lambda_H + \alpha \mu_H$ sont positifs et au moins l'un est nul. Ceci permet de réduire \mathcal{K} d'un élément au moins. Par récurrence on peut supposer que \mathcal{K} contient au plus $d-\dim F$. L'équation (7.9) montre que l'on peut prendre $\mathcal{H}' = \mathcal{K}$.

Si C est de dimension k < d, alors par le lemme 6.46, on peut extraire une sous-famille $\mathcal{H}_{\mathcal{C}} \subset \mathcal{H}$ de taille d - k telle que l'espace engendré par C soit $vec(C) = \bigcap_{H \in \mathcal{H}_{\mathcal{C}}} H^0$. En appliquant ce qui précède aux traces de E et \mathcal{H} dans vec(C), on en déduit une sousfamille $\mathcal{K} \subset \mathcal{H}$ de taille au plus $k - \dim F$ telle que $E \cap vec(C)$ est valide pour le cône $\cap \mathcal{K} \cap vec(C)$. On peut alors prendre $\mathcal{H}' = \mathcal{H}_{\mathcal{C}} \cup \mathcal{K}$. \Box

Lemme 7.14 On considère dans \mathbb{R}^d un point O et une famille finie \mathcal{H} de demi-espaces vectoriels. On peut extraire une sous-famille $\mathcal{H}' \subset \mathcal{H}$ d'au plus d demi-espaces telle que la distance de O à $\cap \mathcal{H}$ et la distance de O à $\cap \mathcal{H}'$ soient minimisées par le même point.

Preuve : On pose $C = \cap \mathcal{H}$. Soit p le point de C qui minimise la distance euclidienne à O. Si p = O on peut prendre $\mathcal{H}' = \emptyset$ (on convient que $\cap \mathcal{H} = \mathbb{R}^d$ si $\mathcal{H} = \emptyset$). Supposons donc $p \neq O$. Par convexité de C, il suit aisément que l'hyperplan E orthogonal à Op passant par p est valide pour C. Par le lemme précédent, on peut choisir une sous-famille $\mathcal{H}' \subset \mathcal{H}$ d'au plus d demi-espaces telle que E est valide pour le cône $C' = \cap \mathcal{H}'$. Il suit que $p \in C \subset C'$ minimise la distance de O à C'.

Théorème 7.15 Soit \mathcal{H} une famille de n demi-espaces de \mathbb{R}^d , et soit $h : \mathbf{x} \mapsto \mathbf{cx}$ une forme linéaire sur \mathbb{R}^d . On fixe également une origine O dans \mathbb{R}^d . Il existe un algorithme randomisé de complexité moyenne $O(2^d(d!)n)$ qui détermine si $\cap \mathcal{H}$ est vide, et dans la négative calcule une solution du problème de programmation linéaire défini par (\mathcal{H}, h, O) .

Preuve : On suppose, par une double récurrence sur d et n qu'il existe un algorithme randomisé répondant aux exigences du théorème. On note T(d, n) sa complexité moyenne. En dimension d = 1, l'intersection $\cap \mathcal{H}$ est un segment possiblement vide ou (semi)infini
qui se détermine aisément en temps O(n). On en déduit ensuite tout aussi aisément la solution du problème en temps constant. On a ainsi T(1, n) = O(n).

Supposons donc d > 1. Si n = 1, alors \mathcal{H} contient un unique demi-espace de la forme $H = \{\mathbf{x} \mid \mathbf{c_1} \mathbf{x} \leq b_1\}$. La solution s'obtient comme suit.

- Si $h(\mathbf{c_1}) < 0$, alors h est non bornée sur H et la solution est le vecteur **c** dual de h.
- Sinon, soit $\mathbf{c}_{\mathbf{H}}$ la projection orthogonale de \mathbf{c} sur l'hyperplan vectoriel bordant H.
 - Si $\mathbf{c}_{\mathbf{H}} \neq \mathbf{0}$, alors *h* est non bornée sur *H* et la solution est le vecteur $\mathbf{c}_{\mathbf{H}}$.
 - Sinon h est bornée sur H. Si $O \in H$, la solution est le point O, sinon c'est la projection de O sur l'hyperplan affine bordant H.

Ce calcul peut triviallement s'effectuer en temps O(d), d'où T(d, 1) = O(d).

Si n > 1, on suppose que les demi-espaces H_1, H_2, \ldots, H_n de \mathcal{H} sont donnés dans un ordre aléatoire pour la loi uniforme. On note respectivement H_i^0, \vec{H}_i , et \vec{H}_i^0 l'hyperplan bordant H_i , le demi-espace vectoriel translaté de H_i , et son hyperplan vectoriel bordant. Par hypothèse de récurrence, on peut calculer la solution du problème défini par $(\mathcal{H} \setminus \{H_n\}, h, O)$ en temps moyen T(d, n - 1). Plusieurs cas se présentent.

- 1. Si $\cap_{i < n} H_i$ est vide, alors c'est le cas pour $\cap \mathcal{H}$.
- 2. Si la solution de $(\mathcal{H} \setminus \{H_n\}, h, O)$ est un vecteur (donc $(\mathcal{H} \setminus \{H_n\}, h, O)$ est non borné) contenu dans \vec{H}_n , alors c'est aussi la solution du problème défini par (\mathcal{H}, h, O) (qui est donc également non borné).
- 3. Si la solution de $(\mathcal{H} \setminus \{H_n\}, h, O)$ est un point contenu dans H_n , alors c'est aussi la solution du problème défini par (\mathcal{H}, h, O) .
- 4. Sinon, la solution de (\mathcal{H}, h, O) est la solution du problème (\mathcal{H}', h', O') où
 - $-h'(\mathbf{x}) = \mathbf{c}'\mathbf{x}$ et \mathbf{c}' est la projection orthogonale de \mathbf{c} sur \vec{H}_n^0 ,

$$- \mathcal{H}' = \{H_j \cap H_n^0\}_{1 \le j < n},$$

– O' est la projection orthogonale de O sur H_n^0 .

Les tests des cas 2 et 3 s'effectuent aisément en temps O(d).

Dans le cas 4, $P' = \cap \mathcal{H}'$ est un polyèdre de dimension d-1 défini par l'intersection de n-1 demi-espaces dans l'hyperplan H_n^0 . Le calcul d'une base orthonormée² de H_n^0 peut s'effectuer en temps $O(d^3)$, puis l'équation dans cette base de chaque demi-espace $H_j \cap H_n^0$ peut s'obtenir en temps $O(d^2)$. Ce qui permet de décrire le problème (\mathcal{H}', h', O') en temps $O(d^3n)$. Par hypothèse de récurrence, on peut ensuite déterminer en temps moyen T(d-1, n-1) la solution de (\mathcal{H}', h', O') et donc de (\mathcal{H}, h, O) .

Analysons la probabilité de se retrouver dans le cas 4. On suppose P non vide. Ou bien h est bornée sur P de valeur maximale h_{max} . On note p la solution de (\mathcal{H}, h, O) , et on pose $E = \{h = h_{max}\}$. Soit O_H la projection orthogonale de O sur E.

• Si $h \neq 0$, par le lemme 7.13, on peut extraire une sous-famille $\mathcal{K} \subset \mathcal{H}$ de taille au plus d telle que l'hyperplan E est valide pour $\cap \mathcal{K}$. Puisque $p \in E$, p est aussi le point de la trace de P sur E le plus proche de O_H . Par le lemme 7.14 appliqué dans E, on peut extraire une sous-famille $\mathcal{K}' \subset \mathcal{H}$ de taille au plus d-1 telle que le point le plus proche de O_H dans $E \bigcap \cap \mathcal{K}'$ est p. On en déduit que si $H_n \notin \mathcal{K} \cup \mathcal{K}'$, alors h était borné sur $Q := \bigcap_{1 \leq j \leq n-1} H_i$ et la solution du problème associé était déjà p, ce qui correspond

^{2.} Une base orthonormée permet de calculer la norme euclidienne d'un vecteur en temps O(d) à partir de ses coordonnées dans la base.

au cas 2. Dit autrement, la probabilité de se retrouver dans le cas 4 est majorée par la probabilité que H_n soit dans $\mathcal{K} \cup \mathcal{K}'$. Cette probabilité est donc majorée par 2d/n.

Si h = 0, alors E = ℝ^d, O_H = O et p est par définition le point de P le plus proche de O. Par le lemme 7.14, on peut extraire une sous-famille de taille au plus d dans H telle que le point de le plus proche de l'intersection de cette sous-famille est p. Un raisonnement analogue à ce qui précède montre que la probabilité de se retrouver dans le cas 4 est majorée par d/n.

Ou bien h est non bornée sur P et un raisonnement analogue montre que la probabilité de se retrouver dans le cas 4 est aussi majorée par d/n.

On en conclut que T(d, n) vérifie la récurrence :

$$T(d,n) \leq \begin{cases} O(n) & \text{si } d = 1, \\ O(d) & \text{si } n = 1, \\ T(d,n-1) + O(d) + \frac{2d}{n}(O(d^3n) + T(d-1,n-1)) & \text{sinon.} \end{cases}$$

La dernière inéquation s'écrit encore.

$$T(d,n) \le T(d,n-1) + O(d^4) + \frac{2d}{n}(T(d-1,n-1))$$

Un calcul simple montre que $T(d, n) = O(2^d(d!)n)$. En effet, supposons que $T(d, m) \leq a(d)2^d(d!)m$ pour une certaine fonction a(d) et tout m < n. En reportant dans l'équation de récurrence ci-dessus, il apparaît que l'inégalité s'étend à m = n si on choisit $a(d) = O(\sum_{1 \leq k \leq d} \frac{d^4}{2^d(d!)})$. Cette série étant convergente, on a a(d) = O(1). \Box

Chapitre 8

Enveloppes convexes, Voronoi et Delaunay

8.1 Calculs d'enveloppes convexes

De manière générale, on veut résoudre

Problème 8.1 Soit un ensemble $S = \{s_1, s_2, \ldots, s_n\}$ de n points de \mathbb{R}^d , calculer le treillis des faces de Conv(S).

Le tri de *n* nombres $\{x_1, x_2, \ldots, x_n\}$ se réduit en temps linéaire au calcul de l'enveloppe convexe des *n* points du plan $\{(x_1, x_1^2), (x_2, x_2^2), \ldots, (x_n, x_n^2)\}$ situés sur la parabole $\{y = x^2\}$. On en déduit

Lemme 8.2 La complexité du calcul d'enveloppes convexes en dimension supérieure ou égale à 2 admet pour borne inférieure $\Omega(n \log n)$.

En utilisant le modèle de l'arbre binaire de décision algébrique pour l'analyse des algorithmes (cf. Preparata et Shamos, 1984, pp 101-104), on montre également que la seule sélection des points extrêmes de Conv(S) (sans leurs adjacences) admet pour borne inférieure $\Omega(n \log n)$.

8.1.1 Algorithmes naïfs

Par sélection des points extrêmes

Pour vérifier qu'un point $s \in S$ est extrême il suffit de vérifier qu'il n'est contenu dans aucun des *d*-simplexes formés sur $S \setminus s$. Il y a $\binom{n-1}{d+1}$ tels simplexes, ce qui induit un algorithme en temps $O(n\binom{n-1}{d+1}) = O(n^{d+2})$.

Par sélection des facettes

Un sous-ensemble de d points de S engendre une facette si son enveloppe affine ne sépare pas les points de S (i.e. est support pour S). On en déduit un algorithme en temps $O(\binom{n}{d}n) = O(n^{d+1})$.

Par propagation à partir d'une facette

Définition 8.3 L'angle dièdre de deux demi-espaces est l'angle compris entre 0 et π formé par les normales aux hyperplans supports tournées vers l'extérieur.

Soit F une facette de P = Conv(S) et G une facette de F, i.e. une (d-2)-face de P. Pour tout $s \in S \setminus F$ on note H_s le demi-espace bordé par G et s et contenant F. Le demi-espace support de la facette adjacente à F le long de G coïncide avec l'hyperplan H_s lorsque celui-ci forme un angle dièdre minimal avec le demi-espace support de F.

Connaissant F et G, on peut donc déterminer la facette adjacente en temps linéaire. En supposant P simplicial (i.e. S en position générale) toute facette de F est déterminée par un sous-ensemble de d-1 des d sommets de F. Cela fournit un algorithme en O(nh) - où h est le nombre de facettes de P – pour déterminer toutes les facettes et leurs adjacences. D'après le théorème de la borne supérieure cela fournit un algorithme en $O(n^{\lfloor d/2 \rfloor + 1})$.

Il reste à trouver une première facette : prendre le point $s_0 \in S$ de coordonnées minimales pour l'ordre lexicographique, puis l'arête s_0s_1 formant un angle minimal avec sa projection sur l'hyperplan $H_0 = \{x_1 = (s_0)_1\}$, puis le triangle $s_0s_1s_2$ formant un angle minimal avec sa projection sur l'hyperplan contenant s_0s_1 dans le faisceau définit par H_0 et l'hyperplan normal à s_0s_1 . De manière générale si on a déterminé une kface $F_k = Conv(\{s_0, s_1, \ldots, s_k\})$ et un hyperplan support H_k de F_k on obtient $F_{k+1} =$ $Conv(F_k \cup \{s_{k+1}\})$ en cherchant s_{k+1} tel que $Conv(F_k \cup \{s_{k+1}\})$ forme un angle minimal avec sa projection sur H_k . Et on obtient H_{k+1} - contenant s_{k+1} - dans le faisceau définit par H_k et l'hyperplan normal à s_ks_{k+1} contenant s_k . Cette étape prend un temps O(dn)et on obtient finalement un algorithme en $O(n^{\lfloor d/2 \rfloor + 1})$ pour la construction de toutes les facettes. Notons que le treillis des faces de P étant coatomique, la connaissance des facettes de P détermine entièrement ce treillis.

On peut remarquer que cet algorithme ne nécessite que des calculs de déterminants pour ordonner selon des angles.

8.1.2 Calcul en 2D

Marche de Jarvis

En deux dimensions l'algorithme précédent s'appelle la marche de Jarvis et fournit un algorithme de complexité $O(n^2)$.

Balayage de Graham

Soit o un point intérieur à P = Conv(S) et soit s_0 un point extrême de S, par exemple le point de coordonnées minimales pour l'ordre lexicographique. On considère l'ordre cyclique sur les sommets s_i de S défini par l'angle polaire des demi-droites os_i avec os_0 (en fait l'ordre lexicographique sur la paire (angle, - distance à o)). P est un sous-ordre de cet ordre cyclique obtenu en ôtant récursivement tous les points formant un angle réflexe avec leur successeur et leur prédécesseur. D'un point de vu algorithmique, il suffit de parcourir la liste ordonnée des points à partir de s_o et de supprimer le point courant tant qu'il est le sommet d'un angle réflexe. Chaque sommet n'étant supprimer qu'un fois au plus, P peut être calculé en temps linéaire à partir de cette liste, ce qui fournit un algorithme en temps $O(n \log n)$.

Diviser pour régner

1. On partitionne les points de S en deux selon la médiane de leurs abscisses en temps O(n).

2. On calcule récursivement les enveloppes convexes ECG et ECD de chaque moitié, et on fait leur fusion en temps O(n).

Le temps total T(n) requis pour le calcul de l'enveloppe convexe vérifie ainsi.

$$T(n) = O(n) + T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil)$$

D'où $T(n) = O(n \log n)$, ce qui est optimal.

La fusion de ECG et ECD s'obtient en recherchant leurs tangentes, ou *ponts*, supérieure et inférieure. Pour cela on part d'une arête joignant le sommet de ECG (ECD) le plus à droite (gauche) qui est visible par ECD (ECG) puis on marche le long de ECD (ECG) vers le haut et vers le bas pour chacun des deux ponts haut et bas jusqu'à ce que les angles aux extrémités de chaque pont avec les chaînes correspondantes de ECG et ECD soient convexes.

balayage

Le calcul de l'enveloppe convexe de n points $S = S_n = \{s_1, s_2, \ldots, s_n\}$ du plan peut se faire à l'aide d'un algorithme par balayage. Cet algorithme repose sur trois étapes.

- 1. On commence par trier les n points s_i selon une direction donnée (par exemple l'axe des abscisses. Si plusieurs points ont même abscisse on utilise l'ordre lexicographique sur les couples de coordonnées). On supposera par la suite que les indices des points correspondent à l'ordre croissant.
- 2. On initialise $Conv(S_3)$ par le triangle $s_1s_2s_3$ dans l'ordre circulaire direct.
- 3. Itération : On construit $Conv(S_i)$ à partir de $Conv(S_{i-1})$ et du point s_i .

Clairement (pourquoi?) $s_i \notin Conv(S_{i-1})$ et $Conv(S_i)$ s'obtient en supprimant les arêtes vues par s_i et en ajoutant deux arêtes issues de s_i et joignant les deux sommets aux extrémités de la chaîne des arêtes supprimées. Le coût de parcours des arêtes supprimées peut être reporté (chargé) sur le coût de création de ces arêtes puisque toute arête créée n'est supprimée qu'une fois au plus. À chaque itération 2 arêtes sont créés donc le coût total des itérations est linéaire. En tenant compte du tri initial on obtient ainsi un algorithme en $O(n \log n)$.

Algorithme dynamique

- Dynamic Planar Convex Hull. G. S. Brodal and R. Jacob. FOCS'02. pp 617 - 626. Vancouver, Canada.

"There exists a data structure for the fully dynamic planar convex hull problem supporting INSERT and DELETE in amortized $O(\log n)$ time, and EXTREME POINT QUERY, TANGENT QUERY and NEIGHBORING-POINT QUERY in $O(\log n)$ time, where ndenotes the size of the stored set before the operation. The space usage is O(n)."

8.1.3 Calcul en 3D

La taille d'une enveloppe convexe de n points en 3D est linéaire. Cela résulte directement du théorème de la borne supérieure ou bien de la formule d'Euler : n - A + F = 2 et $3F < 2A \implies A < 3n - 6$ et F < 2n - 4.

Puisque le bord d'un polytope de dimension 3 est homéomorphe à une sphère on peut représenter le treillis des faces d'un tel polytope par une carte "planaire" (en fait sphérique). Dans ce qui suit on supposera que les enveloppes convexes sont représentées par une carte planaire. Une autre possibilité, qui a l'avantage de se généraliser en toute dimension, est de représenter une enveloppe convexe par le treillis de ses faces, c'est-à-dire par le graphe d'incidence de ses faces : chaque face est représentée par un élément qui pointe sur ses facettes et cofacettes.

balayage

Le calcul de l'enveloppe convexe de n points $S = S_n = \{s_1, s_2, \ldots, s_n\}$ de \mathbb{R}^3 peut se faire à l'aide d'un algorithme par balayage. Cet algorithme repose sur trois étapes.

- 1. On commence par trier les n points s_i selon l'ordre lexicographique sur leurs triplets de coordonnées. On supposera par la suite que les indices des points correspondent à l'ordre croissant.
- 2. On initialise $Conv(S_4)$ par le tétraèdre $s_1s_2s_3s_4$, ou plus précisément par les quatre premiers points non coplanaires, orienté positivement.
- 3. Itération : On construit $Conv(S_i)$ à partir de $Conv(S_{i-1})$ et du point s_i .

Clairement $s_i \notin Conv(S_{i-1})$ et $Conv(S_i)$ s'obtient en supprimant les faces et arêtes vues par s_i et en ajoutant les faces et arêtes du cône de sommet s_i et joignant le bord des faces de $Conv(S_{i-1})$ supprimées.

Le coût de parcours des faces et arêtes supprimées peut être reporté (chargé) sur le coût de création de ces faces et arêtes. À chaque itération O(n) faces et arêtes peuvent être

créées donc le coût total des itérations est quadratique. En tenant compte du tri initial on obtient un algorithme de complexité $O(n^2)$.

Exercice 8.4 Donnez un exemple de S pour lequel l'algorithme est un $\Omega(n^2)$.

Algorithme randomisé incrémental

Il s'agit d'une version randomisée de l'algorithme par balayage. On considère un tétraèdre inclus dans Conv(S), par exemple en prenant quatre points de S affinement indépendants. On ajoute les points de S incrémentalement dans un ordre aléatoire. On note S_i l'ensemble des i premiers points insérés (hormis les 4 points du tétraèdre initial). Soit s_i le i-ième point inséré.

- si $s_i \in Conv(S_{i-1})$, alors $Conv(S_i) = Conv(S_{i-1})$,

- sinon $Conv(S_i)$ s'obtient en supprimant les faces et arêtes vues par s_i et en ajoutant les faces et arêtes du cône de sommet s_i tangent à $Conv(S_{i-1})$.

Pour savoir si $s_i \in Conv(S_{i-1})$ ou pour connaître les faces de $Conv(S_{i-1})$ vues par s_i on maintient un graphe des conflits entre les faces de $Conv(S_{i-1})$ et les points de $S \setminus S_{i-1}$: chaque face f pointe vers la liste S(f) des points de $S \setminus S_{i-1}$ qui la voit (i.e. qui ne sont pas contenus dans le demi-espace support de f) et chaque point s de $S \setminus S_{i-1}$ pointe vers la liste F(s) des faces de $Conv(S_{i-1})$ qu'il voit. On a $s_i \in Conv(S_{i-1})$ si et seulement si $F(s_i)$ est vide. Dans le cas contraire la mise à jour du graphe des conflits s'obtient en supprimant s_i et toutes les faces (et leurs pointeurs) de $F(s_i)$ et en ajoutant toutes les faces du cône de s_i ainsi que leur liste de conflits. Pour déterminer la liste des conflits d'une nouvelle face f incidente à une arête d'horizon e (i.e. du bord du cône), on remarque que tout point voyant f voyait au moins l'une des deux faces f_1 et f_2 de $Conv(S_{i-1})$ incidentes à e. Il suffit donc de parcourir $S(f_1)$ et $S(f_2)$ pour en extraire S(f) : pour chaque sommet s de $S(f_1) \cup S(f_2)$ on test en temps constant si $s \in S(f)$, i.e. si s est dans le demi-espace bordé par le plan support de f et ne contenant pas $Conv(S_i)$.

Analyse

Lemme 8.5 L'espérance du nombre de faces créées au cours de l'algorithme est linéaire.

Preuve : Le nombre de faces créées par l'insertion de s_i est égal au degré d_i de s_i dans $Conv(S_i)$. Or, sur l'ensemble des permutations de S dont on a fixé l'ensemble S_i des i premiers sommets, chaque sommet $s \in S_i$ est le sommet s_i avec la probabilité 1/i, d'où

$$E(d_i|S_i) = \frac{1}{i} \sum_{s \in S_i} d(s) \le \frac{1}{i} (6(i+4) - 12) = O(1)$$

En effet, $\sum_{s \in S_i} d(s)$ vaut 2 fois le nombre d'arêtes de $Conv(S_i)$ et ce dernier nombre est majoré par 3(i + 4) - 6 (voir le début de cette section). On en déduit de manière inconditionnelle $E(d_i) = O(1)$, d'où le résultat par linéarité de l'espérance.

La mise à jour des listes de conflits F(.) et S(.) à l'instant *i* prend clairement un temps proportionnel à la somme

- du nombre de listes supprimées, i.e. du nombre de faces supprimées,
- du nombre de listes créées, i.e. du nombre de faces créées,
- du nombre de conflits supprimés dans les listes, et
- du nombre de conflits ajoutés dans les listes.

Puisque le nombre de listes ou conflits supprimés est respectivement inférieur au nombre de listes ou conflits ajoutés, il suffit de comptabiliser ces derniers, ce qui correspond au coût de construction des listes S(f) pour les nouvelles faces f ajoutées à l'instant i.

On construit initialement les listes de conflits des 4 faces du tétraèdre de départ, ce qui prend un temps O(n) par simple inspection de chaque sommet. Il reste finalement à évaluer le coût de construction des S(f) après initialisation. Pour cela on définit S(e) = $S(f_1) \cup S(f_2)$ la liste de conflits d'une arête *e* incidente à deux faces f_1 et f_2 .

D'après ce qui précède le paragraphe d'analyse, le coût de création des S(f) est proportionnel à la somme $\sum_{e} |S(e)|$ pour toutes les arêtes d'horizon apparaissant au cours de l'algorithme. On veut donc majorer l'espérance de cette somme.

Pour poursuivre l'analyse on définit la *région* associée à un quadruplet de points (p, q, r, s) comme l'union du demi-espace ouvert bordé par p, q, r et ne contenant pas s et du demi-espace ouvert bordé par p, q, s et ne contenant pas r. Les points de S en conflit avec une région R sont par définition les points contenus dans cette zone. On note S(R) l'ensemble de ces points. On supposera pour simplifier l'exposé que les points de S sont en position générale.

À chaque arête e = pq de $Conv(S_{i-1})$ on peut associer la région (p, q, r, s) où pqr et qpssont les deux triangles incidents à e dans $Conv(S_{i-1})$. Réciproquement les régions formées sur les points de S_{i-1} et sans conflit avec les points de S_{i-1} (les régions *actives* à l'instant i-1) sont précisément obtenues à partir des arêtes de $Conv(S_{i-1})$ comme précédemment indiqué.

Si e est une arête d'horizon de $Conv(S_{i-1})$ pour s_i , alors S(e) est l'ensemble des sommets en conflit avec la région associée à e. Pour majorer $\sum_e |S(e)|$, il suffit donc de majorer $\sum_R |S(R)|$, la somme portant sur l'ensemble de toutes les régions actives au cours de l'algorithme. Par le lemme 13.8 des algorithmes randomisés incrémentaux (voir section 13.3.1), l'espérance de cette somme est majorée par

$$\sum_{i=1}^{n} 4^2 \frac{n-i}{i^2} E(|\mathcal{A}_i|)$$

où \mathcal{A}_i est l'ensemble des régions actives à l'étape *i*. Dans le cas présent cet ensemble correspond aux arêtes de $Conv(S_i)$, d'où $E(|\mathcal{A}_i|) \leq 3(i+4) - 6$. On en déduit

$$E(\sum_{e} |S(e)|) = O(n \log n)$$

et par conséquent

Proposition 8.6 L'enveloppe convexe de n points dans \mathbb{R}^3 peut être calculée par un algorithme randomisé en temps moyen $O(n \log n)$.

8.1.4 Calcul en dimension quelconque

On se place dans le cadre dual du calcul du treillis des faces d'un polytope P donné par une intersection de demi-espaces dans \mathbb{R}^d . On s'appuie pour cela sur l'algorithme randomisé de programmation linéaire en temps linéaire due à Seidel [Sei91] et présenté dans la section 7.4.

On suppose que les hyperplans bordant les demi-espaces de P sont en position générale. En conséquence, P est simple et chacun de ses sommets a d arêtes incidentes.

Théorème 8.7 Soit un entier d > 3 fixé. Il existe un algorithme randomisé pour calculer le treillis du polytope intersection de n demi-espaces donnés en position générale en temps moyen $O(n^{\lfloor d/2 \rfloor})$.

Preuve : Soit H_1^0, \ldots, H_n^0 les *n* hyperplans bordant les demi-espaces H_1, \ldots, H_n donnés. Pour simplifier l'exposé on commence par ajouter les 2d demi-espaces supports $H_{1-2d}, H_{2-2d}, \ldots, H_0$ d'un grand cube contenant l'intersection des H_i . Ceci permet de supposer que l'intersection des hyperplans est bornée. Remarquons que l'intersection des H_i est un polytope simple par hypothèse de position générale. Chaque sommet est donc incident à *d* arêtes et chaque arête est incidente à d-1 faces de dimension 2 (une arête peut former une 2-face avec chacune des d-1 arêtes partageant l'une fixée de ses extrémités).

On considère que les H_i , i > 0, sont dans un ordre aléatoire pour la loi uniforme. On maintient incrémentalement le 2-squelette de $P_i := \bigcap_{-2d < j \leq i} H_j$. Par la remarque précédente, ce 2-squelette a une taille proportionnelle à son nombre de sommets (d est fixé!). Le 2-squelette du cube P_0 est directement calculé en temps constant. Supposons avoir calculé le 2-squelette de P_i pour un certain i < n. On détermine si H_{i+1} contient P_i ou non. Pour cela on détermine le sommet p_i de P_i qui maximise la forme linéaire correspondant à H_{i+1} . D'après le théorème 7.15, p_i peut être calculé en temps O(i). Si p_i est dans H_{i+1} , alors H_{i+1} contient P_i et $P_{i+1} = P_i$. Sinon, il faut mettre à jour le 2-squelette de P_{i+1} en coupant la partie de P_i dans le complémentaire H_{i+1}^+ de H_{i+1} . Pour cela on parcourt le 1-squelette de P_i à partir de p_i pour trouver la partie (connexe) contenue dans H_{i+1}^+ et les arêtes a_1, \ldots, a_k coupées par H_{i+1}^0 . Les nouveaux sommets de P_{i+1} sont les intersections de H_{i+1}^0 avec les a_j , et les nouvelles arêtes sont les intersections de H_{i+1}^0 avec les 2-faces incidentes aux a_j . Ceci permet de construire le 1-squelette de P_{i+1} . Le 2-squelette est obtenu en temps proportionnel au nombre de sommets par la propriété de coatomicité des polytopes : en identifiant chaque facette avec son hyperplan support on peut associer à chaque sommet (resp. arête) les d (resp. d-1) facettes qui le définissent, tandis que les 2-faces correspondent à des sous ensembles de d-2 facettes...Puisque chaque sommet ne peut être supprimé qu'une fois après avoir été créé, le coût total de construction des 2-squelettes est proportionnel au coût de créations des sommets. Ce coût est lui-même majoré par le nombre de sommets créés au cours de l'algorithme auquel s'ajoute $\sum_i O(i) = O(n^2)$ pour le calcul des p_i . Soit m_{i+1} le nombre de sommets créés à l'étape i + 1, c'est-à-dire contenu dans H_{i+1}^0 . Fixons les (i + 1) premiers demi-espaces, et considérons toutes les permutations commençant par ces (i + 1) premiers éléments pris dans un ordre quelconque. Ceci fixe P_{i+1} . Puisque chaque sommet de P_{i+1} appartient à d hyperplans, la probabilité qu'un de ces sommets soit créé à l'étape i + 1 est au plus

d/(i+1). Par le théorème de la borne supérieure, la valeur moyenne de m_{i+1} est donc un $O(\frac{d}{i+1}(i+1)^{\lfloor d/2 \rfloor}) = O((i+1)^{\lfloor d/2 \rfloor-1})$. On en déduit que le coût total est

$$\sum_{i} O((i+1)^{\lfloor d/2 \rfloor - 1}) = O(n^{\lfloor d/2 \rfloor})$$

8.2 Diagramme de Voronoi

Soit $S = \{s_1, s_2, \ldots, s_n\}$ un ensemble de *n* points ou sites de \mathbb{R}^d . Pour tout $X \subset S$ on considère la région V_X formée des points de \mathbb{R}^d à égale distance des sites de X et strictement plus proches de X que de $S \setminus X$. Les V_X sont des polyèdres (relativement ouverts), car intersections de demi-espaces "médiateurs" de couples de sites, et partitionnent \mathbb{R}^d (pourquoi?). Cette partition est appelée le diagramme de Voronoi de S.

8.2.1 Lien avec les enveloppes supérieures

On note (x, x_{d+1}) un point de $\mathbb{R}^d \times \mathbb{R} = \mathbb{R}^{d+1}$. On considère dans \mathbb{R}^{d+1} le paraboloïde $U := \{x_{d+1} = |x|^2\}$. On relève chaque site s de S en un point $u(s) = (s, |s|^2)$ sur U et on note $T_U(s)$ l'hyperplan tangent à U en u(s) de sorte que $T_U(s) = \{x_{d+1} = 2\langle s, x \rangle - |s|^2\}$. Enfin, on note $H_U(s)$ le demi-espace bordé par $T_U(s)$ et tourné vers le haut : $H_U(s) = \{x_{d+1} \ge 2\langle s, x \rangle - |s|^2\}$.

Lemme 8.8 Soit $x \in \mathbb{R}^d$ et $s \in S$ un site, alors la distance algébrique verticale de $T_U(s)$ à u(x) vaut $|x - s|^2$.

Voir figure 8.1 pour une illustration.

Preuve : le projeté vertical y de x (ou u(x)) sur $T_U(s)$ vérifie $y_{d+1} = 2\langle s, x \rangle - |s|^2$. D'où $(u(x) - y)_{d+1} = |x|^2 - 2\langle s, x \rangle + |s|^2 = |x - s|^2$.

Proposition 8.9 Le diagramme de Voronoi de S est la projection verticale sur \mathbb{R}^d du bord du polyèdre $\bigcap_{s \in S} H_U(s)$, enveloppe supérieure des $T_U(s)$.

Dit autrement, les cellules du diagramme de Voronoi sont les projections des faces de l'enveloppe supérieure. Notons que les faces (possiblement vides) de $\bigcap_{s \in S} H_U(s)$ sont précisément de la forme $\bigcap_{s \in X} T_U(s) \bigcap \bigcap_{s \in S} H_U(s)$ pour $X \subset S$.¹

^{1.} Chaque $T_U(s)$ est support d'une facette de l'enveloppe supérieure car s est intérieur au polyèdre $\bigcap_{s' \neq s} H_U(s')$ puisque $s \in int(H_U(s'))$ pour $s' \neq s$. Le treillis d'un polytope P étant coatomique (proposition 6.33), chaque face de P est l'intersection de facettes, i.e. l'intersection avec P de leurs hyperplans supports. On étend sans mal à un polyèdre non borné.

Preuve : Soit $X \subset S$.

$$x \in V_X \Leftrightarrow \forall s_i, s_j \in X, \forall s_k \in S \setminus X : |x - s_i|^2 = |x - s_j|^2 < |x - s_k|^2.$$

Ce qui équivaut à dire d'après le lemme précédent que l'intersection de la verticale en xavec l'enveloppe supérieure des $T_U(s)$ est dans $\bigcap_{s \in X} T_U(s)$. C'est-à-dire que x est dans la projection verticale de la face de l'enveloppe supérieure, obtenue comme intersection de $\bigcap_{s \in X} T_U(s)$ avec $\bigcap_{s \in S} H_U(s)$.

Notons que cette projection est non dégénérée puisqu'aucun $H_U(s)$ ne contient la direction verticale.

Corollaire 8.10 Pour d fixé, la complexité totale en nombre de cellules du diagramme de Voronoi de n sites est un $O(n^{\lceil \frac{d}{2} \rceil})$.

Preuve : D'après le théorème de la borne supérieure (version duale) la complexité d'une intersection de *n* demi-espaces en dimension d + 1 est $O(n^{\lfloor \frac{d+1}{2} \rfloor})$.

En dimension 3 cela donne en particulier une borne supérieure en $O(n^2)$. Cette borne est atteinte pour 2n points répartis uniformément pour moitiés sur deux droites non sécantes et non parallèles de \mathbb{R}^3 .

8.2.2 Calcul en 2D

D'après ce qui précède la taille du diagramme de Voronoi de n sites du plan est linéaire.

Algorithme de Fortune

Intéressant du point de vue historique?

Diviser pour régner

1. On divise les points de S en deux selon la médiane de leurs abscisses en temps O(n).

2. On calcul récursivement le diagramme de Voronoi de chaque moitié ainsi que l'enveloppe convexe, et on fait la fusion (expliquée si dessous) en temps O(n).

$$T(n) = O(n) + T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil)$$

D'où $T(n) = O(n \log n)$, ce qui est optimal.

On note Vor(S) le diagramme de Voronoi de S et $V_X(S)$ la région de Voronoi associée à $X \subset S$ dans Vor(S). On note également G et D les moitiés gauche et droite de S calculées à l'étape 1. On pose enfin

$$U_G = \{ p \mid d(p, G) < d(p, D) \}$$

$$I = \{ p \mid d(p, G) = d(p, D) \}$$

$$U_D = \{ p \mid d(p, G) > d(p, D) \}$$

Lemme 8.11

$$Vor(G) \cap U_G = Vor(S) \cap U_G$$
 et $Vor(D) \cap U_D = Vor(S) \cap U_D$.

Preuve : pour la gauche :

$$\forall X \in S : p \in V_X(S) \cap U_G \implies X \subset G$$

d'où

$$p \in V_X(S) \cap U_G \Leftrightarrow p \in V_X(G) \cap U_G$$

-	_	_	
L			
L			
L			

Lemme 8.12 I est une chaîne (verticalement) monotone simple composée des arêtes de Vor(S) du type $V_{\{g,d\}}(S)$ où $g \in G$ et $d \in D$. Comme g est à gauche de d et que $V_{\{g,d\}}(S)$ est orthogonal à la droite gd, $V_{\{g,d\}}(S)$ n'est pas horizontale et peut être orientée vers le haut de sorte que g (d) est à sa gauche (droite).

Preuve : Soit $p \in I$. Par définition il existe $g \in G$ et $d \in D$ tels que d(p,g) = d(p,d) = d(p,S), donc $p \in V_{\{g,d\}}(S)$. Soit ℓ une droite horizontale. On veut montrer que ℓ intersecte I en un unique point. Clairement, il existe un point x de ℓ suffisamment à gauche tel que d(x,G) < d(x,D) et il existe un point y de ℓ suffisamment à droite tel que d(y,G) > d(y,D). Donc f(p) = d(p,G) - d(p,D) change de signe sur ℓ et s'annule nécessairement, fournissant ainsi un point q dans $\ell \cap I$. Soit \mathcal{C} le cercle de centre q et de rayon d(q,G) = d(q,D). \mathcal{C} est vide de points de S et toute droite h de séparation de G et D coupe \mathcal{C} . Supposons que h est à droite de q. Clairement tout $x \in \ell$ à droite de \mathcal{C} est plus proche de d que de tout point à gauche de h (et hors de \mathcal{C}), donc f(x) > 0. Si q est le point le plus à gauche sur ℓ vérifiant f(q) = 0, c'est donc le seul. Par conséquent $|\ell \cap I| = 1$.

La fusion des enveloppes convexes de G et D s'obtient à l'aide des deux ponts haut et bas comme décrit section 8.1.2. Les médiatrices de ces ponts supportent nécessairement les segments (infinis) supérieur et inférieur de I. Notons g_0 et d_0 les sites extrémités du pont supérieur. Pour calculer Vor(S) à partir de Vor(G) et Vor(D), et donc pour calculer I, on commence par la partie "supérieure" de la médiatrice I_0 de g_0 et d_0 qui est nécessairement contenue dans $V_{\{g_0\}}(G)$ et dans $V_{\{d_0\}}(D)$. On parcourt en parallèle le bord de ces régions pour trouver le segment de leur bord qui intersecte I_0 le plus haut. Ce segment, que l'on supposera border $V_{\{g_0\}}(G)$ est une région $V_{\{g_0,g_1\}}(G)$. Le point $i_1 = V_{\{g_0,g_1\}}(G) \cap I$ vérifie $d(i_1,g_0) = d(i_1,g_1) = d(i_1,d_0) = d(i_1,S)$. Donc i_1 est un sommet de I et le segment suivant sur I ne peut être que $V_{\{g_1,d_0\}}(S)$. On poursuit sur ce segment jusqu'à intersecter un segment de $V_{\{g_1\}}(G)$ ou $V_{\{d_0\}}(D)$. On itère ce procédé jusqu'à rencontrer la médiatrice du pont inférieur. Le coût de construction de I est de l'ordre de la taille totale des régions parcourues qui est elle même majorée par la somme des tailles de Vor(G) et Vor(D) et est donc linéaire.

8.2.3 Variantes du diagramme de Voronoi

Diagramme de puissance. Rappelons que par définition la *puissance* d'un point x par rapport à une sphère S(c, r) de centre c et de rayon r vaut $|x - c|^2 - r^2$ et que l'ensemble des points ayant même puissance relativement à deux sphères de centres distincts est un hyperplan (dit *radical*). Étant donné un ensemble de sphères de \mathbb{R}^d , ou de manière équivalente de points pondérés, le *diagramme de puissance* de ces sphères est la partition de \mathbb{R}^d en régions telle que dans chaque région tout point ait une même puissance par rapport à un sous-ensemble fixe de sphères et une puissance plus grande par rapport aux autres sphères. Comme pour le diagramme usuel, les régions sont des polyèdres (intersections de demi-espaces bordés par des hyperplans radicaux). De manière analogue au diagramme de Voronoi usuel, un diagramme de puissance s'obtient comme projection de l'enveloppe supérieure d'hyperplans obtenus en translatant les hyperplans tangents aux relevés des sites sur le paraboloïde U (cf. section 8.2.1). Certains sites peuvent cependant avoir des régions vides contrairement au cas du diagramme usuel.

Diagramme d'ordre k. Soit une famille S de n sites de \mathbb{R}^d et un entier k < n. On associe à chaque k-set (sous-ensemble de taille k) $T \subset S$ la cellule formées de l'ensemble des points qui sont plus proches des sites de T que de $S \setminus T$, i.e. $\{x \mid \forall t \in T, \forall s \in S \setminus T :$ $d(x,t) \leq (x,s)$. Les intérieurs relatifs des intersections de ces cellules constituent une partition de \mathbb{R}^d appelée diagramme de Voronoi d'ordre k. On montre aisément que c'est la projection verticale du k-ième niveau de l'arrangement des $T_U(s)$ (cf. section 8.2.1 et figure 8.1) relativement à la direction verticale vers le bas : c'est l'ensemble des cellules



FIGURE 8.1 – En haut : Lien entre la distance entre les sites s et x et la distance verticale du relevé x^* sur le paraboloïde au plan tangent à s^* . Ici U(s) est noté s^* . En bas à gauche : Le diagramme de Voronoi des points sur la droite horizontale apparaît comme la projection de l'enveloppe supérieure des plans tangents aux relevés. En bas à droite : Le diagramme d'ordre 2 du même ensemble de points.

de cet arrangement qui ont k hyperplans au dessus d'elles (au sens large). Le diagramme d'ordre k = n - 1 s'appelle encore le diagramme du voisin le plus loin. La cellule d'un site (i.e. des n-1 sites complémentaires) est l'ensemble des centres des sphères contenant tous les autre sites. C'est encore la projection verticale de l'enveloppe inférieure des $T_U(s)$.

Autres diagrammes. On peut encore considérer des diagrammes de Voronoi pour des distances autres que la distance Euclidienne. On peut également remplacer les sites ponctuels par des objets plus variés comme des segments, des convexes, des droites, etc...La littérature sur ce sujet est sans fin [AK00].

8.3 Triangulation de Delaunay

Soit $S = \{s_1, s_2, \ldots, s_n\}$ un ensemble de *n* points ou sites de \mathbb{R}^d . On considère les $X \subset S$ tels qu'il existe une boule fermée B_X , circonscrite à X et vérifiant $B_X \cap S = X$. On pose $D_X = Conv(X)$. Le diagramme de Delaunay de S est la collection des D_X ainsi formés. Le bord d'une boule B_X comme ci-dessus s'appelle une sphère de Delaunay.

8.3.1 Lien avec les enveloppes convexes

Lemme 8.13 Soit $X = \{x_1, x_2, \ldots, x_{d+1}\}$ un ensemble de d+1 points de \mathbb{R}^d affinement indépendants. Alors, il existe une unique sphère $\mathcal{S}(X)$ circonscrite à ces points et elle a pour équation

$ x ^2$	$ x_1 ^2$	 $ x_{d+1} ^2$	
x	x_1	 x_{d+1}	= 0
1	1	 1	

Preuve : En développant ce déterminant selon la première colonne on trouve une équation de la forme $\alpha |x|^2 + \langle \beta |x \rangle + \gamma = 0$ avec $\alpha = \begin{vmatrix} x_1 & \dots & x_{d+1} \\ 1 & \dots & 1 \end{vmatrix}$ non nul puisque les points de X sont affinement indépendants. Cette équation est donc celle d'une sphère. Cette sphère est circonscrite aux x_i puisque ce déterminant s'annule pour $x = x_i$, pour tout i.

Par ailleurs, si $|x|^2 + \langle \beta | x \rangle + \gamma = 0$ et $|x|^2 + \langle \beta' | x \rangle + \gamma' = 0$ sont les équations de deux sphères circonscrites à X, alors $\langle \beta - \beta' | x \rangle + \gamma - \gamma'$ s'annule sur X ce qui implique $\beta = \beta'$ et $\gamma = \gamma'$ puisque X n'est contenu dans aucun hyperplan.

On rappelle la notation $u(x) = (x, |x|^2)$ de la section 8.2.1. Soit $X = \{x_1, x_2, \dots, x_{d+1}\}$ un ensemble de d+1 points affinement indépendants. On note H(X) l'unique hyperplan contenant

 $u(X) = \{u(x_1), u(x_2), \dots, u(x_{d+1})\}$. On déduit du lemme précèdent que

$$x \in \mathcal{S}(X) \Leftrightarrow u(x) \in H(X).$$

Plus précisément si $\mathcal{B}(X)$ est la boule fermée bordée par $\mathcal{S}(X)$ et $Int \mathcal{B}(X)$ est son intérieur, on vérifie que

Lemme 8.14

 $x \in Int \mathcal{B}(X) \Leftrightarrow u(x) \text{ est en dessous de } H(X)$

et

$$x \notin \mathcal{B}(X) \Leftrightarrow u(x)$$
 est au dessus de $H(X)$

Corollaire 8.15 Soit $X \subset S$. Il existe une boule fermée B_X circonscrite à X et vérifiant $B_X \cap S = X$ si et seulement s'il existe un hyperplan H_X de \mathbb{R}^{d+1} tel que u(S) est au dessus de H_X et $H_X \cap u(S) = u(X)$.

Preuve : \Rightarrow Soit Y un ensemble de d+1 points de B_X affinement indépendants. D'après ce qui précède on peut prendre $H_X = H(Y)$.

 \Leftarrow Soit Y un ensemble de d+1 points de $H_X \cap U$ tels que leurs projections soit affinement indépendantes. On peut poser poser $B_X = \mathcal{B}(Y)$. \Box

Donc D_X est une cellule de Delaunay si et seulement s'il existe un hyperplan non vertical au dessous de u(S) et qui contient précisément u(X), ce qui équivaut encore à dire que Conv(u(X)) est une face de Conv(u(S)) possédant un demi-espace support tourné vers le haut. En appelant *enveloppe convexe inférieure* la collection des faces d'un polytope qui possèdent un demi-espace support tourné vers le haut, on obtient finalement,

Proposition 8.16 Del(S) est la projection verticale de l'enveloppe convexe inférieure de u(S). En particulier, Del(S) constitue une subdivision polytopale² de Conv(S).

Notons que cette projection est non dégénérée puisque chaque face de l'enveloppe convexe inférieure est contenue dans un hyperplan non vertical.

Corollaire 8.17 Pour d fixé, la complexité totale en nombre de cellules du diagramme de Delaunay de n sites est un $O(n^{\lceil \frac{d}{2} \rceil})$.

On dira que S est en position générale si d + 1 points quelconques de S sont affinement libres et si d + 2 points de S ne peuvent être cosphériques. En particulier, il passe une unique sphère par d + 1 points quelconques de S.

Remarque 8.18 Si S est en position générale, les D_X sont des simplexes. Dans ce cas Del(S) est une triangulation de Conv(S).

^{2.} Une subdivision polytopale d'un espace est un complexe polytopal dont cet espace est l'espace total. Un complexe polytopal est une collection de polytopes telle que toute face de ces polytopes est dans la collection et telle que deux polytopes de la collection s'intersectent selon une face commune (possiblement vide).

Exercice 8.19 Montrer un résultat analogue au lemme 8.13 lorsqu'on suppose seulement dim aff(X) = d - 1 et que les points de X ne sont pas cocycliques (on trouve une équation qui se ramène à celle d'un hyperplan).

Exercice 8.20 Démontrer le lemme 8.14.

En deux dimensions, la liste ordonnée des angles des triangles de Delaunay est maximale pour l'ordre lexicographique sur toutes les triangulations de S.

Une arête est dite *localement de Delaunay* si la sphère circonscrite à l'un de ses deux triangles incidents ne contient pas le troisième sommet de l'autre triangle incident. Si une arête n'est pas localement de Delaunay alors l'arête obtenue par *flip* est localement de Delaunay.

Proposition 8.21 Une triangulation est de Delaunay si et seulement si toutes ses arêtes sont localement de Delaunay.

8.3.2 Dualité entre les diagrammes de Delaunay et de Voronoi

Soit S un ensemble fini de points de \mathbb{R}^d .

Proposition 8.22 Pour tout $X \subset S$, $V_X \in Vor(S) \Leftrightarrow D_X \in Del(S)$.

Preuve : $V_X \neq \emptyset \Leftrightarrow \exists y \ (\in V_X)$ tel que y est le centre d'une sphère de Delaunay pour $X \Leftrightarrow D_X \in Del(S)$.

On note \bar{V}_X l'adhérence de V_X . Comme V_X est l'intérieur relatif d'un polytope, \bar{V}_X désigne l'union de V_X avec ses faces.

Théorème 8.23 Il existe une dualité entre le diagramme de Voronoi et le diagramme de Delaunay de S qui associe toute cellule V_X de Voronoi à la cellule D_X de Delaunay de sorte que

$$\dim V_X = i \Leftrightarrow \dim D_X = d - i$$
$$\bar{V}_X \subset \bar{V}_Y \Leftrightarrow D_Y \subset D_X.$$

Preuve : On donne ici une preuve s'appuyant sur la dualité des cônes polyédriques et sur le lien entre les diagrammes de Voronoi ou Delaunay et la projection de certains polyèdres.

Soit S un ensemble de sites de \mathbb{R}^d . On considère à nouveau le paraboloïde U et le relèvement $u(x) = (x, |x|^2)$ dans \mathbb{R}^{d+1} . Par la proposition 8.9, le diagramme de Voronoi de S est la projection verticale sur \mathbb{R}^d du bord du polyèdre $P = \bigcap_{s \in S} H_U(s)$. Soit $P' = e_{d+2} + P$ le translaté de P par e_{d+2} dans \mathbb{R}^{d+2} . On considère le cône C sur P' de sommet **0**. Rappelons que $H_U(s) = \{2\langle s, x \rangle - x_{d+1} - |s|^2 \leq 0\}$, de sorte que C est l'intersection des demi-espaces vectoriels $\{h_s(x, x_{d+1}, x_{d+2}) \leq 0\}, s \in S$, où $h_s(x, x_{d+1}, x_{d+2}) = 2\langle s, x \rangle - x_{d+1} - |s|^2 x_{d+2}$. Le lemme 6.52 indique que les faces de P sont en correspondance avec les faces de C qui intersectent l'hyperplan $E = \{x_{d+2} = 1\}$. Par la proposition 6.50, l'intérieur relatif de ces faces est de la forme

$$intF = \{ z \mid \forall s' \in S', \forall s'' \in S'' : h_{s'}(z) = 0 \text{ et } h_{s''}(z) < 0 \}$$

pour une partition $S' \cup S''$ de S telle que $E \cap intF$ est non vide (cf. exercice ci-dessous), c'est-à-dire telle qu'il existe un point $(x, x_{d+1}, 1)$ de E vérifiant

$$\forall s' \in S', \forall s'' \in S'' : 2\langle s', x \rangle - |s'|^2 = x_{d+1} \text{ et } 2\langle s'', x \rangle - |s''|^2 < x_{d+1}$$

Ce qui s'écrit encore, en notant $g_x(y, y_{d+1}) = 2\langle y, x \rangle - y_{d+1}$:

$$\forall s' \in S', \forall s'' \in S'' : g_x(u(s')) = x_{d+1} \text{ et } g_x(u(s')) < x_{d+1}$$

On en déduit que le demi-espace $\{g_x \leq x_{d+1}\}$ de \mathbb{R}^{d+1} est support pour la face Conv(u(S'))du polytope Q = Conv(u(S)). On note de plus que ce demi-espace est tourné vers le haut (le coefficient de u_{d+1} dans h_x est -1). Réciproquement, toute face de Q admettant un hyperplan support tourné vers le haut, donc de la forme $H = \{y_{d+1} \geq \langle a, y \rangle - b\}$, correspond à une partition $S' \cup S''$ comme ci-dessus avec $S' = S \cap H$, puisque

$$\forall s' \in S', \forall s'' \in S'' : g_a(u(s')) = b \text{ et } g_a(u(s'')) < b$$

Par la proposition 6.54, le cône C^* dual de C est l'enveloppe conique des vecteurs $v_s = (2s, -1, -|s|^2), s \in S$. De plus, par la proposition 6.58 le treillis des faces de C est opposé au treillis des faces de C' et par le lemme 6.56 une face F comme ci-dessus a pour duale la face $F^{\#}$, enveloppe conique des $v_s, s \in S'$. Mais en appliquant l'isomorphisme linéaire de matrice

$$M = \begin{pmatrix} \frac{1}{2}I_d & 0 & 0\\ 0 & 0 & -1\\ 0 & -1 & 0 \end{pmatrix}$$

aux vecteurs v_s , on déduit que C^* est isomorphe à l'enveloppe conique des vecteurs $Mv_s = (s, |s^2|, 1) = (u(s), 1), s \in S$. Par le lemme 6.52, les faces de C^* sont donc en correspondance avec les faces de l'enveloppe convexe Q des points $u(s), s \in S$. Cette correspondance envoie la face $F^{\#}$ de C^* sur la face Conv(u(S')) de Q.

Pour conclure, les cellules de Voronoi de dimension k correspondent aux faces de P de même dimension qui correspondent aux faces de C de dimension k + 1 intersectant E. Ces faces sont elles mêmes en correspondance avec certaines faces de dimensions d + 2 - (k + 1) = d - k + 1 de C^* qui correspondent précisément aux faces de Q de dimension d - k possédant un hyperplan support tourné vers le haut. Ces dernières se projettent sur les cellules de Delaunay de même dimension par la proposition 8.16. \Box

Exercice 8.24 Soit C un cône et E un hyperplan ne contenant pas l'origine. Montrer que $C \cap E$ est non vide si et seulement si E intersecte l'intérieur relatif de C.

8.3.3 Algorithme randomisé incrémental pour Delaunay

1. On commence par calculer un grand triangle contenant S.

2. On calcul incrémentalement $Del(S_r)$ en insérant s_r dans $Del(S_{r-1})$.

Après avoir trouvé le triangle Δ de $Del(S_r)$ contenant s_r on étoile ce triangle à partir s_r . Si s_r est sur une arête alors on supprime cette arête et on étoile le quadrilatère restant à partir s_r . Les 3 (ou 4) arêtes incidentes à s_r sont nécessairement (localement) de Delaunay : on peut trouver une sphère circonscrite à ces arêtes et incluse dans la sphère (initialement vide) de Δ . Seules les arêtes de Δ peuvent ne plus être localement de Delaunay. On flippe les arêtes qui ne le seraient plus, ce qui crée autant d'arêtes incidentes à s_r et deux fois plus de nouveaux triangles. Les arêtes opposées à s_r dans ces nouveaux triangles peuvent ne plus être localement de Delaunay et on continu ainsi jusqu'à ce que toutes les arêtes soient localement de Delaunay. Le nombre de triangles créés est au plus $2d_r - 3$ où d_r est le degré de s_r dans $Del(S_r)$.

Afin de localiser efficacement s_r dans $Del(S_{r-1})$ – i.e. de trouver le triangle de $Del(S_{r-1})$ qui contient s_r – on maintient une structure de recherche sous forme d'un DAG dont les noeuds sont les triangles créés durant l'algorithme et les feuilles sont les triangles de la triangulation courante. Chaque triangle pointe vers les triangles qui le remplacent et l'intersectent. Le degré sortant maximal dans ce DAG est donc 3.

Le coût total de l'algorithme est proportionnel au nombre de triangles créés plus le nombre de triangles détruits plus la somme des longueurs des chemins de recherche dans le DAG. Notons que le nombre de triangles détruits est inférieur au nombre de triangles créés. Une borne sur ce nombre est donnée par le lemme suivant.

Lemme 8.25 Le nombre moyen de triangles créés durant l'algorithme est majoré par 9n + 1.

Preuve : On note $\mathcal{A}(S_r)$ le nombre de triangles de $Del(S_r)$. D'après ce qui précède le nombre de triangles créés à l'étape r est majoré par $2d_r - 3$. À S_r fixé, s_r est l'un quelconque des points de S_r de manière équiprobable donc

$$E(|\mathcal{A}(S_r) \setminus \mathcal{A}(S_{r-1})| \,|\, S_r) \le \frac{1}{r} \sum_{s \in S_r} (2d(s) - 3)$$

Or $Del(S_r)$ étant une triangulation sur r + 3 sommets bordée par un triangle on a $|\mathcal{A}(S_r)| = 3(r+3) - 6 = 3r + 3$. Si on omet les 3 sommets du triangle initial qui ont degré deux au moins on obtient

$$\sum_{s \in S_r} d(s) \le 2|\mathcal{A}(S_r)| - 6 = 2(3r+3) - 6 = 6r$$

Par conséquent le nombre moyen de triangles créés à l'étape r est majoré par 9. Si on tient compte du triangle initial on peut conclure par le résultat annoncé.

Il reste à estimer la somme des longueurs des chemins de recherche dans le DAG. Le chemin de recherche de s_r visite tous les triangles créés avant l'étape r et contenant s_r .

Chaque noeud ayant au plus trois enfants, ceci représente bien le coût de la recherche à un facteur 3 près. Si on compte à part le triangle contenant s_r dans $\mathcal{A}(S_{r-1})$, le coût de la recherche est proportionnel à 1 plus le nombre de triangles créés *et* détruits avant l'étape r et contenant s_r . Ces triangles étaient soient des triangles de Delaunay à une étape antérieure soit des triangles intermédiaires créés et détruits durant une même étape. Dans ce dernier cas le triangle adjacent qui a servi au flip était effectivement un triangle de Delaunay et son cercle circonscrit contenait le triangle détruit et donc s_r . On peut ainsi reporter dans tous les cas le coût de visite d'un triangle dans le DAG sur un triangle de Delaunay dont le cercle circonscrit contenait s_r . Notons que ces triangles de report sont tous distincts. Un triangle de Delaunay dont le cercle circonscrit continu un site est dit *en conflit* avec ce site. Le coût total de la recherche est donc borné par n (1 par site) plus le nombre total de tous les conflits entre des triangles de Delaunay créés (et détruits) au cours de l'algorithme et des sites insérés par la suite. Par le lemme 13.8, ce nombre est borné en moyenne par

$$\sum_{r=1}^{n} 3^2 \frac{n-r}{r^2} E(|\mathcal{A}(S_r)|).$$

Comme $|\mathcal{A}(S_r)| = O(r)$, on en déduit que ce coût moyen est un $O(n \log n)$.

Proposition 8.26 La triangulation de Delaunay de n sites dans le plan peut être calculée par un algorithme randomisé en temps moyen $O(n \log n)$.

Chapitre 9

Arrangements

9.1 Introduction : problème de la discrépance

On se donne un ensemble fini \mathcal{P} de points dans le carré unité et on compare la mesure de l'intersection d'un demi-plan avec le carré avec la proportion de points de \mathcal{P} contenu dans cette intersection. On cherche à évaluer la *discrépance*, c'est-à-dire le maximum de la différence entre ces valeurs pour tous les demi-plans possibles. Ce maximum est obtenu avec une droite qui passe soit par au moins deux points de \mathcal{P} soit par un seul point mais pour des positions extrêmes de la droite, dues à la forme du carré.

On traite explicitement ces derniers cas en temps $O(n^2)$: pour chaque point de \mathcal{P} on calcule les positions des droites passant par ce point et coupant le carré en deux de manière extrême (des extremums locaux pour l'aire délimitée par les droites). Chaque position extrême se calcule en temps constant et on obtient la proportion de points de \mathcal{P} de chaque côté de la droite obtenue en temps O(n) par simple test d'inclusion sur chaque point. Puisque les positions extrêmes sont en nombre borné pour chaque point de \mathcal{P} on obtient la discrépance en temps $O(n^2)$ pour l'ensemble de ces situations extrêmes.

Le traitement des $\binom{n}{2}$ droites passant par deux points de \mathcal{P} peut se faire simplement en calculant les proportions associées en temps O(n) par droite. Ceci fournit trivialement un algorithme de complexité $O(n^3)$ pour le calcul de la discrépance. L'utilisation des arrangements de droites permet cependant de concevoir un algorithme plus efficace. On note pour cela que les points duaux des droites passant par deux points de \mathcal{P} sont des sommets de l'arrangement des droites duales de \mathcal{P} . Le nombre de points du dessus/dessous d'une droite d est le nombre de droites duales de ces points au dessous/dessus du point dual de d. Pour calculer ces nombres, on construit explicitement la carte de l'arrangement dual de \mathcal{P} et on marche le long de chaque droite à partir d'une extrémité. Il est facile de maintenir pour chaque sommet rencontré au cours de cette marche son *niveau*, qui est le nombre de droites duales (des points de \mathcal{P}) strictement au dessus de ce sommet. Le calcul des niveaux, et donc de la discrépance, s'obtient ainsi en temps proportionnel à la taille $O(n^2)$ de l'arrangement.

9.2 Préliminaire : subdivision du plan

Une k-cellule du plan est un sous-espace du plan homéomorphe à une k-boule ouverte. Une subdivision du plan est un ensemble fini de cellules disjointes de dimension 0, 1 et 2 du plan dont l'union est le plan de sorte que la frontière d'une cellule est une union de cellules de dimensions inférieures. Dans la pratique on considère souvent que le plan est compactifié par l'ajout d'un point "à l'infini" de sorte que la subdivision porte véritablement sur la sphère et non sur le plan. On pourra également consulter le chapitre 2 de ces notes sur ce point.

Un graphe est planaire s'il peut être plongé dans le plan. Un graphe plan – ou carte planaire – est un graphe plongé dans le plan. Tout graphe plan connexe induit une subdivision du plan dont les *faces* sont les composantes bornées du complémentaire du graphe.

Théorème 9.1 (Relation d'Euler) Tout graphe plan connexe ayant s sommets, a arêtes et f faces vérifie : s - a + f = 1.

Preuve : Par récurrence sur le nombre d'arêtes. Trivial si a = 1. Si le graphe contient un cycle, alors on ôte une arête de ce cycle qui borde deux faces distinctes (par le théorème de séparation Jordan). L'hypothèse de récurrence s'applique donc avec une face et une arête en moins. Sinon le graphe est sans cycle et contient un sommet de degré un. En ôtant ce sommet et l'arête incidente on peut à nouveau appliquer l'hypothèse de récurrence avec un sommet et une arête en moins.

Note : généralement on considère la relation d'Euler pour un graphe tracé sur la sphère. On passe du plan à la sphère via une projection stéréographique. La face non-bornée du plan devient une face pour la sphère d'où la relation d'Euler pour la sphère : s-a+f=2. Soit $\pi : G \to \mathbb{R}^2$ un plongement polygonal d'un graphe planaire G (les plongements des arêtes sont donc des arcs polygonaux). On définit pour chaque sommet s du graphe un ordre circulaire sur les arêtes a_1, \ldots, a_k incidentes à ce sommet par l'ordre circulaire dans le sens indirecte autour de $\pi(s)$ des segments de $\pi(a_1), \ldots, \pi(a_k)$ incidents à $\pi(s)$. Si le plongement n'est pas polygonal on peut encore définir un ordre circulaire autour de chaque sommet via un homéomorphisme du plan qui transforme les arcs intérieurs à un petit disque centré en $\pi(s)$ en des arcs polygonaux tout en laissant fixe l'extérieur du disque. On montre [MT01, p. 88] que cet ordre est indépendant de l'homéomorphisme employé. La donnée combinatoire du graphe G avec ces ordres circulaires est appelé une *carte combinatoire* ou système de rotations.

On montre [MT01, cor. 3.2.5] qu'une carte combinatoire détermine un unique plongement de G à homéomorphisme près. Dit autrement, si $\pi_1, \pi_2 : G \to \mathbb{R}^2$ sont deux plongements de G qui induisent la même carte combinatoire alors il existe un homéomorphisme ϕ du plan tel que $\pi_2 = \phi \circ \pi_1$. Dans la pratique on représente souvent une carte combinatoire à l'aide d'une liste de *demi-arêtes* représentant chacune une des deux orientations possibles de chaque arête du graphe. À chaque demi-arête a on adjoint deux pointeurs *inv* et *rot* respectivement vers la demi-arête d'orientation opposée et vers la demi-arête suivant a dans le sens indirecte autour du sommet origine de a. Notons que la demi-arête suivant a dans la face située à gauche de a (en regardant de l'origine vers la pointe de la demiarête) s'obtient en suivant le pointeur *inv* puis *rot*. La structure porte souvent le nom de DCEL pour Doubly-Connected Edge List. Une telle structure peut être définie pour un plongement dans la sphère et plus généralement dans une surface orientable de genre quelconque.

Références :

```
- [MT01]
```

- Primitives for the manipulation of three-dimensional subdivisions. L. Guibas and J. Stolfi, ACM Transactions on Graphics, 1985, 4(2), pp 74-123.

9.3 Arrangement de droites

Définition 9.2 Un arrangement d'un ensemble de droites du plan est la décomposition du plan induite par ces droites. Les 2-faces ou cellules de cet arrangement sont les composantes connexes du complémentaire des droites, les 1-faces et 0-faces sont les faces des fermetures de ces cellules. Un arrangement est simple si deux droites se coupent toujours mais trois droites ont une intersection vide.

Théorème 9.3 Un arrangement de n droites a au plus $\binom{n}{2}$ sommets, n^2 arêtes et $\binom{n}{2} + n + 1$ faces.

Preuve : Ces bornes sont exactes pour un arrangement simple : chaque paire de droites s'intersecte en un sommet et chaque droite est coupée en n segments par les n-1 autres droites. Par la relation d'Euler, obtenue sur la sphère en ajoutant un sommet à l'infini au plan, on obtient le nombre de faces. On peut se passer de la relation d'Euler en utilisant un balayage par une droite non parallèle à celle de l'arrangement. Le nombre de faces à gauche de cette droite passe de 0, à l'extrême gauche, au nombre de sommets, une fois à droite du dernier sommet de l'arrangement, auquel on doit ajouter n+1 faces à l'extrême droite. Ces égalités deviennent des majorations dans le cas d'un arrangement quelconque. En effet, en perturbant de manière infinitésimale un arrangement non simple \mathcal{A} on peut obtenir un arrangement simple \mathcal{A}' dont le nombre de faces majore celui de \mathcal{A} en toute dimension. Pour cela, il suffit de déplacer chaque droite une à une dans un "voisinage tubulaire", ne contenant aucun autre sommet de l'arrangement que ceux sur cette droite, de sorte que tous les sommets sur cette droite deviennent simples et que cette droite ne soit parallèle à aucune autre. Chaque droite avant perturbation étant coupée au plus n-1 fois, le nombre de segments de \mathcal{A}' majore celui de \mathcal{A} . De même pour le nombre de sommets, puisque chaque paire de droites définie au plus un sommet. Pour obtenir une majoration du nombre de facettes on peut associer à chaque facette de \mathcal{A} un point intérieur. Ces points restent intérieurs au complémentaire des droites après perturbation (suffisamment petite pour cela) et restent dans des composantes connexes distinctes car tout segment joignant deux de ces points coupe une droite de l'arrangement avant et après perturbation.



FIGURE 9.1 – La zone de la droite ℓ . L'ensemble E correspondant est constitué des 3 arêtes en trait (rouge) épais.



FIGURE 9.2 – La région diédrale supérieure entre ℓ_i et la droite support de e est strictement au dessus de ℓ . Comme $e' \in Z(\ell)$, il suit que ℓ_i sépare e' et e. Une troisième arête (en vert) incidente à ℓ_i et au dessus de e et e' est nécessairement hors de la zone $Z(\ell)$.

Définition 9.4 La zone d'une droite dans un arrangement est l'union des faces de l'arrangement dont la fermeture intersecte cette droite, et des bords de ces faces. La complexité de cette zone est le nombre total de faces, arêtes et sommets qu'elle contient.

Théorème 9.5 (de la zone) Le nombre d'arêtes de la zone d'une droite dans un arrangement de n droites du plan est majoré par 6n.

Preuve 1 : Soient $L = \{\ell_1, \ell_2, \ldots, \ell_n\}$ les *n* droites d'un arrangement et ℓ une autre droite dont la direction est dite horizontale (pente nulle). On considère l'ensemble *E* des arêtes de la zone $Z(\ell)$ de ℓ dont la fermeture est strictement au dessus de ℓ . On associe à toute arête *e* non-horizontale de *E* la droite $\phi(e) \in L$ qui suit la droite support $\ell(e)$ de *e* en tournant dans le sens indirect autour de l'extrémité inférieure de *e* (cf. figure 9.1). Si *e* est horizontale et bornée à gauche, on définit $\phi(e)$ comme la droite qui suit $\ell(e)$ en tournant dans le sens indirect autour de l'extrémité gauche de *e*. Notons que $\phi(e)$ ne peut être horizontale car *e* serait alors hors $Z(\ell)$. Montrons que pour toute $\ell_i \in L$ on a $|\phi^{-1}(\ell_i)| \leq 2$. Dans le contraire on note *e* et *e'* les deux arêtes de $\phi^{-1}(\ell_i)$ (dont les extrémités sur ℓ_i sont) les plus basses. Les deux dièdres "supérieurs" définis par ℓ_i et $\ell(e)$ d'une part et par ℓ_i et $\ell(e')$ d'autre part sont de part et d'autre de ℓ_i et au dessus de ℓ (cf. figure 9.2). Leur intérieur est donc disjoints de $Z(\ell)$. Les autres arêtes de $\phi^{-1}(\ell_i)$ sont contenues dans l'intérieur de ces dièdres et ne sont donc pas incidentes à une face de $Z(\ell)$, une contradiction. On déduit de ce qui précède $|E| \leq 2n$. (S'il y a une droite horizontale alors $|E| \leq 2(n-1) + 1$).

Le nombre d'arêtes supérieures ou inférieures de la zone et non-incidentes à ℓ est donc

majoré par 4n. Par ailleurs, le nombre maximal d'arêtes de $Z(\ell)$ incidentes à ℓ est majoré par 2n puisque chaque droite de L est le support d'au plus deux telles arêtes. \Box

La preuve qui suit permet de majorer par 8n - 4 le nombre de couples (e, f) où e est une arête incidente à une face f de la zone.

Preuve 2 : Avec les notations de la preuve 1, on oriente les ℓ_i non horizontales vers le bas et les autres vers la droite. On considère ensuite une droite ℓ' infinitésimalement translatée au dessus de ℓ . Cette droite coupe les faces dont la fermeture intersecte ℓ et situées (partiellement) au dessus de ℓ selon une suite ordonnée de gauche à droite. Pour chacune de ces faces on écrit la chaîne des droites supports de ses arêtes situées au dessus de ℓ dans un parcours dans le sens indirecte du bord de la face, en commençant par une arête incidente à ℓ . On ajoute par ailleurs un signe + (resp. -) à chaque droite support selon son sens de parcours. La concaténation de gauche à droite de ces listes fournit un mot sur les symboles $\pm \ell_i$. On scinde ce mot en deux sous-mots L_+ et L_- selon les signes des symboles.

Remarque : Une arête a non horizontale (resp. horizontale) de L_+ a sa face f(a) à gauche (resp. en dessous) de sa droite support l(a) en raison du sens de parcours des faces. Par conséquent toutes les faces à gauche de f(a) (relativement à ℓ') sont également à gauche de $\ell(a)$.

Les sous-mots L_+ et L_- vérifient les propriétés suivantes :

- ils ne contiennent pas de facteur $\ell_i \ell_i$ car deux arêtes portées par les mêmes droites appartiennent à des faces distinctes séparées par un chemin joignant ℓ_i à ℓ . Le dernier segment de ce chemin est porté par une droite $\ell_j \neq \ell_i$ et est parcouru dans les deux sens. Autrement dit pour toute double occurrence de ℓ_i dans L_+ ou L_- il existe un ℓ_j qui les sépare.
- ils ne contiennent pas de sous mot de la forme $\ell_i \ell_j \ell_i \ell_j$. Supposons que $\ell_i \ell_j \ell_i$ est sousmot de L_+ . Soient a_g , b et a_d les segments correspondant respectifs. On forme un cycle avec la chaîne gauche c_g (resp. droite c_d) de $f(a_g)$ (resp. $f(a_d)$) comprise entre ℓ et ℓ_i et les segments des droites ℓ et ℓ_i joignant $f(a_g)$ et $f(a_d)$. Ce cycle est convexe donc simple. b rencontre l'intérieur de cette région de part sa position dans L_+ . ℓ_j ne rencontrant aucune des deux chaînes c_g et c_d rencontre ℓ_i entre a_g et a_d (ainsi que ℓ (Jordan again)). On en déduit que a_d est à droite de ℓ_j (qui ne peut être horizontale). La remarque précédente montre qu'une nouvelle occurrence de ℓ_j dans L_+ impliquerait que a_d est à gauche de ℓ_j , une contradiction.

De tels mots sont appelés (n,2)-suites de Davenport-Shinzel. Ce sont précisément des mots formés sur une alphabet à n lettres, ne comportant pas de facteur aa ni de sous-mot abab. Le lemme suivant permet de conclure.

Lemme 9.6 Une (n,2)-suite de Davenport-Shinzel est de longueur au plus 2n-1.

Preuve : Soit S une (n, 2)-suite de Davenport-Shinzel. Le lemme est trivial si chaque lettre n'apparaît qu'une fois. Sinon on considère le plus petit facteur S' de S séparé par deux occurrences d'un même lettre de sorte que $S = S_1 a S' a S_2$. En posant k = |S'|, on vérifie que $S_1 a S_2$ est une (n - k, 2)-suite de Davenport-Shinzel. On conclut par récurrence sur n que $|S| \leq 2(n - k) - 1 + k + 1 \leq 2n - 1$.

Corollaire 9.7 La complexité de la zone d'une droite dans un arrangement de n droites du plan est linéaire (i.e. en O(n)).

Preuve : Le théorème précédent borne le nombre d'arêtes de la zone. Le nombre de faces est par ailleurs majoré par 2n (exercice !) et le nombre de sommets est borné par le nombre d'arêtes de la zone.

Exercice 9.8 Montrer que le nombre de faces de la zone est bien majoré par 2n et donner un exemple où cette borne est atteinte.

On construit incrémentalement la carte d'un arrangement en introduisant chaque droite une à une dans la carte déjà construite. Initialement, on introduit une première droite en ajoutant un sommet à l'infini p_{∞} à la carte qui représente donc une sphère (le plan compactifié). Chaque droite introduite est donc doublement incidente à ce sommet. Pour introduire une nouvelle droite ℓ , on l'oriente et on part de la face ayant deux arêtes incidentes à p_{∞} dont les deux vecteurs directeurs (issus de p_{∞}) encadrent celui de ℓ . On marche dans cette face jusqu'à trouver la sortie – une arête ou un sommet coupé par ℓ – et on modifie la carte en conséquence. On poursuit avec la face suivante : si on sort par l'intérieur d'une arête alors la face suivante est bien définie. Sinon il faut tourner autour du sommet de sortie pour la trouver. Voir la figure 9.3. La complexité d'insertion d'une



FIGURE 9.3 - La droite épaisse est introduite en partant de son point infini à gauche. Le cercle représente un éclatement du point à l'infini.

droite est proportionnelle à la complexité de sa zone dans l'arrangement déjà construit. Compte tenu du théorème 9.5 on en déduit :

Proposition 9.9 L'arrangement de n droites dans le plan peut être calculé en temps optimal $O(n^2)$.

9.4 Arrangement d'hyperplans

Soit H un ensemble de n hyperplans dans E^d .

Définition 9.10 L'arrangement $\mathcal{A}(H)$ de H est la subdivision de E^d en polyèdres convexes induite par H.

Définition 9.11 un k-flat est un sous-espace affine de dimension k. C'est encore l'enveloppe affine de k + 1 points affinement indépendants. Il est vertical s'il contient la direction verticale (par convention, cela désigne la dernière direction de la base canonique).

Si un hyperplan est non vertical on peut donc parler des demi-espaces *dessous* et *dessus* cet hyperplan.

Étant donné un arrangement d'hyperplans non verticaux $H = \{h_i\}_{1 \le i \le n}$, on associe à chaque point p de E^d un vecteur position dans $\{-, 0, +\}^n$ dont la *i*ème composante indique si p est dessous, sur ou dessus h_i . On note h_i^+ et h_i^- les demi-espaces ouverts respectivement au dessus et au dessus de h_i et on pose $h_i^0 = h_i$. Ainsi, un point p a vecteur position $(\epsilon_1, \ldots, \epsilon_d)$ si et seulement si $p \in \bigcap_i h_i^{\epsilon_i}$.

On définit les faces de $\mathcal{A}(H)$ comme les composantes connexes du complémentaire de $\cup_i h_i$ dans \mathbb{R}^d auxquelles on ajoute récursivement les faces des arrangements induits dans chaque h_i par les intersections de h_i avec les hyperplans de $H \setminus \{h_i\}$. On vérifie que deux points d'une même face de $\mathcal{A}(H)$ ont même vecteur position et que ce vecteur position définit de manière univoque cette face. Notons cependant que tout vecteur position de $\{-, 0, +\}^n$ ne correspond pas nécessairement à une face de l'arrangement. Le vecteur position d'une sous-face d'une face est obtenu en annulant certaines des composantes du vecteur position de la face. Une k-face est une face de dimension k, c'est-à-dire dont l'enveloppe affine est un k-flat. Ainsi, une 0-face est un sommet et une 1-face est une arête. Une d-face est encore appelée une cellule et une (d-1)-face, une facette.

Un arrangement est simple si l'intersection de d hyperplans quelconques de l'arrangement est toujours réduite à un point tandis que l'intersection de d + 1 hyperplans est toujours vide. S'il y a strictement moins de d hyperplans on demande que leur intersection soit un (d-n)-flat. De manière équivalente un arrangement est simple si toute face de codimension k est incluse dans exactement k hyperplans, i.e. a exactement k zéros dans son vecteur position et si les directions (orthogonales) de toute famille d'au plus d hyperplans forment une famille libre. En particulier une sous-face d'une face a exactement un zéro de plus dans son vecteur position.

9.4.1 Dénombrement des faces et incidences

On note $f_k(H)$ le nombre de k-faces de l'arrangement H et $i_k(H)$ le nombre d'incidences entre les k-faces et les (k+1)-faces de H. On note également $f_k^d(n)$ et $i_k^d(n)$ le maximum de ces nombres sur tous les arrangements de n hyperplans dans E^d .

Lemme 9.12 Pour tout arrangement simple H de $n \leq d$ hyperplans dans E^d on a :

$$f_k(H) = \binom{n}{d-k} 2^{n-d+k} \ et \ i_k(H) = 2(d-k)f_k(H)$$

De plus chaque cellule de H est incidente à $\binom{n}{d-k}$ k-faces.

Preuve : Utiliser le vecteur position et le fait que chaque vecteur position correspond effectivement à une face. On peut en effet écrire l'équation du *i*ème hyperplan sous la forme $x_i = 0$ dans une base convenablement choisie.

Lemme 9.13 Pour tout arrangement simple H de n hyperplans dans E^d on a

$$f_k(H) = \sum_{i=0}^k \binom{d-i}{k-i} \binom{n}{d-i} = \binom{n}{d-k} \sum_{i=0}^k \binom{n-d+k}{k-i} et \ i_k(H) = 2(d-k)f_k(H).$$

En particulier, pour d constant on a $f_k(H) = \Theta(n^d)$ et $i_k(H) = \Theta(n^d)$.

Preuve : Pour n < d c'est le lemme 9.12. La preuve précédente pour le nombre d'incidences est encore valide pour n > d. On fait une preuve par récurrence sur d pour $f_k(H)$ lorsque n > d: le lemme est trivialement vrai en dimension 1. On suppose qu'il est vrai en dimension d-1. Soit H comme dans l'énoncé du lemme. Puisque l'arrangement est simple il a $f_0(H) = \binom{n}{d}$ sommets. On choisit une direction d'hyperplan ne contenant aucune des droites vectorielles déterminées par les paires de sommets de l'arrangement de H. On balaye l'arrangement par cet hyperplan de gauche à droite. À chaque fois que l'on balaye un sommet p de l'arrangement une cellule de plus se trouve à gauche de l'hyperplan de balayage. Cette cellule c correspond à l'une des cellules du sous-arrangement simple des dhyperplans de H s'intersectant au point p. Les k-faces supplémentaires balayées en p sont toutes des faces de c et sont donc au nombre de $\binom{d}{d-k}$ par le lemme 9.12. Lorsqu'on balaye le sommet le plus à droite on a donc balayé un nombre $f_0(H) \begin{pmatrix} d \\ d-k \end{pmatrix}$ de k-faces. Il reste à compter les k-faces à l'extrême droite de l'arrangement. Elles sont en bijection avec les (k-1)-faces de l'arrangement simple de dimension d-1 obtenu en coupant $\mathcal{A}(H)$ par un hyperplan de balayage situé à droite du dernier sommet. Par hypothèse de récurrence il y a $\sum_{i=0}^{k-1} {d-1-i \choose k-1-i} {n \choose d-1-i}$ telles faces. D'où :

$$f_k(H) = \binom{n}{d}\binom{d}{d-k} + \sum_{i=1}^k \binom{d-i}{k-i}\binom{n}{d-i} = \sum_{i=0}^k \binom{d-i}{k-i}\binom{n}{d-i}.$$

Théorème 9.14 Les valeurs de $f_k(H)$ et $i_k(H)$ du lemme 9.13 dans le cas simple sont des bornes supérieures dans le cas général, i.e. :

$$f_k^d(n) = \sum_{i=0}^k \binom{d-i}{k-i} \binom{n}{d-i} \ et \ i_k^d(n) = 2(d-k)f_k^d(n).$$

En considérant d constant on a en particulier

$$f_k^d(n) = \Theta(n^d) \ et \ i_k^d(n) = \Theta(n^d).$$

Preuve : On utilise une double récurrence sur n et d. Le théorème est trivialement vrai pour d = 1 et n quelconque et pour n = 2 et d quelconque. On suppose le théorème vrai pour n hyperplans jusqu'à la dimension d - 1 et en toute dimension pour au plus n - 1hyperplans. On se donne une famille $H = \{h_i\}_{1 \le i \le n}$ de n hyperplans dans E^d et on pose $G = H \setminus \{h_n\}$. L'hyperplan h_n intersecte $\mathcal{A}(G)$ en un arrangement $\mathcal{A}(G')$ de dimension d - 1 de n - 1 hyperplans au plus. Une k-face de $\mathcal{A}(H)$ est soit

- une k-face de $\mathcal{A}(G)$,
- une k-face de $\mathcal{A}(G')$ ou
- une k-face de $\mathcal{A}(G)$ coupée en deux par h_n et qui correspond donc à une (k-1)-face de $\mathcal{A}(G')$.

On en déduit

$$f_k(H) \le f_k(G) + f_k(G') + f_{k-1}(G').$$

Par hypothèse de récurrence on en déduit

$$\begin{aligned}
f_k(H) &\leq \sum_{i=0}^k \binom{d-i}{k-i} \binom{n-1}{d-i} + \sum_{i=0}^k \binom{d-1-i}{k-i} \binom{n-1}{d-1} + \sum_{i=0}^{k-1} \binom{d-1-i}{k-1-i} \binom{n-1}{d-1-i} \\
&= \sum_{i=0}^k \binom{d-i}{k-i} \binom{n}{d-i}.
\end{aligned}$$

Une paire de faces incidentes de dimensions respectives k et k + 1 de $\mathcal{A}(H)$ provient soit – d'une telle paire dans $\mathcal{A}(G)$,

- d'une telle paire dans $\mathcal{A}(G')$,
- d'une (k+1)-face de $\mathcal{A}(G)$ coupée en deux par h_n et de la k-face correspondant à leur intersection ou
- d'une paire dans $\mathcal{A}(G)$ coupée en deux par h_n et qui correspond donc a une paire de faces incidentes de dimension k et k-1 dans $\mathcal{A}(G')$.

On en déduit

$$i_k(H) \le i_k(G) + i_k(G') + 2f_k(G') + i_{k-1}(G') \le 2(d-k)(f_k(G) + f_k(G') + f_{k-1}(G')) \le 2(d-k)f_k^d(n)$$

Définition 9.15 Soit H un arrangement de n hyperplans dans E^d et h un (n + 1)-ième hyperplan. La zone de h dans H est l'union des cellules dont l'adhérence intersecte h et des faces de ces cellules. On note $Z_k(h, H)$ l'ensemble des paires (f, c) où f est une face de codimension k de la cellule c de la zone. On note également $z_k(h, H)$ le cardinal de $Z_k(h, H)$ et $z_k(n, d)$ le maximum de cette valeur pour tout h et H.

Théorème 9.16 (Théorème de la zone) Pour d fixé, $z_k(n,d) = O(n^{d-1})$.

Preuve : On suppose que $H \cup \{h\}$ est un arrangement simple et que n > d (le résultat est asymptotique). On considère un hyperplan g de H et on compare la zone de h dans Havec celle de h dans $H \setminus \{g\}$ et de $h \cap g$ dans H/g (= arrangement de dimension d-1 de $H \setminus \{g\}$ intersecté avec g) afin d'établir une formule de récurrence. Plus précisément on s'intéresse au nombre $n_k(g)$ de paires (f, c) de $Z_k(h, H)$ où f n'est pas contenue dans g. Si l'adhérence \bar{c} de c ne rencontre pas g, alors (f, c) est une paire de $Z_k(h, H \setminus \{g\})$. Sinon (f, c) provient d'une paire (ϕ, κ) de $Z_k(h, H \setminus \{g\})$ coupée en deux par g. On note (f', c')la seconde "moitié" de (ϕ, κ) . Si (f', c') est également dans $Z_k(h, H)$ (i.e. si g ne sépare pas f' de h) alors la paire $(\phi \cap g, \kappa \cap g)$ est dans $Z_k(h \cap g, H/g)$ ce qui permet de comptabiliser de manière univoque (f,c) dans $Z_k(h, H \setminus \{g\})$ et (f',c') dans $Z_k(h \cap g, H/g)$. On en déduit :

$$n_k(g) \le z_k(h, H \setminus \{g\}) + z_k(h \cap g, H/g).$$

En sommant cette relation sur tous les hyperplans g de H et en remarquant que chaque paire de $Z_k(h, H)$ est comptée (n - k) fois dans cette somme (par hypothèse de simplicité f est contenue dans exactement k hyperplans), on obtient

$$(n-k)z_k(h,H) \le \sum_{g \in H} (z_k(h,H \setminus \{g\}) + z_k(h \cap g,H/g))$$

D'où

$$z_k(n,d) \le \frac{n}{n-k}(z_k(n-1,d) + z_k(n-1,d-1))$$

Pour résoudre la récurrence, on commence par poser :

$$z_k(n,d) = \binom{n}{k} w_k(n,d) = O(n^k w_k(n,d))$$

pour se ramener à la récurrence :

$$w_k(n,d) \le w_k(n-1,d) + w_k(n-1,d-1)$$

Il s'agit alors de montrer que $w_k(n, d) = O(n^{d-1-k})$. On résout cette récurrence en partant du cas planaire d = 2 déjà traité $(z_k(n, 2) = O(n) \implies w_k(n, 2) = O(n^{1-k}))$. Par hypothèse de récurrence (sur d) on a donc $z_k(n, d-1) = O(n^{d-2})$. On en déduit

$$w_k(n,d) = w_k(n-1,d) + O(n^{d-2-k}) = w_k(d,d) + O(\sum_{i=d+1}^n i^{d-2-k})$$

Si $d-2-k \ge 0$ on obtient $w_k(n,d) = O(n^{d-1-k})$ comme annoncé. Sinon, pour k = d-1 et d, i.e. pour les sommets et les arêtes, on utilise un autre argument : On note f(0,3,c) le nombre de triplets (f_0, f_3, c) où f_0 est un sommet d'une 3-face f_3 d'une cellule c de la zone. On note de même f(2,3,c) le nombre de triplets (f_2, f_3, c) où f_2 est une 2-face d'une 3-face f_3 d'une cellule c de la zone. Comme chaque f_3 est un 3-polyèdre, on a dans une f_3 donnée : $\#f_0 \le 2\#f_2$ (se déduit de la relation d'Euler et du fait que chaque sommet étant simple, chaque 2-face de c est contenue dans (d-2) des 3-faces de c (penser au vecteur position). En sommant sur la zone, on en désuit : $\sum_c f(2,3,c) \le (d-2)f_{d-2}(h,H)$. On en déduit

$$f_d(h, H) \le \sum_c f(0, 3, c) \le 2 \sum_c f(2, 3, c) \le 2(d - 2)z_{d-2}(h, H),$$

d'où $z_d(h, H) = O(n^{d-1}).$

Enfin, par simplicité de l'arrangement, la clôture de chaque cellule est un polytope simple et chaque sommet y est incident à d arêtes. D'où $2z_{d-1}(h, H) = dz_d(h, H)$. On conclut avec l'inégalité précédente.

Dans le cas d'un arrangement non simple on peut montrer en perturbant les hyperplans que la complexité de la zone (les $z_k(n, d)$) est majorée par celle d'une configuration simple d'hyperplans.

On retrouvera la preuve du théorème de la zone selon les mêmes lignes dans [Mul94, Sec. 6.2] et [Mat02, Sec. 6.4].

9.5 Niveaux d'un arrangement

Le niveau k d'une cellule c d'un arrangement $\mathcal{A}(H)$ est le bord de l'ensemble des points séparés de c par au plus k hyperplans de H. Plus généralement on considère un arrangement de demi-espaces (ou hyperplans orientés) et le niveau k de cette famille est le bord de l'ensemble des points qui sont exclus d'au plus k demi-espaces. Un tel arrangement est dit réalisable si l'intersection des demi-espaces est non-vide. C'est le cas ci-dessus où tout les demi-espaces contiennent la cellule c. Le niveau k d'un arrangement réalisable est une sphère ou un disque topologique ou vide. Sinon, un niveau peut être non-connexe. Il est difficile de donner une bonne borne pour la complexité du niveau k d'un arrangement. En dimension 2 et 3, cependant, de telles bornes existent pour les arrangements réalisables. Clarkson et Shor [CS89] montrent qu'en dimension 3 le niveau k a $\Theta(k(n-k))$ sommets dans le cas le pire. Des bornes supérieures ont été obtenues dans le cas général mais pour l'accumulation des k premiers niveaux. Ainsi le nombre de sommets de l'union des k premiers niveaux est un $O(n^{\lfloor d/2 \rfloor}(k+1)^{\lceil d/2 \rceil})$.

On déduit de l'étude de la taille des niveaux des informations sur la complexité des diagrammes de Voronoi d'ordres supérieurs. Voir la section 8.2.3 à ce sujet.

On pourra consulter sur ce sujet le chapitre d'Uli Wagner dans [GPP08].

9.6 Dualité

À un point $x = (\mathbf{x}, x_d) \in E^d$ on associe l'hyperplan $\mathcal{D}(x)$ d'équation $y_d = 2\mathbf{x} \cdot \mathbf{y} - x_d$. On définit ainsi une bijection de E^d dans l'espace des hyperplans non verticaux de E^d . On utilise la même notation pour l'inverse de \mathcal{D} .

On note U le paraboloide de révolution d'équation $x_d = \mathbf{x}^2$. Ce qui s'écrit encore $(x_d + \frac{1}{4})^2 = (x_d - \frac{1}{4})^2 + \mathbf{x}^2$. U a donc le point $(\mathbf{0}, \frac{1}{4})$ pour foyer et l'hyperplan $\{x_d = -\frac{1}{4}\}$ pour directrice. On note $T_U(x)$ l'hyperplan tangent en un point x de U.

Propriétés géométriques :

- deux points, p et q sont sur une même verticale si et seulement si $\mathcal{D}(p)$ et $\mathcal{D}(q)$ sont parallèles,
- plus spécifiquement, si τ est une translation verticale alors $\mathcal{D}(\tau . x) = \tau^{-1} . \mathcal{D}(x)$,
- Si $x \in U$ alors $\mathcal{D}(x) = T_U(x)$, ce qui avec la propriété précédente montre que pour x quelconque $\mathcal{D}(x)$ s'obtient comme l'hyperplan symétrique par rapport à $T_U(x')$ de l'hyperplan parallèle à $T_U(x')$ et passant par x, où x' est le point de U à la verticale de x,
- pour x en dessous de U, l'hyperplan tangent à U en un point de $\mathcal{D}(x) \cap U$ passe par x,

• pour x en dessous de $U, \mathcal{D}(x) \cap U$ se projette verticalement sur l'hyperplan horizontal en une (d-1)-sphère dont le centre est la projection de x et le carré du rayon la distance de x au point de U situé à la verticale de x.

Propriétés de dualité :

- $\mathcal{D} \circ \mathcal{D}$ est l'identité (par définition),
- x est dessous (resp. sur, resp. dessus) l'hyperplan non vertical h si et seulement si $\mathcal{D}(h)$ est dessous (resp. sur, resp. dessus) $\mathcal{D}(x)$,
- Soit h un hyperplan non vertical et $Q = \{x_1, \ldots, x_d\}$ un ensemble de d points de h affinement indépendants alors $\mathcal{D}(Q)$ définit un arrangement simple i.e. a un unique point d'intersection.

Preuve : Un point appartient à l'intersection des hyperplans de $\mathcal{D}(Q)$ si et seulement si son hyperplan dual contient Q, i.e. si et seulement si son dual est h.

• Soit h un hyperplan non vertical et Q un ensemble quelconque de points dans h. L'enveloppe affine de Q est de dimension k si et seulement si $\mathcal{D}(Q)$ s'intersecte en un sous-espace de dimension d - k - 1.

Preuve : Soit Q' une famille de k + 1 points de Q affinements indépendants. On les complètes en une famille indépendantes de d points de h. D'après le point précédent les hyperplans duaux de cette famille forment un arrangement simple et c'est donc le cas pour la sous famille $\mathcal{D}(Q')$. Leur intersection est donc un sous-espace affine de dimension d - k - 1. Les autres points de Q étant affinement liés à Q' l'équation de leurs hyperplans duaux sont des combinaisons linéaires de celles de Q' et contiennent donc leur intersection.

• Comme corollaire du point précédent on a :

Soit Q un ensemble de n points dans E^d et $H = \mathcal{D}(Q)$. Soit h un hyperplan non vertical. L'enveloppe affine des points de Q situés sur h est de dimension k si et seulement si $\mathcal{D}(h)$ est contenu dans l'intérieur d'une (d-1-k)-face de $\mathcal{A}(H)$.

Preuve : les hyperplans duaux des points de Q sur h définissent un sous-espace affine de dimension d-k-1. Or les autres points de Q n'étant pas sur h, leur dual ne contient pas $\mathcal{D}(h)$ et évitent une petite boule centrée en ce point. L'intersection de cette boule avec le sous-espace s'étend en une unique (d-1-k)-face de $\mathcal{A}(H)$.

• Puisque la dualité préserve les positions relatives (dessous/dessus) entre points et hyperplans non verticaux, un hyperplan dual d'un point p intérieur à une cellule de $\mathcal{A}(H)$ sépare les points de Q en deux ensembles indépendants du point p choisi dans la cellule. La réciproque est cependant fausse car il peut y avoir des cellules dont les vecteurs positions sont "opposés". Cela ne peut arriver que pour des paires de cellules non bornées.

Exercice 9.17 Donner un algorithme de complexité $O(n \log n)$ pour trouver une boîte rectangulaire parallèle aux axes contenant tous les sommets d'un arrangement de n droites.

Exercice 9.18 Soient n points du plan. Donner un algorithme de complexité quadratique pour trouver une droite passant par le plus possible de ces points.

Exercice 9.19 Soient n segments du plan. Trouver un algorithme quadratique pour décider si ces segments peuvent être tous percés par une même droite.

Exercice 9.20 Étant donnés n points rouges et m points bleus en position générale dans le plan, trouver une droite qui "coupe" simultanément les points rouges et les points bleus en deux, c'est-à-dire telle que chaque demi-plan ouvert délimité par cette droite contient au plus n/2 points rouges et m/2 points bleus. Quelle est la complexité de votre algorithme?

L'exercice précédent est une version discrète du célèbre théorème du Sandwich. De multiples extensions ont été considérées dans la littérature, que ce soit en dimension supérieure ou avec des ensembles de points comportant plus de couleurs. Matoušek [LMS94] résoud l'exercice précédent en temps linéaire!

Références

- Computational Geometry. Algorithms and applications. de Berg, van Kreveld, Overmars and Schwarzkopf. Springer 1997.

- Algorithms in Combinatorial Geometry. H. Edelsbrunner. Springer Verlag Monographs in TCS. 1987.

- Algorithmic Geometry. J-D Boissonat et M. Yvinec. Cambridge University Press, 1998 (version française : Ediscience international, 1995).

Pour les arrangements et les balayages voir également l'article d'Edelsbrunner et Guibas [EG89]. On y montre en particulier qu'on peut balayer un arrangement de n droites dans le plan en temps $O(n^2)$. Un balayage simple selon les abscisses des intersections auraient une complexité en $O(n^2 \log n)$. L'idée est ici de remplacer la droite verticale de balayage par une front x-monotone de balayage. Ce front est composé d'un segment par droite dans un ordre "vertical". Chaque fois que l'on balaye un point d'intersection dans l'arrangement on intervertit l'ordre de deux droites. Le problème est de savoir quelles sont les paires de droites qui peuvent s'intersecter dans la suite du balayage (i.e. dont les segments du front ont leur point droit en commun). Pour avoir cette information en temps constant on maintient deux structures annexes : les arbres d'horizons supérieur et inférieur. La mise à jour de ces arbres après chaque balayage utilise un parcours de ligne d'horizon. Voir aussi les nombreuses applications dans le même article.

Chapitre 10

Localisation

Le problème de la localisation consiste, étant donnée une collection fixée de n objets dans \mathbb{R}^d , à trouver (les) l'objet(s) de la collection contenant un point de requête quelconque. C'est d'une certaine manière le problème dual de la recherche multidimensionnelle du chapitre 11. Pour d = 1, cela revient à trouver les intervalles (ce sont les objets de la collection) contenant le point de requête. Les structures d'arbre d'intervalles et d'arbre de segments permettent de répondre efficacement à ce type de requête mono-dimensionnelle en temps $O(\log n + k)$ où k est le nombre d'intervalles contenant le point de requête. On pourra consulter [dBCvKO08, chap. 10] pour des détails sur ces structures. Dans ce chapitre on se restreindra au cas d = 2 où les objets forment une subdivision d'une partie du plan, c'est-à-dire que les régions s'identifient aux faces de la subdivision. Ce cas est en particulier utile pour les interactions avec les systèmes d'information géographique.

Les deux sections suivantes sont essentiellement des traductions du chapitre 6 de de Berg et al. [dBCvKO08].

10.1 Localisation par découpe en bandes verticales

On suppose ici que l'ensemble des régions est donné sous forme d'une carte planaire combinatoire (cf. section 9.2) de taille n, i.e. possédant n arêtes. Notons que si la carte est connexe, on peut marquer, lors d'un simple parcours en temps linéaire, chaque demiarête de la carte avec l'indice de la face située à sa gauche. La localisation consiste alors à reporter l'indice de la face de la carte combinatoire contenant le point de requête. La méthode qui suit est due à Dobkin et Lipton [DL76] et repose sur une découpe de la carte en bandes verticales. On commence par tracer par chaque sommet de la carte une droite verticale comme sur la figure ci-après. Ces droites verticales découpent le plan en bandes verticales. À l'intérieur d'une bande les régions traversées sont ordonnées verticalement. Ainsi pour localiser un point p de requête il suffit d'effectuer deux recherches mono-dimensionnelles : une recherche *horizontale* pour déterminer la bande contenant p, suivie d'une recherche *verticale* permettant d'identifier la région, ou face, de la carte contenant p. Chaque recherche s'effectuant sur un ensemble de taille O(n), la localisation peut s'effectuer en temps $O(\log n)$ à l'aide d'une structure de dictionnaire classique



(déterministe) ou randomisée (cf. chapitre 5).

Plus précisément, pour construire la structure de recherche, on commence par ranger les sommets de la carte dans un dictionnaire en utilisant l'ordre lexicographique sur leurs coordonnées pour les comparaisons. On effectue ensuite un balayage des sommets de la carte de gauche à droite. Dit autrement, on parcourt les sommets dans l'ordre lexicographique. Pour chaque sommet q balayé on construit une structure de recherche V[q] correspondant à la bande verticale située à droite de q. Cette structure de recherche est elle même un dictionnaire sur les arêtes, ordonnées verticalement, qui traversent V[q]. Clairement V[q] peut être obtenue à partir du dictionnaire de la bande précédente en ajoutant les arêtes incidentes à q et tournées vers la droite et en supprimant les arêtes incidentes à q et tournées vers la gauche.

Dans le pseudo-code suivant H désigne le dictionnaire des sommets de la carte pour l'ordre lexicographique et V[q] désigne un dictionnaire sur les arêtes traversant la bande verticale à droite du sommet q.

Construction de la structure de recherche

On suppose les sommets $p_0, ..., p_m$ de la carte ordonnés de gauche à droite.

Initialiser un dictionnaire V_{aux} à vide **Pour** i := 0 à m **faire** Insérer p_i dans le dictionnaire H $V[p_i] := V_{aux}$ **Pour** chaque demi-arête a d'origine p_i **faire Si** a est orientée vers la gauche **Alors** Supprimer la demi-arête opposée de a dans $V[p_i]$ **Sinon** Insérer a dans $V[p_i]$ $V_{aux} := V[p_i]$

Localisation d'un point p

Rechercher dans H le sommet q de la carte juste à gauche de p
Si p == q Alors renvoyer cette information
Sinon Si p == NILL (p est à l'extrême gauche de la carte) Alors retourner la face la plus à gauche de la carte
Sinon rechercher la demi-arête a (orientée vers la droite) sous p dans V[q]
Si a == NILL Alors retourner la face la plus basse de la carte

Sinon Si p est sur a Alors retourner cette information Sinon retourner la région à gauche de a

Le temps de construction de la structure de recherche, hormis la recopie des dictionnaires verticaux est $O(n \log n)$: chaque demi-arête est insérée et supprimée une fois dans l'ensemble des bandes au cours du balayage. Cela dit on recopie O(n) fois la structure V_{aux} . Comme cette structure peut avoir une taille O(n) ceci donne finalement une construction en temps et espace $O(n^2)$. On peut cependant améliorer ces résultats en utilisant des dictionnaires (arbres) "persistants".

Dans un dictionnaire persistant on ordonne dans le temps les opérations d'insertion et de suppression et on peut faire une requête du type : rechercher un élément dans le dictionnaire tel qu'il se trouvait au moment de la kième opération. On peut obtenir sur n éléments et m opérations un temps d'insertion/suppression en $O(\log n)$, un temps de recherche en $O(\log m)$ et une taille amortie en O(n). Appliqué au présent problème cela donne

Théorème 10.1 Soit S un ensemble de n segments formant une carte. L'algorithme exposé calcule en temps $O(n \log n)$ une structure de recherche associée à S de taille O(n). De plus pour tout point p du plan le temps de localisation dans cette structure est $O(\log n)$.

Référence :

- Éléments d'algorithmique. D. Beauquier, J. Berstel et P. Chrétienne. MASSON, 1992.

10.2 Localisation par décomposition trapézoïdale

On peut obtenir un temps de recherche équivalent à l'approche par découpe verticale avec une structure de recherche de taille linéaire sans utiliser de structure de données persistante. Pour cela on trace comme précédemment une verticale par chaque sommet de la subdivision mais on stoppe cette verticale dès que l'on rencontre un segment de la carte. Afin de simplifier l'exposé, on commence par entourer la carte d'une grande boîte rectangulaire et on ne s'intéresse qu'à l'intérieur de cette boîte. En particulier chaque verticale est bornée (cf. figure suivante). Chaque face de la subdivision ainsi obtenue est



convexe : les angles en chaque sommet d'une telle face (sommet d'arête ou intersection d'une verticale et d'une arête) sont plus petits que π . Chaque face a donc au plus deux

côtés verticaux et donc au plus deux non-verticaux (un sommet commun à deux segments non-verticaux est l'origine d'une verticale) et donc exactement deux non-verticaux puisque bornée. Dit autrement chaque face de la subdivision est géométriquement un *trapèze* possédant quatre côtés dont deux verticaux ou trois côtés dont un vertical. La subdivision ainsi obtenue porte ainsi le nom de *carte des trapèzes*.

Un trapèze est entièrement défini par les deux segments qui le bordent en haut et en bas et par les deux sommets qui ont permis de construire ses parois verticales gauche et droite. On note *segh*, *segb*, *ptg* et *ptd* respectivement ces quatre attributs. La figure suivante montre deux exemples de trapèzes avec les attributs correspondants.



On distingue cinq configurations pour le côté vertical gauche d'un trapèze comme illustré figure 10.2 :

- 1. il est réduit à un point qui est l'extrémité gauche commune aux segments supérieur et inférieur.
- 2. c'est l'extension verticale du sommet gauche du segment inférieur qui aboutit sur le segment supérieur.
- 3. même chose en échangeant supérieur et inférieur.
- 4. c'est l'extension verticale de l'extrémité droite d'un autre segment qui le coupe donc en deux.
- 5. c'est un côté de la boîte englobante.

Il existe de même cinq configurations distinctes pour le côté vertical droit d'un trapèze.

Lemme 10.2 La carte des trapèzes a un nombre linéaire de faces, arêtes et sommets en fonction du nombre n de segments de la carte planaire d'origine.



FIGURE 10.1 – Le côté vertical gauche d'un trapèze peut se trouver dans cinq configurations distinctes.
Preuve : Chaque sommet d'origine (au plus 2n) donne lieu à au plus trois sommets : luimême + 2 sommets pour les extensions inférieures et supérieures des verticales. Avec les 4 sommets de la boîte englobante cela donne au plus 6n+4 sommets en tout. De plus chaque sommet d'origine donne lieu à 4 nouveaux segments : ses deux extensions verticales + 2 segments obtenus par subdivision des segments rencontrés par ses extensions verticales. La carte des trapèzes possèdent donc au plus 4.2n+n+4 = 9n+4 segments en tenant compte de la boîte englobante. La relation d'Euler permet finalement de borner trivialement le nombre de faces par le nombre d'arêtes plus un.

Exercice 10.3 Montrer qu'on peut borner le nombre de faces par 3n + 1 en associant judicieusement chaque face à l'un des segments intersectant le bord de cette face.

On associe à la carte des trapèzes une structure \mathcal{T} la représentant, mais plus simple que la représentation classique en demi-arêtes, et une structure \mathcal{R} de recherche pour localiser un point. Nous supposerons par la suite les sommets de la carte en position générale de sorte que chaque droite verticale passe par au plus un sommet. De cette manière chaque trapèze possède au plus quatre voisins adjacents à ses parois verticales. On note T_{gh} , T_{gb} , T_{dh} et T_{db} ces voisins comme sur la figure ci-dessous.



• La structure \mathcal{T} se présente sous forme d'une liste d'enregistrements correspondant à chacun des trapèzes. Chaque enregistrement contient les pointeurs sur les attributs *segh*, *segb*, *ptg*, *ptd*, T_{gh} , T_{gb} , T_{dh} et T_{db} précédemment définies.

• La structure \mathcal{R} est un graphe orienté sans cycle possédant une unique racine et exactement une feuille par trapèze de \mathcal{T} . Chaque trapèze de \mathcal{T} pointe sur sa feuille correspondante dans \mathcal{R} et réciproquement ¹. Les noeuds internes de \mathcal{R} ont tous deux enfants et pointent soit sur un sommet soit sur un segment de la subdivision originale. Chaque noeud interne ν est la racine dans \mathcal{R} d'un sous-graphe orienté sans cycle qui est une structure de recherche pour l'intersection d'une certaine zone Z_{ν} avec \mathcal{T} . Cette zone est définie récursivement comme suit. La zone de la racine de \mathcal{R} est l'intérieur de la boîte englobante. Si ν pointe sur un sommet alors la zone de sa fille gauche (droite) est la partie de Z_{ν} à gauche (droite) de ce sommet. Si ν pointe sur un segment alors la zone de sa fille gauche (droite) est la partie de Z_{ν} au dessus (au dessous) de ce segment.

Pour effectuer une recherche d'un point p dans la structure de recherche \mathcal{R} , on part de la racine et on descend progressivement jusqu'à une feuille qui pointe sur le trapèze

^{1.} On peut également construire la structure \mathcal{T} directement sur les feuilles de \mathcal{R}

contenant p. À chaque noeud pointant sur un sommet on s'oriente vers la fille gauche ou droite selon que p est à gauche ou à droite de ce sommet. De même, à chaque noeud pointant sur un segment on s'oriente vers la fille gauche ou droite selon que p est au dessus ou au dessous de ce segment. La feuille atteinte correspond au trapèze contenant p. Ce trapèze peut lui-même être indéxé par la face le contenant dans la carte d'origine (via la face à gauche d'une demi-arête de son segment inférieur) fournissant ainsi le résultat cherché.

 \mathcal{T} et \mathcal{R} sont construits de manière incrémentale randomisée en insérant successivement chaque segment s_i de la subdivision d'origine, pris dans un ordre aléatoire, dans les structures \mathcal{T}_{i-1} et \mathcal{R}_{i-1} associées aux i-1 premiers segments insérés.

L'algorithme de construction est le suivant :

Données : Une carte planaire formée de n segments s_1, \ldots, s_n énumérés dans un ordre aléatoire. **Sorties** : Les structures \mathcal{T} et \mathcal{R} associées.

1. Déterminer une boite englobante contenant les n segments (strictement) et initialiser \mathcal{T}_0 et \mathcal{R}_0

2. Pour $i := 1 \ge n$ faire

3. Localiser grâce à \mathcal{R}_{i-1} le trapèze T_1 de \mathcal{T}_{i-1} contenant l'extrémité gauche de s_i .

4. En déduire pas simple parcours dans \mathcal{T}_{i-1} les trapèzes T_1, T_2, \ldots, T_k traversés par s_i .

5. Remplacer chaque trapèze T_1, T_2, \ldots, T_k de \mathcal{T}_{i-1} par sa subdivision, induite par s_i ,

et chacune composée de 2,3 ou 4 trapèzes.

6. Faire de même dans \mathcal{R}_{i-1} en remplaçant cette fois chaque trapèze feuille par un sous-arbre de recherche élémentaire sur sa subdivision.

7. Fusionner les trapèzes des subdivisions pour rétablir la décomposition trapézoïdale induite par s_1, \ldots, s_i .

finpour

Pour l'étape 6, il y a essentiellement 3 types de sous-arbres élémentaires possibles selon qu'un trapèze est découpé en 2,3 ou 4 morceaux. Les noeuds internes des ces sous-arbres correspondent au segment s_i ou à ses extrémités. La figure suivante résume les étapes 6 et 7 sur deux exemples.

La structure de recherche \mathcal{R} dépend évidemment de l'ordre d'insertion des n segments de la carte d'origine. On considère \mathcal{R} comme une variable aléatoire sur l'espace des permutations (sur l'insertion) des segments muni de la loi uniforme.

Lemme 10.4 Soit p un point fixé du plan. Le temps moyen de localisation de p dans \mathcal{R} est un $O(\log n)$.

Preuve : Le temps de localisation de p est clairement proportionnel à la longueur de son chemin de recherche l(p) dans \mathcal{R} .

Remarque : à chaque insertion d'un segment dans la construction incrémentale la hauteur de \mathcal{R} augmente d'au plus 3. Un chemin de recherche a donc une longueur bornée par 3n.

On note X_i la variable aléatoire dénombrant les arcs de l(p) apparus dans \mathcal{R} lors de l'insertion de s_i , i.e. les arcs de l(p) présents dans \mathcal{R}_i mais pas dans \mathcal{R}_{i-1} . En notant Y_i



la variable aléatoire valant 1 si ce chemin est rallongé à l'étape i et 0 sinon, on a suivant la remarque précédente

$$|l(p)| = \sum_{i} X_i \le 3 \sum_{i} Y_i.$$

D'où en prenant les espérances

$$E(|l(p)|) \le 3\sum_{i} E(Y_i) = 3\sum_{i} P(Y_i = 1).$$

Or l(p) est rallongé à l'étape *i* si et seulement si le segment s_i intersecte le trapèze contenant *p* dans la carte \mathcal{T}_{i-1} c.-à-d. si et seulement si le trapèze Δ_i contenant *p* dans la carte \mathcal{T}_i n'existe pas dans la carte \mathcal{T}_{i-1} . Mais $\Delta_i \notin \mathcal{T}_{i-1}$ si et seulement si l'un des quatre attributs qui le définissent, $(segh(\Delta_i), segb(\Delta_i), ptg(\Delta_i) \text{ ou } ptd(\Delta_i))$ n'est pas présent dans \mathcal{T}_{i-1} . Selon le principe de l'analyse arrière fixons le sous-ensemble S_i des *i* premiers segments insérés. Notons que \mathcal{T}_i – et donc Δ_i – ne dépend que de S_i et non de l'ordre d'insertion des *i* premiers segments. On a alors

$$P(Y_{i} = 1 \mid S_{i}) \leq P(segh(\Delta_{i}) \notin \mathcal{T}_{i-1} \mid S_{i}) + P(segb(\Delta_{i}) \notin \mathcal{T}_{i-1} \mid S_{i}) + P(ptg(\Delta_{i}) \notin \mathcal{T}_{i-1} \mid S_{i}) + P(ptd(\Delta_{i}) \notin \mathcal{T}_{i-1} \mid S_{i})$$

Puisque chaque segment de S_i a une probabilité 1/i d'être inséré le dernier, on a $P(segh(\Delta_i) \notin \mathcal{T}_{i-1} \mid S_i) = P(segh(\Delta_i) = s_i \mid S_i) = 1/i$ et de même $P(segb(\Delta_i) \notin \mathcal{T}_{i-1} \mid S_i) = 1/i$. Si plusieurs segments de S_i ont $ptg(\Delta_i)$ pour extrémité alors $P(ptg(\Delta_i) \notin \mathcal{T}_{i-1} \mid S_i)$ est nul car $ptg(\Delta_i)$ est alors à coup sûr dans \mathcal{T}_{i-1} . Sinon, en notant s l'unique segment de S_i d'extrémité $ptg(\Delta_i)$, on a $P(ptg(\Delta_i) \notin \mathcal{T}_{i-1} \mid S_i) = P(s = s_i \mid S_i) = 1/i$. D'où dans tous les cas $P(ptg(\Delta_i) \notin \mathcal{T}_{i-1} \mid S_i) \leq 1/i$. En résumé,

$$P(Y_i = 1 \mid S_i) \le 4/i$$

D'où inconditionnellement $P(Y_i = 1) \le 4/i$. On en déduit $E(|l(p)|) \le 12H_n = O(\log n)$. \Box

Lemme 10.5 La taille moyenne de la structure \mathcal{R} de recherche est linéaire.

Preuve : On vérifie que si k_i est le nombre de nouveaux trapèzes créés à l'étape i, alors le nombre de noeuds internes ajoutés à \mathcal{R}_{i-1} est $k_i - 1$: soient T_1, \ldots, T_k les trapèzes intersectés par s_i . Si T_j est subdivisé en $n(T_j)$ morceaux par l'introduction de s_i alors son sous-arbre élémentaire a $n(T_j) - 1$ noeuds internes. Soit un total de $N_i = \sum_j n(T_j) - k$ noeuds internes créés par l'introduction de s_i . Mais le nombre de nouveaux trapèzes est $k_i = \sum_j n(T_j) - (k-1) = N_i + 1$ car il faut effectuer k-1 fusions (le nombre de trapèzes intersectés moins 1) parmi les $\sum_j n(T_j)$ morceaux précédemment obtenus pour créer les nouveaux trapèzes.

Puisque la carte des trapèzes a une taille linéaire (lemme 10.2) \mathcal{R} possède O(n) feuilles plus $\sum_i (k_i - 1)$ noeuds internes. D'où $E(|\mathcal{R}|) = O(n) + \sum_i E(k_i)$. On utilise à nouveau une analyse arrière : pour S_i fixé, un trapèze de \mathcal{T}_i a une probabilité au plus 4/i d'avoir été créé par l'insertion de s_i . Le nombre moyen $E(k_i)$ de trapèzes créés à l'étape i, qui est la somme de cette probabilité pour tous les trapèzes de \mathcal{T}_i , est donc majoré par O(i)4/i = O(1) puisqu'il y a O(i) trapèzes sans \mathcal{T}_i .

En ce qui concerne le temps moyen de construction on remarque que le coût d'insertion de s_i est le temps de localisation de son extrémité gauche plus $O(k_i)$ pour parcourir les trapèzes traversés et mettre à jour \mathcal{R}_{i-1} et \mathcal{T}_{i-1} . Le temps moyen de localisation est $O(\log i)$ par le lemme 10.4 et $E(k_i) = O(1)$ d'après ce qui précède. On en déduit un coût moyen total de $O(n \log n)$.

En résumé,

Théorème 10.6 Soit S un ensemble de n segments formant une carte planaire. L'algorithme exposé calcule en temps moyen $O(n \log n)$ une structure de recherche pour la carte des trapèzes associée à S de taille moyenne O(n). De plus pour tout point p du plan le temps moyen de localisation dans cette structure est $O(\log n)$.

Notons que les moyennes sont ici prises relativement aux permutations sur les segments et que p est fixé une fois pour toute. En particulier, ce résultat ne dit rien sur le temps maximal moyen relativement à l'ensemble des points requêtes. Autrement dit il se pourrait que pour la plupart des permutations des segments il existe un point, dépendant de la permutation considérée, ayant un "mauvais" temps de localisation. En fait le temps maximal moyen est lui-même majoré par $O(\log n)$. Pour le voir on commence par remarquer que deux points appartenant à la même cellule de l'arrangement des droites supports des n segments et des 2n verticales aux sommets des segments partagent un même chemin de recherche dans \mathcal{R} quelle que soit la permutation considérée. Pour chaque permutation, il y a donc $O(n^2)$ chemins de recherche distincts possibles ² qui ont bien sûr chacun une longueur moyenne en $O(\log n)$. Si on montre que pour chaque chemin cette moyenne n'est dépassée que pour un sous-ensemble suffisamment petit de permutations alors ce sera encore le cas pour le maximum des longueurs de ces $O(n^2)$ chemins. C'est l'objet de ce qui suit.

^{2.} Voir le théorème 9.3 sur les arrangements de droites.

Pour faire une analyse plus fine de la queue de distribution on utilise la technique de Chernoff. Le problème est ici que les variables aléatoires Y_i introduites plus haut ne sont pas indépendantes (exercice : le vérifier pour S formé des cotés d'un triangle et un point interne à ce triangle). On utilise une définition modifiée qui les rend indépendantes. Pour cela on considère le diagramme de Hasse du treillis (booléen) des parties de S pour l'inclusion. Ce treillis a $\binom{n}{i}$ noeuds de *niveau* i ayant chacun i arcs entrants et n-iarcs sortants. Les permutations de S sont en bijection avec les chaînes maximales de ce treillis (joignant vide à plein) et chaque arc du treillis correspond à un segment de S (le segment ajouté). On fixe un point p et on marque un arc si la suppression du segment correspondant modifie le trapèze contenant p dans la carte associée à la tête de cet arc. On a vu que chaque noeud a au plus 4 arcs entrants marqués. L'astuce est de compléter systématiquement à 4 le nombre d'arcs entrants marqués (pour les noeuds de niveaux 1, 2, 3 on marque tous les arcs entrants). Le point p étant fixé, on note X_i la variable aléatoire définit sur les chaînes maximales du diagramme (donc les permutations) valant 1 si l'arc entre les niveaux i-1 et i est marqué et 0 sinon. Ainsi la longueur du chemin de recherche de p est majorée par $3\sum_i X_i$.

Lemme 10.7 Les X_i sont mutuellement indépendantes.

Preuve : Commençons par remarquer que si A est une condition portant sur des niveaux supérieurs ou égaux à i alors $P(X_i = \epsilon \mid A)$ ne dépend pas de A et vaut donc $P(X_i = \epsilon)$. Par exemple, si a est une arête du treillis entre les niveaux i et i + 1, la probabilité que $X_i = 1$ restreinte aux permutations contenant a vaut 4/i si $i \ge 4$.

On raisonne ensuite par récurrence sur le nombre k de v.a. prises en compte. On suppose que les k premières variables aléatoires X_1, \ldots, X_k sont mutuellement indépendantes et plus précisément que pour toute condition A_k portant sur des niveaux $\geq k$, les v.a. $X_1|A_k, \ldots, X_k|A_k$ le sont avec des probabilités uniformes par rapport à A_k . Dit autrement, pour tout $\epsilon_1, \ldots, \epsilon_k \in \{0, 1\}^k$, on a

$$P(X_1 = \epsilon_1 \wedge \dots \wedge X_k = \epsilon_k \mid A_k) = \prod_{j=1}^k P(X_j = \epsilon_j)$$
(10.1)

Notons *B* l'événement $(X_1 = \epsilon_1 \land ... \land X_k = \epsilon_k)$. On a alors (cf. exercice 1.20) pour toute condition A_{k+1} portant sur des niveaux $\geq k + 1$ et pour tout $\epsilon_{k+1} \in \{0, 1\}$:

$$P(B \land X_{k+1} = \epsilon_{k+1} \mid A_{k+1}) = P(B \mid X_{k+1} = \epsilon_{k+1} \land A_{k+1})P(X_{k+1} = \epsilon_{k+1} \mid A_{k+1})$$

Ce qui vaut encore $\prod_{j=1}^{k+1} P(X_j = \epsilon_j)$ d'après l'équation (10.1) et la remarque initiale. \Box

On peut maintenant appliquer la technique de Chernoff (cf. section 1.7.4) à la variable $X_p = \sum_i X_i$. Pour cela on regarde la probabilité que X_p dépasse $\lambda \ln(n+1)$. En utilisant l'inégalité de Markov on a

$$\forall t > 0, \ P(X_p \ge \lambda \ln(n+1)) = P((e^{tX_p}) > (n+1)^{t\lambda}) \le E(e^{tX_p})/(n+1)^{t\lambda}$$

Or de part l'indépendance des X_i on a

$$E(e^{tX_p}) = \prod_i E(e^{tX_i})$$

Avec $t = \ln(5/4)$ on calcule pour $i \ge 4$, $E(e^{tX_i}) = (5/4)4/i + 1(1-4/i) = (i+1)/i$ et $E(e^{tX_i}) = 5/4$ pour i = 1, 2, 3. D'où $E(e^{tX_p}) \le n+1$. Finalement on trouve

$$P(X_p \ge \lambda \ln(n+1)) \le 1/(n+1)^{\lambda \ln(5/4)-1}$$

On a vu plus haut que X_p ne dépend que de la cellule d'un certain arrangement de taille $O(n^2)$. La probabilité que l'un des X_p définis pour chacune des cellules de l'arrangement dépasse $\lambda \ln(n+1)$ est donc majorée par $O(1/(n+1)^{\ln(5/4)\lambda-3})$. Comme X_p est uniformément borné par O(n) on en déduit que

$$E(\max_{p} X_{p}) = \sum_{k} kP(\max_{p} X_{p} = k)$$

$$\leq \lambda \ln(n+1)P(\max_{p} X_{p} < \lambda \ln(n+1)) + O(n)P(\max_{p} X_{p} \ge \lambda \ln(n+1))$$

$$\leq \lambda \ln(n+1) + O(1/(n+1)^{\lambda \ln(5/4)-4})$$

Finalement, en choisissant λ tel que $\lambda \ln(5/4) \ge 4$, on obtient

Lemme 10.8 L'espérance du temps maximal de recherche est $O(\log n)$.

On montrerait de même que la probabilité que la taille (resp. le temps de construction) ne dépasse pas λn (resp. $\lambda n \log n$) est aussi proche de 1 que l'on veut si on choisit λ assez grand. Du coup la probabilité P que le temps maximal de recherche, la taille, et le temps de construction soient simultanément comme désirés est non nulle (et en fait P est aussi proche de 1 que l'on veut si on fait croître λ). On en déduit

Théorème 10.9 Soit S l'ensemble des n segments d'une carte planaire. On peut construire en temps moyen $O(n \log n)$ une structure de recherche pour la carte des trapèzes associée à S de taille O(n) et de temps de requête $O(\log n)$ dans le cas le pire.

Pour cela on déroule l'algorithme précédent pour une permutation aléatoire jusqu'à s'apercevoir que la taille ou la profondeur de \mathcal{R} dépasse λn ou $\lambda \log n$. Si c'est le cas on recommence avec une autre permutation aléatoire jusqu'à obtenir les taille et profondeur désirées. La moyenne du nombre d'essais à effectuer est 1/P et le temps moyen de construction est donc $O(n \log n)/P = O(n \log n)$.

Référence :

- Computational Geometry. Algorithms and applications. de Berg, van Kreveld, Overmars and Schwarzkopf. Springer 1997. pf. Springer 1997.

10.3 Localisation dans une triangulation

Une autre approche, proposée par Kirkpatrick [Kir83], consiste à partir d'une triangulation. Cette dernière peut être obtenue pour une carte planaire de taille n en temps $O(n \log n)$ suivant les algorithmes présentés au chapitre 3. La méthode de localisation de Kirkpatrick repose sur une représentation hiérarchique de la triangulation obtenue par simplifications progressives. Pour passer d'une triangulation à une triangulation simplifiée on supprime un ensemble de sommets indépendants (deux à deux non-adjacents) et de degré borné puis on retriangule si nécessaire les "trous" laissés par les sommets supprimés. Pour que cet algorithme soit efficace on cherche à construire des ensembles de sommets indépendants, et de degrés bornés, les plus grands possibles. La borne sur les degrés assure que chaque triangle à un niveau de la hiérarchie intersecte un nombre borné de triangles du niveau suivant.

Dans ce qui suit le terme *triangulation* désigne une triangulation rectiligne du plan dont la face externe est elle-même un triangle (c'est donc combinatoirement une triangulation de la sphère). On peut toujours se ramener à une telle triangulation à partir d'une triangulation quelconque d'un domaine convexe en ajoutant trois sommets "à l'infini" et en joignant convenablement les sommets du bord du domaine à ces trois sommets. Les sommets d'une triangulation distincts des 3 sommets de la face externe sont dits *internes*.

Lemme 10.10 Soit \mathcal{T} une triangulation ayant n sommets internes et un entier $d \geq 6$. On peut sélectionner en temps O(n) un nombre au moins αn de sommets internes, indépendants et de degrés au plus d avec $\alpha = \frac{d-5}{(d+1)(d-2)}$.

Preuve : On considère l'algorithme glouton suivant : parcourir les sommets internes de \mathcal{T} et sélectionner le sommet courant du parcours si son degré est au plus d et si aucun de ses voisins n'est sélectionné. L'algorithme glouton sélectionne clairement un ensemble de sommets internes, indépendants et de degrés au plus d en temps linéaire. Reste à minorer sa taille n_s .

Soit x et y les nombres de sommets internes de \mathcal{T} respectivement de degré au plus d et au moins d+1. En particulier, x+y=n. On note a le nombre d'arêtes de \mathcal{T} . Par la relation d'Euler et par double comptage de la relation d'incidence arête/face, on obtient aisément a = 3n + 3. De la relation d'incidence arête/sommet on déduit $2a = \sum_s \operatorname{degré}(s)$, la somme portant sur tous les sommets de \mathcal{T} . En coupant cette somme en trois selon les sommets internes de degrés inférieurs ou supérieurs à d et les 3 sommets de la face externe, et en remarquant que tout sommet a degré 3 au moins, on obtient

$$2a \ge 3x + (d+1)y + 9$$

Ce qui, avec les relations précédentes implique

$$x \ge \frac{(d-5)n+3}{d-2} \ge \frac{(d-5)n}{d-2}$$

Mais l'algorithme glouton sélectionne un ensemble maximal de sommets. Dit autrement, l'ensemble des sommets sélectionnés et leurs voisins couvrent les sommets internes de degrés au plus d. On en déduit $(d+1)n_s \ge x$, soit encore

$$n_s \ge \frac{d-5}{(d+1)(d-2)}n$$

On construit une suite de triangulations $\mathcal{T} = \mathcal{T}_1, \ldots, \mathcal{T}_m$ ayant respectivement $n = n_1, \ldots, n_m$ sommets internes avec $n_m = 0$. On obtient \mathcal{T}_{i+1} à partir de \mathcal{T}_i en 2 étapes :

- 1. on sélectionne dans \mathcal{T}_i un ensemble maximal de sommets internes, indépendants et de degré au plus d par l'algorithme glouton ci-dessus,
- 2. on supprime ces sommets de \mathcal{T}_i et on retriangule l'étoile de ces sommets (de taille au plus d) en temps constant par étoile.

Le lemme précédent montre que $n_{i+1} \leq (1-\alpha)n_i$, d'où $m = O(\log n)$. On attache de plus à chaque triangle t de \mathcal{T}_{i+1} au plus d pointeurs vers les triangles de \mathcal{T}_i qui intersectent t. Clairement cette structure est de taille $O(\sum_i dn_i) = O(n)$ et peut être construite dans le même temps.

On détermine le triangle de \mathcal{T} contenant un point p de requête par les étapes suivantes

- 1. déterminer si p est dans l'unique triangle de \mathcal{T}_m ,
- 2. pour *i* allant de m-1 à 1, connaissant le triangle *t* de \mathcal{T}_{i+1} contenant *p*, déduire celui de \mathcal{T}_i contenant *p* en parcourant les triangles de \mathcal{T}_i pointés par *t*.

Chacune des $m = O(\log n)$ étapes ci-dessus prend un temps constant puisque l'étoile à parcourir possède au plus d triangles.

Finalement,

Théorème 10.11 Étant donné une triangulation \mathcal{T} du plan, on peut construire en temps linéaire en la taille de \mathcal{T} une structure de taille linéaire, permettant de localiser un point quelconque en temps logarithmique.

Chapitre 11

Recherche multidimensionnelle

Soit d un entier positif. On se donne un ensemble S de n points de \mathbb{R}^d et un ensemble \mathcal{B} de parties de \mathbb{R}^d appelées boîtes. Le problème de la recherche multidimensionnelle est de répondre à une requête du type :

Compter ou reporter les points de S contenus dans une boîte donnée de \mathcal{B} .

Pour répondre le plus rapidement possible à ce type de requête on construit généralement une structure de recherche qui aidera à répondre à toutes les requêtes possibles. Cette construction est donc considérée comme un pré-calcul dont le temps d'exécution peut raisonnablement être important par rapport au temps de réponse de chaque requête individuelle. La taille de cette structure doit par contre rester relativement faible dans la mesure où elle persiste pour toutes les requêtes. Dans la pratique un algorithme possédant un faible temps de requête nécessitera une structure plus volumineuse qu'un algorithme moins performant. Un exemple extrême consiste en une structure de taille linéaire se limitant à la liste des points de S et un algorithme de recherche au mieux linéaire obtenu en testant individuellement si chaque point de S est contenu dans la boîte de requête. Le problème général de la recherche multidimensionnelle suppose ainsi de faire des compromis entre l'espace requis pour la structure de données et la rapidité de réponse à une requête.

Références :

- [dBCvKO08, Chap. 5 and 16], [Mat94], [AE99], [GO04, Chap. 36].

11.1 Recherche orthogonale

La recherche orthogonale désigne le cas particulier de la recherche multidimensionnelle où l'ensemble des boîtes \mathcal{B} est l'ensemble (infini) des pavés de \mathbb{R}^d .

Une approche directe consiste à pré-calculer toutes les réponses possibles ($\subset \mathcal{P}(S)$) et trouver un critère sur les boîtes pour savoir quand elles donnent la même réponse. Par ce moyen on obtient des temps de requêtes très performants mais des temps de pré-calculs très importants – ce qui n'est pas forcément rédhibitoire – et surtout des stockages très importants - ce qui est plus ennuyeux car ils persistent. Exemple dans le plan : on trace une verticale et une horizontale par chaque point de S de manière à obtenir une grille. Deux boîtes donnent la même réponse si et seulement si leurs coins supérieurs gauches et inférieurs droits sont dans les mêmes cases de la grille. On a donc au plus $\binom{(n+1)^2}{2}$ réponses non-vides possibles (la grille possède $(n+1)^2$ cases). Comme chaque réponse peut contenir O(n) points, on obtient naïvement une structure de stockage de taille $O(n^5)$ pour le problème du report des points.

On caractérise un algorithme de recherche par la paire (taille stockage, temps de requête).

11.1.1 Recherche unidimensionnelle

En dimension 1, le problème de la recherche orthogonale revient à compter ou reporter tous les points de S contenus dans un intervalle [x, x'] de requête.

Pour ce faire, on utilise un arbre binaire de recherche équilibré (la hauteur des sous-arbres gauche et droit de tout noeud diffère de un au plus) avec les points rangés aux feuilles, les noeuds internes contenant des valeurs de séparation entre les clés des sous-arbres gauche et droit. On suppose également que les feuilles sont chaînées dans l'ordre croissant des coordonnées des points. Pour le problème du report, on peut chercher dans l'arbre la feuille contenant x puis marcher jusqu'à x' en utilisant le chaînage. Ceci fournit une réponse en temps ($\log n + k$), où k est le nombre de points à reporter. Cette méthode se généralise difficilement en dimension supérieure. Pour y remédier on commence par introduire la notion suivante :

Définition 11.1 L'ensemble des clés des feuilles associées à un sous arbre de racine ν s'appelle l'ensemble canonique de ν et est noté $P(\nu)$.

Plutôt que d'utiliser un chaînage aux feuilles on travaille avec la notion d'ensemble canonique. On commence ainsi par rechercher les feuilles contenant x et x' dans l'arbre de recherche. Les chemins de recherche de x et x' partent de la racine, restent confondus sur une certaine longueur, puis se séparent en deux sous-chemins γ_x et $\gamma_{x'}$ à partir d'un noeud ν_s . Il est facile de voir que l'ensemble cherché est l'union des ensembles canoniques des enfants droits des noeuds internes du chemin γ_x et des enfants gauches des noeuds internes du chemin $\gamma_{x'}$. À cette union, on ajoute éventuellement les feuilles de γ_x et γ'_x suivant les comparaisons de x ou x' relativement à ces feuilles. Notons que le calcul de l'ensemble canonique $P(\nu)$ d'un noeud ν peut s'obtenir en temps proportionnel à sa taille $|P(\nu)|$ puisque la taille d'un arbre binaire équilibré est proportionnelle à son nombre de feuilles. En résumé,

Lemme 11.2 (Le problème de la recherche orthogonale en dimension 1) En utilisant un arbre binaire de recherche on obtient : pré-calcul : $O(n \log n)$ espace : O(n)temps requête : $O(\log n + k)$ où k est le nombre de points à reporter.

Exercice 11.3 Modifier l'algorithme pour le problème du comptage seul et non du report des points eux-mêmes. Quelle est la complexité d'une requête ?

Voir également : arbres d'intervalles, arbre de recherche de priorité et arbres de segments.

11.1.2 Kd-trees (arbres k-dimensionnels) (Bentley 1975)

Pour résoudre le problème de la recherche orthogonale en dimension d > 1, on construit récursivement un arbre binaire équilibré sur les points de S, stockés dans les feuilles de cet arbre, tel que pour chaque noeud ν de profondeur k, les ensembles canoniques de ses noeuds enfants constituent une partition de $P(\nu)$ en deux selon la valeur médiane de la $(k \mod d)$ -ème coordonnée. Ainsi, l'ensemble canonique de l'enfant gauche (resp. droit) de ν est le sous-ensemble des points de $P(\nu)$ dont la $(k \mod d)$ -ème coordonnée est majorée (resp. strictement minorée) par cette valeur médiane.

Notons que la valeur médiane des points ordonnés selon une coordonnée particulière peut se calculer en temps linéaire. On peut soit utiliser d listes – une par coordonnées – triées au départ et que l'on décime au fur et à mesure que l'on descend dans l'arbre, soit utiliser l'algorithme linéaire classique de calcul de médiane rappelé ci-dessous :

- 1. Choisir un pivot soit aléatoirement soit par la méthode suivante : couper la liste en n/5 groupes de 5 clés et calculer la médiane de chaque groupe, puis calculer récursivement la médiane de ces médianes,
- 2. scinder la liste en deux selon cette valeur,
- 3. itérer sur la sous-liste contenant la vraie médiane (i.e. la valeur de rang milieu).

Complexité : T(n) = O(n) + T(n/5) + T(7n/10) = O(n). En effet la plus petite sous-liste contient au moins $3 \times 1/2 \times n/5$ éléments et du coup la plus grande en contient au plus 7n/10.

Dans le plan, le temps de construction C(n) d'un 2*d*-arbre est donné par

$$C(n) = \begin{cases} O(1) & \text{si } n = 1\\ O(n) + 2C(\lceil n/2 \rceil)) & \text{sinon} \end{cases}$$
(11.1)

On en déduit un temps de construction en $O(n \log n)$.

On peut associer à chaque noeud ν d'un 2*d*-arbre arbre sur S une boîte $B(\nu)$ de la forme $[x_1, x'_1] \times [x_2, x'_2]$ avec $x_1, x'_1, x_2, x'_2 \in \mathbb{R}$, de sorte que $P(\nu) = S \cap B(\nu)$. Pour cela, on associe à la racine la boîte égale au plan tout entier et on associe aux enfants d'un noeud ν de profondeur k les deux boîtes obtenues en coupant la boîte $B(\nu)$ en deux par le plan d'équation $x_{k \mod 2} = x_{med}$ où x_{med} est la valeur médiane de séparation associée à ce noeud. Clairement, les boîtes de tous les noeuds de même profondeur forment une partition de l'espace et de ce fait leurs ensembles canoniques constituent une partition de S.

Pour rechercher tous les points dans une boîte B donnée on descend récursivement à partir de la racine d'un 2d-arbre arbre sur S:

- 1. Si *B* contient la boîte $B(\nu)$ du noeud courant ν on renvoie l'ensemble canonique $P(\nu)$ et on arête la récursion,
- 2. sinon, si B et $B(\nu)$ sont disjoints, on arête la récursion,

3. sinon on lance la recherche sur les deux enfants de ν .

L'ensemble $S \cap B$ est l'union (disjointe) des $P(\nu)$ collectés lors de la recherche. Le temps de recherche est égal au nombre de noeuds visités plus le temps total pour rendre les $P(\nu)$. Ce dernier temps est proportionnel au nombre de points de $S \cap B$. En effet, l'ensemble canonique d'un noeud peut être collecté en temps proportionnel à sa taille puisque les 2d-arbres sont équilibrés. Pour majorer le nombre de noeuds internes visités (cas 3 dans la récurrence ci-dessus) on remarque que ceux-ci ont leur boîte intersectée par le bord de Bet donc par les droites supports de B. En majorant le nombre de boîtes $B(\nu)$ intersectées par une droite donnée de direction horizontale ou verticale on obtient donc un majorant du nombre de noeuds internes visités (à un facteur 4 près) durant la recherche pour B.

Fixons une droite verticale D. On note I(n) le nombre maximal de noeuds internes dont la boîte intersecte D dans un 2*d*-arbre quelconque de taille n. Puisque la racine d'un 2*d*-arbre est associée à une dichotomie verticale, la droite D ne peut intersecter que 2 des 4 boîtes associées aux noeuds de profondeur 2. Comme les noeuds de profondeur 2 sont également associés à des dichotomies verticales et eux-mêmes racines de 2*d*-arbres de taille au plus $\lceil n/4 \rceil$, on peut écrire

$$I(n) \leq \begin{cases} O(1) & \text{si } n = 1\\ 2 + 2I(\lceil n/4 \rceil)) & \text{sinon} \end{cases}$$
(11.2)

D'où, suivant le master theorem 1.34 ou suivant un calcul directe sur l'arbre de récursion, $I(n) = O(\sqrt{n})$. Un raisonnement analogue permet de borner le nombre de noeuds intersectés par une droite horizontale.

En résumé

Lemme 11.4 Le problème de la recherche orthogonale dans le plan peut se résoudre en utilisant un 2d-arbre avec : $pré-calcul : O(n \log n)$ espace : O(n)temps requête : $O(\sqrt{n} + k)$ où k est le nombre de points à reporter.

Exercice 11.5 Généraliser à d > 2 dimensions. Montrer en particulier que le temps de recherche est en $O(n^{1-1/d} + k)$.

11.1.3 Arbres de domaines

Les arbres de domaines (range trees), introduits par Bentley en 1979, sont des structures de partitionnement de l'espace à plusieurs niveaux. En dimension 2, on commence par construire un arbre binaire de recherche équilibré selon la première coordonnée, puis pour chaque noeud ν on construit un arbre binaire de recherche équilibré sur l'ensemble canonique de ce noeud selon la deuxième coordonnée. En remarquant que les ensembles canoniques $P(\nu)$ correspondant à des noeuds de profondeur donnée forment une partition de l'ensemble des points, et que la hauteur de l'arbre de recherche "primaire" est en $O(\log n)$ on voit que la taille d'un arbre de domaines est en $O(n \log n)$. Pour la construction, on commence par trier les points selon la deuxième coordonnées y. Puis on crée une racine en la faisant pointer sur un arbre binaire de recherche selon la deuxième coordonnée. Cet arbre binaire de recherche est construit en temps linéaire à partir de la liste triée de manière "bottom-up". On scinde alors la liste en deux selon x et on construit les deux sous-listes triées en y à partir de la liste triée de départ. Puis on continue récursivement. De cette manière le temps de construction est proportionnel à la taille de la structure finale (chaque arbre "secondaire" associé à un ensemble canonique est construit en temps proportionnel à sa taille).

Pour effectuer une requête avec une boîte $B = [x, x'] \times [y, y']$ on commence par rechercher les valeurs x et x' dans l'arbre primaire, puis pour chaque sous-arbre à droite (gauche) du chemin de recherche γ_x ($\gamma_{x'}$) situé après le noeud de séparation on fait une recherche en y dans la structure secondaire associée à sa racine. Le temps de la recherche est donc majoré par

$$\sum_{\nu} O(\log n + k_{\nu}) = O(\log^2 n + k)$$

où ν décrit les racines des sous-arbres sus-cités et k_{ν} le nombre de points reportés dans les ensembles canoniques associés.

Lemme 11.6 Le problème de la recherche orthogonale dans le plan peut se résoudre en utilisant un arbre de domaines de dimension 2 avec : $pré-calcul : O(n \log n)$ $espace : O(n \log n)$ $temps requête : O(\log^2 n + k)$ où k est le nombre de points à reporter.

Remarque : Par rapport aux 2d-arbres on a diminué le temps de requête mais augmenté la taille de la structure.

Exercice 11.7 Généraliser les arbres de domaines à d > 2 dimensions.

Exercice 11.8 Jusqu'à maintenant, on a implicitement supposé que les k-ièmes coordonnées des points étaient deux à deux distinctes pour chaque k. Si ce n'est pas le cas on peut considérer la transformation

 $(x, y) \mapsto ((x, y), (y, x))$ et utiliser l'ordre lexicographique sur les couples (x, y). La boîte de requête devient alors $[(x, -\infty), (x', \infty)] \times [(y, -\infty), (y', \infty)]$. Généraliser ce procédé à la dimension d.

Remarque : cette astuce revient à perturber les données, c'est à dire à opérer une transformation du type $(x, y) \mapsto (x + \epsilon y, y + \epsilon x)$ où $\epsilon |y_i|_{max} \leq \min_{x_i \neq x_j} |x_i - x_j|$ et de même pour y.

Exercice 11.9 Si on permet aux points d'avoir une partie de leurs coordonnées identiques on peut s'intéresser à tous les points ayant certaines de leurs coordonnées fixées par une requête. On parle alors de requête d'identification partielle (partial match query).

1. Montrer qu'avec un 2d-arbre on peut répondre à une requête d'identification partielle en temps $O(\sqrt{n}+k)$ où k est le nombre de points à reporter. 2. Trouver une structure de données utilisant un espace linéaire et qui répond à une requête d'identification partielle en temps $O(\log n + k)$.

3. Montrer qu'avec un kd-arbre de dimension d (i.e. un dd-arbre) on peut répondre à une requête d'identification partielle sur s < d coordonnées en temps $O(n^{1-s/d} + k)$.

4. Trouver une structure de données pour répondre à une requête d'identification partielle en temps $O(d \log n + k)$ en dimension d, si on permet que la structure soit de taille $O(d2^dn)$.

11.1.4 Fractionnement en cascade

Le fractionnement en cascade (Lueker 1978 et Willard 1978) est une modification des arbres de domaines qui permet de gagner un facteur $\log n$ dans le temps de requête. On s'intéresse à la dimension 2. Dans les arbres de domaines les ensembles canoniques sont rangés dans des arbres binaires de recherche en la deuxième coordonnée. On remplace ces arbres par de simples listes triées (selon la deuxième coordonnée) en ajoutant à chaque élément d'une liste un pointeur vers le premier élément supérieur ou égal dans chacune des listes des deux noeuds enfants. On conserve de plus l'arbre binaire de recherche sur l'ensemble canonique de la racine.

La taille de la structure est la même que celle d'un arbre de domaines, $O(n \log n)$, et elle peut facilement être construite dans le même temps.

Pour effectuer une requête avec une boîte $B = [x, x'] \times [y, y']$ on procède comme pour un arbre de domaines en déterminant les chemins de recherches γ_x et $\gamma_{x'}$. On recherche ensuite dans l'ensemble canonique du noeud racine le premier sommet dont l'ordonnée est supérieure ou égale à y et l'on "propage" ce premier sommet le long des chemins de recherche à l'aide des pointeurs précédemment définis. Dans chaque noeud dont on doit reporter une partie de l'ensemble canonique on marche alors à partir de ce premier sommet jusqu'au dernier sommet d'ordonnée inférieure ou égale à y'. On en déduit un temps de requête en

$$\log n + \sum_{\nu} (O(1) + k_{\nu}) = O(\log n + k)$$

où ν et k_{ν} sont définis comme pour les arbres de domaines.

Lemme 11.10 Le problème de la recherche orthogonale dans le plan peut se résoudre en utilisant le fractionnement en cascade avec : $pré-calcul : O(n \log n)$ $espace : O(n \log n)$ $temps requête : O(\log n + k)$ où k est le nombre de points à reporter.

Exercice 11.11 Généraliser la technique du fractionnement en cascade en dimension d > 2. Montrer en particulier qu'on obtient un temps de recherche en $O(\log^{d-1} n + k)$.

Note : Le problème de la recherche othogonale est l'un des plus étudiés en géométrie algorithmique. Étant donné l'interaction entre temps de requête et taille de la structure,

Chazelle [Cha90] a étudié la taille minimale requise si on impose un temps de requête avec report en temps O(k + polylog(n)). Autrement dit, on impose un temps de réponse proportionnel à cette dernière plus un polylog¹. Chazelle étudie cette question dans le cadre d'une variante de machine à pointeurs et montre que toute structure de recherche en dimension d nécessite un espace $\Omega(n(\log n/\log \log n)^{d-1})$. En deux dimensions, et pour le modèle RAM en supposant que les points sont sur une grille entière de taille n^2 où la machine RAM traite des entiers de log n bits, Chan et al. [CLP11] obtiennent une première structure de recherche de taille $O(n \log \log n)$ pour un temps de requête $O(\log \log n(1+k))$ et une seconde structure de taille O(n) pour un temps de requête $O((1 + k) \log^{\varepsilon} n)$.

11.2 Recherche simpliciale et par demi-espaces

S désigne à nouveau un ensemble de n points de \mathbb{R}^d fixé une fois pour toute. Le problème de la recherche simpliciale (resp. par demi-espace) consiste à reporter, ou compter, tous les points de S contenus dans un simplexe (resp. un demi-espace) de requête. Si une requête de type demi-espace peut sembler plus simple que pour un pavé parallèle aux axes, elle est en réalité bien plus difficile et bien plus générale du fait de l'orientation a priori quelconque de l'hyperplan support du demi-espace de requête. En particulier, la recherche par demi-espaces permet de répondre à des requêtes portant sur des boîtes non-linéaires par un procédé classique de linéarisation. Supposons par exemple vouloir effectuer une requête se traduit aisément en une requête de type demi-espace dans \mathbb{R}^3 . La recherche simpliciale, que l'on peut voir comme une application de la recherche par demi-espaces, permet quant-à elle de répondre à des requêtes portant sur des boîtes de tailles bornées en effectuant une triangulation préalable des polyèdres.

11.2.1 Arbre de partitions

L'idée est de partitionner les points en sous-ensembles de tailles approximativement égales inclus dans des domaines dont la description est simple de sorte que la boîte de requête intersecte une faible proportion de ces domaines. On procède alors récursivement sur les points inclus dans chaque domaine pour définir un arbre de partitions.

On regarde le problème dans le plan.

Définition 11.12 Une partition simpliciale d'un ensemble S de n points du plan est une famille de couples $\{(S_i, \Delta_i)\}_{i \in I}$ où les S_i forment une partition de S et les Δ_i sont des triangles du plan de sorte que pour chaque i on a $S_i \subset \Delta_i$. Notons que les triangles Δ_i peuvent se chevaucher et qu'un point de S peut appartenir à plusieurs Δ_i . Le nombre |I|de parties est la taille de la partition.

Soit $r \geq 1$. Une partition simpliciale de paramètre r, ou r-partition, sur S est une partition simpliciale de S dont chaque sous-ensemble S_i contient entre n/r et 2n/r points. Notons que la taille d'une r-partition est comprise entre r/2 et r.

^{1.} polylog(n) désigne une fonction de la forme $P(\log n)$ pour un polynôme P fixé.

Le nombre de croisements d'une droite avec une partition simpliciale est le nombre de triangles de la partition intersectés par cette droite. Le nombre de croisements de la partition est le maximum de ce nombre sur toutes les droites du plan.

Le théorème 12.9 de la partition simpliciale affirme que pour tout r fixé on peut construire en temps O(n) une r-partition de S de nombre de croisements $O(\sqrt{r})$.

On construit récursivement un *arbre de partitions* de paramètre r sur un ensemble S de n points en associant à la racine un nombre d'enfants en correspondance avec les sousensembles d'une partition simpliciale de S de paramètre r fixé. On stocke les descriptions des triangles Δ_i de la partition au niveau des enfants puis on continue récursivement sur chaque enfant avec son sous-ensemble S_i associé tant que ce sous-ensemble compte au moins 2r points.

Lemme 11.13 Soit r > 2. Un arbre de partitions de paramètre r sur un ensemble de n points a une taille linéaire (i.e. en O(n)) et peut être construit en temps $O(n \log n)$.

Preuve : Puisque chaque partition simpliciale utilisée est de paramètre r, le degré de chaque noeud interne de l'arbre de partitions est au moins r/2. Soit n_I et n_E les nombres respectifs de noeuds internes et externes (feuilles) de l'arbre de partitions. En comptant de deux façons différentes le nombre d'arêtes de l'arbre de partitions à partir de l'extrémité respectivement inférieure ou supérieure de chaque arête, on obtient :

$$(r/2)n_I \le n_I + n_E - 1.$$

Soit

$$n_I \le \frac{2n_E - 2}{r - 2}.$$

Or chaque feuille de l'arbre de partitions est associée à au moins un point et les ensembles associés aux feuilles sont disjoints. On en déduit $n_E \leq n$ et par suite la taille de l'arbre de partitions est linéaire.

Le temps de construction C(n) vérifie d'après le théorème 12.9 de la partition simpliciale

$$C(n) \le \begin{cases} O(1) & \text{si } n = 1\\ O(n) + \sum_{\nu} C(n_{\nu}) & \text{sinon} \end{cases}$$
(11.3)

où ν parcourt les enfants de la racine. Comme $n_{\nu} \leq 2n/r$, l'arbre de partitions a une hauteur $O(\log n)$. Par ailleurs, l'ensemble des noeuds de profondeur donnée forme une partition des n points et peut donc être traité en temps O(n). On en déduit $C(n) = O(n \log n)$.

Pour répondre à une requête du type demi-plan l^+ bordé par une droite l on part de la racine puis on teste si chacun des triangles des (au plus r) enfants est contenu dans l^+ , disjoint de l^+ , ou intersecte l. Dans les deux premiers cas on sait quoi faire sinon on "récurre" sur les triangles enfants intersectés. Ceci permet de sélectionner un ensemble de noeuds dont les ensembles canoniques forment une partition des points contenus dans l^+ . La complexité R(n) de cette sélection est donnée par

$$R(n) \leq \begin{cases} O(1) & \text{si } n = 1\\ O(r) + \sum_{\nu} R(n_{\nu}) & \text{sinon} \end{cases}$$
(11.4)

Soit, compte tenu du paramètre r de la partition simpliciale :

$$R(n) \le cr + c\sqrt{r}R(2n/r).$$

pour une certaine constante c que l'on peut supposer > $1/\sqrt{2}$. En choisissant $r = \lfloor 2(c\sqrt{2})^{1/\epsilon} \rfloor$ on a

$$0 < \log_{r/2} c\sqrt{r} = \frac{\log(c\sqrt{r})}{\log(r/2)} = \frac{\log\sqrt{r/2}\log(c\sqrt{2})}{\log(r/2)} = \frac{1}{2} + \frac{\log(c\sqrt{2})}{\log(r/2)} \le \frac{1}{2} + \frac{\log(c\sqrt{2})}{\log((c\sqrt{2})^{1/\epsilon})} \le \frac{1}{2} + \epsilon \cdot \frac{\log(c\sqrt{2})}{\log((c\sqrt{2})^{1/\epsilon})} \le \frac{1}{2} + \epsilon \cdot \frac{\log(c\sqrt{2})}{\log(c\sqrt{2})} \le \frac{1}{2} + \epsilon \cdot \frac{\log(c\sqrt{2})}{\log(c\sqrt{2})} \le \frac{1}{2} + \epsilon \cdot \frac{\log(c\sqrt{2})}{\log(c\sqrt{2})} \le \frac{1}{2} + \frac{\log(c\sqrt{2})}{\log(c\sqrt{2})} \le \frac{\log(c\sqrt{$$

On en déduit, par le master theorem 1.34, $R(n) = O(n^{\log_{r/2} c\sqrt{r}}) = O(n^{1/2+\epsilon})$. Dans le cas du report, il faut rendre les ensembles canoniques des noeuds sélectionnés. Pour chaque noeud, cet ensemble est l'union des points stockés aux feuilles du sous-arbre dont il est racine. Comme ce sous-arbre à une taille linéaire, on en déduit que le problème du report peut-être résolu avec un temps supplémentaire proportionnel à la taille de la réponse.

Note : En prenant $r = \sqrt{n}$ et en résolvant $R(n) = O(r) + O(\sqrt{r})R(2n/r)$ on trouve $R(n) = O(\sqrt{n} 2^{O(\log \log n)}) = O(\sqrt{n} \operatorname{polylog} n).$

Pour répondre à une requête de type triangle au lieu de demi-plan on peut utiliser la même structure et obtenir la même complexité asymptotique de recherche en remarquant que, à chaque niveau de la recherche, le nombre de triangles d'une partition intersectés par les 3 droites supports d'un triangle de requête est en $O(3\sqrt{r})$. Finalement :

Lemme 11.14 Le problème de la recherche simpliciale dans le plan peut se résoudre en utilisant un arbre de partitions avec :

 $pré-calcul : O(n \log n)$

espace : O(n)

temps requête : $O(n^{1/2+\epsilon})$ pour le décompte et $O(n^{1/2+\epsilon}+k)$ pour le report où k est le nombre de points à reporter.

11.2.2 Arbres de cuttings et recherche par demi-plan

Les arbres de partitions ont évidemment une taille optimale mais un temps de requête important. Quitte à augmenter la taille de la structure de recherche, il est possible d'obtenir des temps de recherche logarithmiques. C'est l'objet de ce qui suit. La méthode repose ici sur une transformation préalable des données par la dualité point/droite du plan. Celle ci, notée *, est donnée par la transformation $(a, b) \mapsto \{y = ax - b\}$, et son inverse, et définit une bijection entre le plan et l'ensemble des droites non verticales du plan.

Propriété : le point p est au dessus de la droite d si et seulement si d^* est au dessus de p^* .

Étant donné une droite d, trouver tous les points s_i de S au dessus de d revient ainsi à trouver l'ensemble des droites de $\{s_i^* | s_i \in S\}$ au dessous de d^* . Clairement cet ensemble

ne dépend que de la cellule de l'arrangement des droites s_i^* contenant d^* . Il y a $O(n^2)$ telles cellules. On peut donc espérer un algorithme utilisant une place $O(n^2)$ avec un temps de requête en $O(\log n)$.

Définition 11.15 Soit L un ensemble de n droites du plan et soit $r \in [1, n]$. Un (1/r)cutting pour L est une partition du plan en triangles (possiblement non bornés) telle que chaque triangle est intersecté par au plus n/r droites de L. La taille de ce cutting est son nombre de triangles.

Le théorème 12.1 affirme que l'on peut construire en temps O(nr) un (1/r)-cutting de taille $O(r^2)$ en collectant pour chaque triangle du cutting les droites de L qui le coupent.

On s'intéresse tout d'abord au problème du décompte. On définit pour cela récursivement un *arbre de cuttings* pour L:

- si L contient une unique droite alors son arbre est réduit à une feuille qui stocke cette droite,
- sinon on crée autant d'enfants de la racine que de triangles dans un (1/r)-cutting de L. Pour chaque enfant ν , associé au triangle $\Delta(\nu)$, on construit récursivement un arbre de cuttings pour les droites $L(\nu) \subset L$ intersectant $\Delta(\nu)$. L'ensemble de droites $L(\nu)$ est appelé l'*ensemble canonique* de ν . Pour le problème du comptage on stocke également pour chaque enfant les nombres $\ell^-(\nu)$ et $\ell^+(\nu)$ de droites de L respectivement au dessus et au dessous de $\Delta(\nu)$.

La taille T(n) d'un arbre de cuttings construit à l'aide de (1/r)-cuttings, chacun de taille $O(r^2)$, vérifie

$$T(n) \le \begin{cases} O(1) & \text{si } n = 1\\ O(r^2) + \sum_{\nu} T(n_{\nu}) & \text{sinon} \end{cases}$$
(11.5)

Où ν décrit les enfants d'un arbre de cuttings sur *n* droites et n_{ν} la taille de son ensemble canonique. Par définition d'un 1/r-cutting on a encore pour une certaine constante *c* :

$$T(n) \le cr^2 + cr^2 T(n/r)$$

D'où $T(n) = O(n^{\log_r cr^2})$. Soit en prenant $r = \lceil c^{1/\epsilon} \rceil$, $T(n) = O(n^{2+\epsilon})$.

Compte tenu du temps de calcul d'un (1/r)-cutting, le temps C(n) de construction de cette structure vérifie

$$C(n) \leq \begin{cases} O(1) & \text{si } n = 1\\ O(nr) + \sum_{\nu} C(n_{\nu}) & \text{sinon} \end{cases}$$
(11.6)

Soit encore $C(n) \leq cnr + cr^2 C(n/r)$ pour une certaine constante c, ce qui donne à nouveau $C(n) = O(n^{2+\epsilon})$ pour $r = \lceil c^{1/\epsilon} \rceil$.

Pour trouver le nombre de droites sous un point p de requête on descend récursivement dans l'arbre de cuttings depuis la racine jusqu'aux feuilles en s'orientant pour chaque noeud visité vers l'unique enfant dont le triangle associé contient p. La somme des nombres ℓ^- associés aux noeuds de ce parcours est le nombre cherché. Bien entendu, en accumulant les valeurs ℓ^+ au lieu de ℓ^- la même structure permet de calculer le nombre de droites au dessus d'un point de requête. Le temps de recherche R(n) vérifie ainsi :

$$R(n) \le \begin{cases} O(1) & \text{si } n = 1\\ O(r^2) + R(n/r) & \text{sinon} \end{cases}$$
(11.7)

D'où $R(n) = O(\log n)$ si r est constant.

Pour le problème du report, pour chaque noeud ν de l'arbre de cuttings, si μ est son parent, on stocke explicitement au niveau de ν les sous-ensembles $L^+(\nu), L^-(\nu) \subset L(\mu)$ de droites respectivement au dessus et au dessous du triangle $\Delta(\nu)$. L'ensemble des droites au dessous (resp. au dessus) d'un point de requête est obtenu récursivement comme dans le cas du décompte en accumulant cette fois les ensembles L^- (resp. L^+) le long du parcours. Le temps de requête se décompose alors en un temps de parcours $O(\log n)$ comme pour le décompte plus un terme proportionnel à la taille de la réponse. La taille T'(n) de la structure de recherche vérifie maintenant

$$T'(n) \le \begin{cases} O(1) & \text{si } n = 1\\ O(nr^2) + \sum_{\nu} T'(n_{\nu}) & \text{sinon} \end{cases}$$
(11.8)

D'où $T'(n) = O(nr^2 + r^2T'(n/r))$. Par application du master theorem 1.34 on déduit à nouveau $T'(n) = O(n^{2+\epsilon})$. En résumé,

Lemme 11.16 Le problème de la recherche par demi-plan peut se résoudre en utilisant un arbre de cuttings avec : **pré-calcul :** $O(n^{2+\epsilon})$

espace : $O(n^{2+\epsilon})$

temps requête : $O(\log n)$ pour compter et $O(\log n + k)$ pour reporter où k est le nombre de points à reporter.

11.2.3 Application à la recherche simpliciale

Soit Δ un triangle, intersection de trois demi-plans H_1, H_2, H_3 bordés respectivement par les droites h_1, h_2, h_3 . On suppose que H_1 est au dessous de h_1 tandis que H_2 et H_3 sont au dessus de leur droite bordante respective. Les autres cas se traitent de manière analogue. Le problème de la sélection des points de S inclus dans Δ se dualise alors en un problème du type : trouver les droites de S^* simultanément au dessus de h_1^* et au dessous de h_2^* et de h_3^* .

Pour répondre à ce type de requête – et donc par dualité à une requête de type triangle – on peut utiliser un structure d'arbre de cuttings à trois niveaux : le premier niveau est un arbre de cuttings simple. On associe de plus à chaque noeud ν deux arbres de cuttings (à deux niveaux) : l'un sur son ensemble $L^+(\nu)$ et l'autre sur son ensemble $L^-(\nu)$. Finalement on associe à chaque noeud de la structure secondaire deux arbres de cuttings simples sur ses ensembles L^+ et L^- . La recherche procède comme suit. On effectue une requête avec h_1^* sur l'arbre primaire pour sélectionner les droites au dessus de h_1^* . Pour chaque noeud ν sélectionné dans cette recherche, plutôt que de rendre (ou comptabiliser) $L^+(\nu)$, on effectue une requête avec h_2^* sur sa structure secondaire afin de sélectionner les droites de $L^+(\nu)$ au dessous de h_2^* . Finalement on effectue une requête avec h_3^* sur les structures

associées aux noeuds secondaires sélectionnés pour rechercher les droites au dessous de h_3^* . Clairement, cette procédure permet de sélectionner les droites simultanément au dessus de h_1^* et au dessous de h_2^* et de h_3^* .

Le temps de recherche dans une structure secondaire vérifie :

$$R'(n) \le \begin{cases} O(1) & \text{si } n = 1\\ O(\log n + r^2) + R'(n/r) & \text{sinon} \end{cases}$$
(11.9)

D'où $R'(n) = O(\log^2 n)$ si r est constant. Le temps de recherche dans la structure primaire vérifie donc :

$$R(n) \le \begin{cases} O(1) & \text{si } n = 1\\ O(\log n^2 + r^2) + R(n/r) & \text{sinon} \end{cases}$$
(11.10)

D'où $R(n) = O(\log^3 n)$ si r est constant. La taille de la structure secondaire vérifie :

$$T'(n) \le \begin{cases} O(1) & \text{si } n = 1\\ O(r^2 + n^{2+\epsilon}) + \sum_{\nu} T'(n_{\nu}) & \text{sinon} \end{cases}$$
(11.11)

Soit (cf. master theorem) $T'(n) = O(n^{2+\epsilon})$ si r est constant et suffisamment grand. On en déduit de même que la taille de la structure primaire est un $O(n^{2+\epsilon})$. On montre selon les mêmes lignes que celle-ci peut-être construite dans le même temps. Finalement,

Lemme 11.17 Le problème de la recherche simpliciale dans le plan peut se résoudre en utilisant un arbre de cuttings à trois niveaux avec : **pré-calcul :** $O(n^{2+\epsilon})$ **espace :** $O(n^{2+\epsilon})$

temps requête : $O(\log^3 n)$ pour compter et $O(\log^3 n + k)$ pour reporter où k est le nombre de points à reporter.

Chapitre 12

Cuttings et partitions simpliciales

12.1 Cuttings

Soit L un ensemble de n droites du plan en position générale et soit $r \in [1, n]$. On rappelle qu'un (1/r)-cutting pour L est un découpage du plan en triangles généralisés (i.e. des vrais triangles du plan projectif si on y plonge canoniquement le plan euclidien; ils sont possiblement non bornés dans le plan euclidien) tel que l'intérieur de chaque triangle rencontre au plus n/r droites ¹ de L. On entend par découpage que les intérieurs des triangles sont deux à deux disjoints et que l'union des triangles recouvre le plan. La taille de ce cutting est son nombre de triangles.

L'objectif est de construire un (1/r)-cutting de la plus petite taille possible. L'argument suivant montre que cette taille est minorée par un $\Omega(r^2)$. Par l'hypothèse de position générale, l'arrangement des n droites de L possède $\binom{n}{2} = \Omega(n^2)$ sommets. Un triangle d'un (1/r)-cutting étant coupé par au plus n/r droites, il contient au plus $\binom{n/r}{2} = O((n/r)^2)$ de ces sommets. Le plan doit donc être couvert par au moins $\Omega(r^2)$ triangles. Le théorème suivant montre que l'on peut construire des (1/r)-cuttings dont la taille est optimale à un facteur constant près. La preuve repose sur une analyse d'échantillonnages aléatoires à la Clarkson [CS89, CMS93]. Une approche non randomisée due à B. Chazelle et J. Friedman [CF90] est exposée par J. Matoušek dans [Mat91]. Voir aussi [PA95, p. 173-175].

Théorème 12.1 Pour tout ensemble L de n droites du plan et tout $r \leq n$ il existe un (1/r)-cutting de taille $O(r^2)$ qui peut être construit en temps O(nr) en collectant de plus pour chaque triangle les droites qui le coupent.

La construction repose sur une triangulation de l'arrangement d'un sous-ensemble des droites de L. Cette triangulation peut être obtenue de manière canonique en triangulant chaque cellule (polygone) de l'arrangement à partir de son sommet le plus bas (minimal pour l'ordre lexicographique sur les coordonnées), c'est à dire en remplaçant chaque cellule

^{1.} Pour étendre le résultat à des arrangements non-simples on voit bien qu'il est nécessaire de restreindre la condition à l'intérieur des triangles puisque tout sommet de degré 2n/r ou plus de l'arrangement de L est incident à au moins n/r droites.

par le cône issu du sommet le plus bas sur les arêtes du polygone ne contenant pas ce sommet. Ce sommet peut être à l'infini si la cellule est non-bornée. Un triangle peut donc être soit un triangle fini au sens usuel, soit un secteur bordé par deux demi-droites nonparallèles, soit un demi-cylindre bordé par deux demi-droites parallèles et un segment. De manière générale on obtient la *triangulation canonique* d'un arrangement d'hyperplans de \mathbb{R}^d en triangulant récursivement le *i*-squelette de l'arrangement par "étoilement" de chaque *i*-cellule à partir de son sommet le plus bas.²

Montrons tout d'abord comment collecter les droites coupant chaque triangle d'un (1/r)cutting de taille $k = O(r^2)$. Remarquons tout d'abord que l'on peut déterminer pour chaque droite $\ell \in L$ un triangle la coupant en temps $O(\log k)$. Il suffit par exemple de pré-calculer pour une droite donnée la liste ordonnée de ses intersections avec les triangles du cutting, puis de situer en temps $O(\log k)$ dans cette liste l'intersection de ℓ avec cette droite. Une fois connu un triangle intersectant ℓ il suffit de longer ℓ dans le cutting (donné sous forme de liste d'adjacence entre triangles³) pour déterminer tous les triangles du cutting coupés par ℓ . Ceci prend un temps proportionnel au nombre n_{ℓ} de triangles coupés. Le temps total requis pour collecter toutes les intersections est alors $\sum O(\log k + n_d)$, la somme portant sur les n droites de L. Or $\sum n_d \leq k(n/r)$ par définition d'un (1/r)-cutting. Pour $k = O(r^2)$ comme dans le théorème, on obtient un temps O(nr).

On montre d'abord une version sous-optimale du théorème.

Lemme 12.2 Pour tout ensemble L de n droites du plan et tout $r \leq n$ il existe un (1/r)-cutting de taille $O(r^2 \log^2 r)$ qui peut être construit en temps $O(r^2 \log^2 r + nr \log r)$ en collectant de plus pour chaque triangle les droites qui le coupent.

Preuve I: Considérons l'ensemble des droites coupant l'intérieur d'un triangle quelconque. Lorsque le triangle décrit tous les triangles (généralisés) possibles du plan, les ensembles de droites correspondant forment un système de parties (cf. Chapitre 14). On montre que ce système a une dimension de Vapnik-Chervonenkis finie (cf. [PA95, p.256-257]). Par le corollaire 14.8, ce système admet un $\frac{1}{r}$ -net R de taille $O(r \log r)$. La triangulation canonique $\mathcal{T}(R)$ de l'arrangement des droites de R comporte donc $O(r^2 \log^2 r)$ triangles. De plus, par définition d'un $\frac{1}{r}$ -net, un triangle dont l'intérieur est disjoint de R– c'est en particulier le cas des triangles de $\mathcal{T}(R)$ – est coupé par au plus n/r droites de L. Les triangles de $\mathcal{T}(R)$ forment donc un (1/r)-cutting de taille $O(r^2 \log^2 r)$. \Box

La construction d'un ϵ -net de petite taille peut s'obtenir par échantillonnage aléatoire. La preuve suivante s'appuie directement sur un échantillonnage aléatoire de L sans passer par la théorie de Vapnik-Chervonenkis. Elle utilise en contrepartie quelques résultats sur les échantillonnages aléatoires explicités plus loin.

^{2.} De manière équivalente on peut trianguler récursivement le (d-1)-arrangement induit dans chaque hyperplan par les autres hyperplans de l'arrangement puis étoiler les *d*-cellules à partir de leur sommet le plus bas. Cette méthode fournit bien la même triangulation si on prend garde de déduire correctement le système de coordonnées de chaque hyperplan par "projection" du système de coordonnées canonique.

^{3.} On suppose implicitement ici que le cutting est formé à partir de la triangulation canonique d'un arrangement, de sorte que chaque triangle a au plus trois triangles adjacents.

Preuve II : Soit R un échantillon aléatoire de taille $t = cr \log r$ pour la loi uniforme sur $\binom{L}{t}$ (cf. section 1.8.5) où c est une constante déterminée plus loin. Le théorème 12.4 ci-après indique qu'avec une forte probabilité tout triangle de la triangulation canonique $\mathcal{T}(R)$ de l'arrangement de R intersecte au plus $b\frac{n}{t}\log t$ droites de L pour une certaine constante b. En particulier, $\mathcal{T}(R)$ est un $b\log t/t$ -cutting de taille $O(t^2) = O(r^2\log^2 r)$. Or

$$b\log t/t = b\frac{\log(cr\log r)}{cr\log r} = \frac{b}{rc}\left(1 + \frac{\log\log r}{\log r} + \frac{\log c}{\log r}\right)$$

cette dernière quantité étant majoré par 1/r dès que $r \ge c \ge 3b$. Pour r < c il suffit de choisir $t = c^2 \log c < Kr \log r$ pour une certaine constante K puisqu'un $\frac{1}{c}$ -cutting est alors un $\frac{1}{r}$ -cutting.

Preuve du théorème 12.1 : On considère un r-échantillon aléatoire R pour la loi uniforme sur $\binom{L}{r}$. Certains triangles de la triangulation canonique $\mathcal{T}(R)$ de l'arrangement de R sont possiblement coupés par plus de n/r droites de L. L'idée est de subdiviser chaque triangle Δ de $\mathcal{T}(R)$ par un cutting restreint au sous-ensemble de droites $L_{\Delta} \subset L$ coupant ce triangle. On peut en effet s'arranger pour que les triangles de la subdivision soient coupés par au plus n/r droites tout en utilisant $O(r^2)$ triangles au total dans les subdivisions. Pour cela on construit à l'aide du lemme 12.2 pour chaque triangle Δ un $1/r_{\Delta}$ -cutting \mathcal{C}_{Δ} de L_{Δ} composé de $O(r_{\Delta}^2 \log r_{\Delta}^2)$ triangles en choisissant $r_{\Delta} = \ell_{\Delta} r/n$ où $\ell_{\Delta} = |L_{\Delta}|$ (bien sûr, on ne fait rien si $\ell_{\Delta} \leq n/r$). Chaque triangle de ce cutting est coupé par au plus $\ell_{\Delta}/r_{\Delta} = n/r$ droites. Ceci est encore vrai pour les $O(r_{\Delta}^2 \log r_{\Delta}^2)$ triangles obtenus en intersectant \mathcal{C}_{Δ} avec Δ et en retriangulant les éventuels k-gones $(k \leq 6)$ résultant. La réunion de ces subdivisions est donc un 1/r-cutting \mathcal{C} de L de taille $O(r^2) + \sum_{\Delta \in \mathcal{T}(R)} r_{\Delta}^2 \log r_{\Delta}^2$. Évaluons l'espérance de cette dernière somme relativement à R.

$$E(\sum_{\Delta} r_{\Delta}^2 \log r_{\Delta}^2) \le E(\sum_{\Delta} r_{\Delta}^4) = (r/n)^4 E(\sum_{\Delta} \ell_{\Delta}^4)$$

Par le lemme 12.7, $E(\sum_{\Delta} \ell_{\Delta}^4) = O((n/r)^4 r^2)$. On en déduit que \mathcal{C} a en moyenne $O(r^2)$ triangles.

On pourra consulter [HP00] pour des constructions effectives de cuttings avec des bornes effectives sur la complexité de l'algorithme.

Le théorème 12.1 admet une version pondérée. Cette fois on se donne une famille pondérée (L, w) de *n* droites avec des poids $w = (w_1, w_2, \ldots, w_n)$ non-négatifs, de poids total $|w| = \sum_i w_i$. Un 1/r-cutting est alors tel que le poids total des droites de *L* coupant l'intérieur d'un triangle du cutting est majoré par |w|/r.

Corollaire 12.3 Pour toute famille pondérée (L, w) de n droites du plan et tout $r \leq n$ on peut construire en temps O(nr) un (1/r)-cutting de taille $O(r^2)$ en collectant de plus pour chaque triangle les droites qui le coupent.

Preuve : Quitte à renormaliser les poids en temps linéaire, on peut supposer que |w| = n. On considère le multi-ensemble L' de droites obtenu en incluant chaque droite de L de poids w_i avec multiplicité $\lceil w_i \rceil$. Notons que $|L'| \leq \sum_i (w_i + 1) \leq 2n$. Suivant le

théorème 12.1, on calcule en temps O(nr) un $\frac{1}{2r}$ -cutting de taille $O(r^2)$ pour la collection non-pondérée de droites L'. Pour cela on peut considérer que toutes les droites de L'sont distinctes à l'aide de perturbations symboliques. Puisque chaque triangle est coupé par au plus $|L'|/(2r) \leq n/r$ droites de L', ce cutting est un (1/r)-cutting pour la famille pondérée (L, w).

12.2 Echantillonnage aléatoire

On effectue ici quelques calculs en moyenne portant sur des arrangements de droites et utiles pour le calcul de cuttings. On s'en tient au cadre des arrangements de droites et de leur triangulation canonique (cf. section 12.1) mais les résultats peuvent être établis dans un cadre plus abstrait comme au chapitre 13. Le terme d'échantillon aléatoire se réfère toujours à la loi uniforme (cf. section 1.8.5) mais les résultats ci-dessous peuvent être établis avec d'autres lois. On pourra consulter [Mul00] à ce sujet.

Soit L un ensemble de n droites du plan en position générale. On note $\mathcal{T}^0(L)$ l'ensemble des triangles de la triangulation canonique de L. On note ensuite $\mathcal{T}(L)$ l'ensemble des triangles réalisables sur L c.-à-.d apparaissant dans la triangulation canonique d'au moins une partie de L, i.e. $\mathcal{T}(L) = \bigcup_{R \subset L} \mathcal{T}^0(R)$. Pour tout triangle $\sigma \in \mathcal{T}(L)$ on note L_{σ} l'ensemble des droites de L qui rencontrent l'intérieur de σ et on pose $\ell_{\sigma} := |L_{\sigma}|$. On note également D_{σ} l'ensemble des droites de $L \setminus L_{\sigma}$ qui rencontrent le bord de σ . On dit que D_{σ} détermine σ . Notons que par l'hypothèse de position générale le degré $d_{\sigma} := |D_{\sigma}|$ de σ est majoré par 5. De plus σ est dans la triangulation canonique d'un échantillon $R \subset L$ si et seulement si $D_{\sigma} \subset R \subset L \setminus L_{\sigma}$. En particulier, le nombre de triangles possédant le même ensemble déterminant est majoré par une constante (la taille maximale de la triangulation canonique de 5 droites). On en déduit

$$|\mathcal{T}(L)| = O(n^5)$$

On note enfin $\mathcal{T}^i(R)$ l'ensemble des triangles σ réalisables sur l'échantillon R tels que $|L_{\sigma} \cap R| = i$.

Théorème 12.4 Soit R un r-échantillon aléatoire pour la loi uniforme sur $\binom{L}{r}$. Alors pour tout $\epsilon > 0$, il existe une constante c_{ϵ} indépendante de r et n telle que :

$$P\left(\max_{\sigma\in\mathcal{T}^{0}(R)}\ell_{\sigma}\leq c_{\epsilon}\frac{n}{r}\log r\right)>1-\epsilon$$

Dit autrement, avec une forte probabilité chacun des triangles de la triangulation canonique de R est coupé (en son intérieur) par $O(\frac{n}{r}\log r)$ droites de L.

Preuve : On note q(c) la probabilité pour qu'il existe un triangle de la triangulation canonique de R qui soit coupé par plus de $c^{\underline{n}}_r \log r$ droites de L. Il suffit donc de montrer que l'on peut choisir c de sorte que $q(c) \leq \epsilon$. On a

$$q(c) = P\Big(\bigvee_{\substack{\sigma \in \mathcal{T}(L)\\ \ell_{\sigma} > c\frac{n}{r}\log r}} \sigma \in \mathcal{T}^{0}(R)\Big) \le \sum_{\substack{\sigma \in \mathcal{T}(L)\\ \ell_{\sigma} > c\frac{n}{r}\log r}} P\Big(\sigma \in \mathcal{T}^{0}(R)\Big)$$

Puisque $\mathcal{T}^0(R) \subset \mathcal{T}(R)$ on peut écrire

$$P(\sigma \in \mathcal{T}^{0}(R)) = P(\sigma \in \mathcal{T}^{0}(R) \mid \sigma \in \mathcal{T}(R)) \cdot P(\sigma \in \mathcal{T}(R))$$

Or

$$P(\sigma \in \mathcal{T}^{0}(R) \mid \sigma \in \mathcal{T}(R)) = \binom{n - d_{\sigma} - \ell_{\sigma}}{r - d_{\sigma}} / \binom{n - d_{\sigma}}{r - d_{\sigma}} = \prod_{i=0}^{r - d_{\sigma} - 1} \frac{n - d_{\sigma} - \ell_{\sigma} - i}{n - d_{\sigma} - i}$$
$$\leq (1 - \frac{\ell_{\sigma}}{n - d_{\sigma}})^{r - d_{\sigma}} \leq e^{-\frac{\ell_{\sigma}}{n - d_{\sigma}}(r - d_{\sigma})}$$

Avec les hypothèses $\ell_{\sigma} > c_{\overline{r}}^n \log r$ et $d_{\sigma} \leq 5$ on en déduit $P(\sigma \in \mathcal{T}^0(R) \mid \sigma \in \mathcal{T}(R)) < r^{-\frac{c}{\ln 2}(1-\frac{5}{r})}$. D'où

$$q(c) \le r^{-\frac{c}{\ln 2}(1-\frac{5}{r})} \sum_{\substack{\sigma \in \mathcal{T}(L)\\ \ell_{\sigma} > c\frac{n}{r}\log r}} P(\sigma \in \mathcal{T}(R)) \le r^{-\frac{c}{\ln 2}(1-\frac{5}{r})} E(|\mathcal{T}(R)|)$$

Mais comme noté plus haut le nombre $|\mathcal{T}(R)|$ de triangles réalisables sur R est un $O(r^5)$. D'où

$$q(c) \le r^{5 - \frac{c}{\ln 2}(1 - \frac{5}{r})}$$

On peut donc choisir c (indépendamment de r et de n) pour rendre cette quantité aussi petite que désirée (notons que pour r petit, disons $r \leq 10$, le théorème est trivialement vrai).

Lemme 12.5 Soit R un r-échantillon de L et soit Q un r/2-échantillon aléatoire de R. Alors pour tout i

$$|\mathcal{T}^{i}(R)| \le c_{i} E(|\mathcal{T}^{0}(Q)|) = O(r^{2})$$

pour une certaine constante $c_i > 0$.

Preuve : C'est évident si $\mathcal{T}^i(R)$ est vide. On pose pour tout $\sigma \in \mathcal{T}^i(R)$,

$$p(\sigma) := P(\sigma \in \mathcal{T}^0(Q)) = P(D_\sigma \subset Q \subset R \setminus L_\sigma)$$

les probabilités étant relatives à la loi uniforme sur $\binom{R}{r/2}$. Puisque $|R \cap L_{\sigma}| = i$, on a

$$p(\sigma) = \binom{r - d_{\sigma} - i}{r/2 - d_{\sigma}} / \binom{r}{r/2}$$

On montre (cf. exercice ci-dessous) que cette dernière quantité est minorée par une constante $k_i > 0$ dès que r est supérieur à une constate K_i ne dépendant que de i. On a alors

$$E(|\mathcal{T}^{0}(Q)|) = \sum_{\sigma \in \mathcal{T}(R)} p(\sigma) \ge \sum_{\sigma \in \mathcal{T}^{i}(R)} p(\sigma) \ge k_{i} |\mathcal{T}^{i}(R)|$$

Ce qui permet de conclure puisque $|\mathcal{T}^0(Q)| = O(r^2)$ d'après le théorème 9.3 sur la complexité des arrangements. Notons que quitte à augmenter c_i , l'inégalité du lemme est inconditionnellement vraie puisque pour $i < r < K_i$, $E(|\mathcal{T}^0(Q)|)$ est uniformément minorée par une constante positive tandis que $|\mathcal{T}^i(R)|$ est uniformément majorée. \Box **Exercice 12.6** Montrep que pour tout $1/2 > \epsilon > 0$ et tout $r \ge \frac{1}{\epsilon} \max(d-1, i-1)$:

$$\binom{r-d-i}{r/2-d} / \binom{r}{r/2} \ge (\frac{1}{2}-\epsilon)^{i+d}$$

Lemme 12.7 Pour tout i on a la majoration suivante pour l'espérance du moment d'ordre i du nombre de droites coupant un triangle de la triangulation canonique d'un réchantillon aléatoire R (pour la loi uniforme) :

$$E(\sum_{\sigma \in \mathcal{T}^0(R)} \binom{\ell_{\sigma}}{i}) = O\left((\frac{n}{r})^i r^2\right)$$

Preuve : Pour tout triangle σ réalisable sur L et pour j = 0 ou j = i on pose $p_j(\sigma) := P(\sigma \in \mathcal{T}^j(R))$. On a ainsi

$$p_0(\sigma) = P(D_{\sigma} \subset R \subset L \setminus L_{\sigma}) = \binom{n - d_{\sigma} - \ell_{\sigma}}{r - d_{\sigma}} / \binom{n}{r}$$

et

$$p_i(\sigma) = P(D_{\sigma} \subset R \text{ et } |L_{\sigma} \cap R| = i) = \binom{\ell_{\sigma}}{i} \binom{n - d_{\sigma} - \ell_{\sigma}}{r - d_{\sigma} - i} / \binom{n}{r}$$

D'où $\binom{\ell_{\sigma}}{i}p_0(\sigma) = p_i(\sigma)\binom{n-d_{\sigma}-\ell_{\sigma}}{r-d_{\sigma}}/\binom{n-d_{\sigma}-\ell_{\sigma}}{r-d_{\sigma}-i}.$

Or dès que r > 2(i+4) on a (cf. exercice ci-dessous) $\binom{n-d_{\sigma}-\ell_{\sigma}}{r-d_{\sigma}} / \binom{n-d_{\sigma}-\ell_{\sigma}}{r-d_{\sigma}-i} < c_i(n/r)^i$ pour une certaine constante c_i . On en déduit

$$E(\sum_{\sigma\in\mathcal{T}^0(R)} \binom{\ell_{\sigma}}{i}) = \sum_{\sigma\in\mathcal{T}(L)} \binom{\ell_{\sigma}}{i} p_0(\sigma) \le c_i(n/r)^i \sum_{\sigma\in\mathcal{T}(L)} p_i(\sigma) \le c_i(n/r)^i E(|\mathcal{T}^i(R)|)$$

Et on conclut avec le lemme précédent.

Exercice 12.8 Montree que pour r > 2(i + d - 1) on a

$$\binom{n-d-\ell}{r-d} / \binom{n-d-\ell}{r-d-i} < 2^{i} (n/r)^{i}$$

Note : Une preuve alternative du théorème 12.4 utilise la notion d' ϵ -net (cf. définition 14.7). En voici une esquisse. On se référera au chapitre 14 pour les définitions et résultats appropriés. On considère le système de parties $(\mathcal{D}, \{\Delta_t\}_t)$ où \mathcal{D} est l'ensemble des droites du plan et où $\{\Delta_t\}_t$ est l'ensemble des parties de \mathcal{D} indexé par les triangles du plan avec Δ_t défini comme le sous-ensemble des droites rencontrant l'intérieur du triangle t. On montre (cf. proposition 14.5) que ce système a une dimension de Vapnik-Chervonenkis finie. On en déduit par le corollaire 14.8 que tout ensemble L de n droites possède un $\frac{\log r}{r}$ -net E de taille O(r) relativement à $(\mathcal{D}, \{\Delta_t\}_t)$. Dit autrement, tout triangle du plan coupé par au moins $\frac{n}{r}\log r$ droites de L contient une droite de E. Par conséquent, les triangles de la triangulation canonique de E sont coupés par au plus $\frac{n}{r}\log r$ droites de L. Le théorème 14.6 montre finalement que E peut être obtenu par tirage aléatoire avec une bonne probabilité.

12.3 Partitions Simpliciales

On rappelle qu'une r-partition (simpliciale) d'un ensemble S de n points du plan est une famille de couples $\{(S_i, \Delta_i)\}_{i \in I}$ où les S_i forment une partition de S et ont chacun une taille comprise entre n/r et 2n/r et où Δ_i est un triangle du plan contenant S_i . Le triangle Δ_i peut éventuellement être dégénéré en un segment, ce qui est utile lorsque Sn'est pas en position générale. Notons que la *taille* |I| de la partition est majorée par r. Son nombre de croisements relativement à un ensemble de droites et le nombre maximal de triangles intersectés (transversalement pour les triangles dégénérés) par l'une de ces droites. On sous-entendra l'ensemble de toutes les droites du plan lorsque cet ensemble n'est pas spécifié. Je suis la présentation de Matoušek [Mat92].

Théorème 12.9 (de la partition simpliciale, Matoušek 1992) Pour tout ensemble S de n points et pour tout r ($2 \le r \le n/2$), il existe une r-partition de S de nombre de croisements $O(\sqrt{r})$ qui peut être construite en temps O(n) si r est majoré par une constante.

La preuve procède en deux étapes. Dans un premier temps on montre que pour tout ensemble de droites fini L, on peut construire une r-partition de S de nombre de croisements $O(\sqrt{r} + \log |L|)$ relativement à L. On montre ensuite qu'on peut construire un ensemble test L_0 composé de O(r) droites tel que le nombre de croisements de *toute* r-partition de S est en gros égal à son nombre de croisements relativement à L_0 .

Lemme 12.10 Soit S, n, r comme dans le théorème et soit L un ensemble fini de droites. On peut construire en temps $O(nr \log r + |L|r^{3/2})$ une r-partition de S de nombre de croisements $O(\sqrt{r} + \log |L|)$ relativement à L.

Preuve : On considère la famille de droites pondérées (L, w_1) avec des poids unitaires; on note $|w_1| = |L|$ son poids total. On pose $\Sigma_1 = S$ et $r_1 = r$. Par le corollaire 12.3 on peut construire un $\frac{1}{c\sqrt{r_1}}$ -cutting de (L, w_1) , pour une constante c appropriée, possédant au plus r_1 faces (triangles et arêtes) au total. En particulier, le poids total des droites de L coupant un triangle de ce cutting est majoré par $O(|w_1|/\sqrt{r_1})$. Il en est de même pour les arêtes, puisque toute droite qui coupe une arête coupe ses triangles incidents. L'une des faces Δ_1 du cutting contient au moins $|\Sigma_1|/r_1 = n/r$ sommets de Σ_1 . On choisit arbitrairement un sous-ensemble S_1 de $\lceil n/r \rceil$ sommets dans Δ_1 , ce qui fournit la première paire (S_1, Δ_1) de la r-partition. On pose ensuite $\Sigma_2 = \Sigma_1 \setminus S_1$, $r_2 = |\Sigma_2|r/n$ et on définit des poids w_2 pour L en doublant les poids des droites de L coupant Δ_1 et en laissant inchangé les autres poids. Si $|\Sigma_2| \leq 2n/r$, on pose $S_2 = \Sigma_2$ et $\Delta_2 = \mathbb{R}^2$. Sinon on itère le procédé avec Σ_2 , r_2 et (L, w_2) . Au bout de $m \leq r$ étapes, on obtient une r-partition simpliciale de S :

$$(S_1, \Delta_1), (S_2, \Delta_2), \ldots, (S_m, \Delta_m)$$

On considère une droite $\ell \in L$. Soit k le nombre de triangles de la r-partition coupés par ℓ . On a ainsi $w_{m+1}(\ell) = 2^k$. Par ailleurs, si p est le poids total des droites de L coupant Δ_i à l'étape i on a d'une part $p = O(|w_i|/\sqrt{r_i})$, par les propriétés du cutting utilisé, et d'autre part $|w_{i+1}| = (|w_i| - p) + 2p$, par la règle de doublement des poids. D'où $|w_{i+1}| = |w_i|(1 + O(\frac{1}{\sqrt{r_i}}))$, et partant de $|w_1| = |L|$:

$$|w_{m+1}| = |w_1| \prod_{i=1}^m (1 + O(\frac{1}{\sqrt{r_i}})) \le |L| e^{O(\sum_{i=1}^m \frac{1}{\sqrt{r_i}})}$$

Cette dernière relation provenant de l'inégalité $1 + x \leq e^x$. Or,

$$r_i = |\Sigma_i| r/n = (n - (i - 1) \lceil n/r \rceil) r/n \ge r - i + 1$$

Il suit que $\sum_{i=1}^{m} \frac{1}{\sqrt{r_i}} \leq \sum_{i=1}^{r} \frac{1}{\sqrt{i}} \leq 2\sqrt{r}$ en majorant la somme par une intégrale. On déduit finalement de $w_{m+1}(\ell) \leq |w_{m+1}|$ la majoration cherchée : $k = O(\log |L| + \sqrt{r})$.

Le temps requis à chaque étape est $O(|L|\sqrt{r_i})$ pour la construction du cutting plus $O(n_i \log r_i)$ pour sélectionner les points de S_i en utilisant par exemple une structure de recherche pour le cutting telle que décrite section 10.2. Le temps total de construction de la *r*-partition est donc $O(|L|r^{3/2} + nr \log r)$.

Lemme 12.11 (de l'ensemble test) Soient S, n, r comme dans le théorème. Il existe un ensemble L de O(r) droites tel que le nombre de croisements de toute r-partition \mathcal{P} de S est majoré par

$$3c_L + \sqrt{r}$$

où c_L est le nombre de croisements de \mathcal{P} relativement à L. De plus, un tel ensemble L peut être construit en temps $O(n\sqrt{r})$.

Preuve : On note * la dualité point/droite comme à la section 11.2.2. Soit S^* l'ensemble des droites duales des points de S. Par le théorème 12.1, on peut construire en temps $O(n\sqrt{r})$ un $\frac{1}{\sqrt{r}}$ -cutting de S^* constitué de O(r) triangles, arêtes et sommets. On définit L comme les droites duales des O(r) sommets de ce cutting. Reste à vérifier que L a la propriété désirée. Soient \mathcal{P} et c_L comme dans le lemme et soit ℓ une droite du plan. On considère le triangle Δ du cutting contenant le point dual ℓ^* et on note D l'ensemble des 3 droites duales des sommets de Δ . ⁴ Soit c le nombre de triangles de \mathcal{P} coupés par ℓ et par au moins l'une des droites de D. Par définition de c_L , on a $c \leq 3c_L$. On note c' le nombre de triangles de \mathcal{P} coupés par ℓ mais par aucune des droites de D. Ces triangles contiennent au moins c'n/r sommets de S et sont contenus dans la zone de ℓ dans l'arrangement des droites de D. Il suit aisément (exercice!) que les droites duales de ces sommets coupent l'intérieur de Δ . Par le choix du cutting, leur nombre est majoré par n/\sqrt{r} . D'où $c' \leq \sqrt{r}$. \Box

Exercice 12.12 Soit Δ un triangle généralisé du plan et p un point intérieur à Δ . Soit D l'ensemble des 3 droites duales des sommets de Δ . Ces droites peuvent être verticales si Δ est non borné (cf. note de la preuve ci-dessus). Montrer que tout point de la zone de p^{*} dans l'arrangement de D se dualise en une droite coupant Δ .

^{4.} On étend la dualité aux points à l'infini (= les directions de plan) en associant la droite verticale $\{x = a\}$ à la direction de pente a. Si Δ est non-borné ses points à l'infini sont les directions des côtés non-bornés.

Supposons que $\Delta = [d_1^*, d_2^*, d_3^*]$ est un triangle borné. Un point d^* est dans $\Delta = [d_1^*, d_2^*, d_3^*]$ si et seulement s'il est disons au dessus des droites $\ell_{12} = d_1^* d_2^*$ et $\ell_{31} = d_3^* d_1^*$ et au dessous de la droite $\ell_{23} = d_2^* d_3^*$. Ce qui équivaut à ℓ_{12}^* et ℓ_{31}^* sont au dessus de la droite d et ℓ_{23}^* est au dessous de d. Par tout point p de la zone de ℓ dans D, il passe une droite une telle droite d. Donc p^* , qui contient d^* , coupe Δ . Les cas où Δ est non borné sont similaires.

Preuve du théorème 12.9 : Par le lemme 12.11, on construit en temps $O(n\sqrt{r})$ un ensemble L de O(r) droites tests pour les r-partitions de S. Par le lemme 12.10, on construit en temps $O(nr \log r + r^{5/2})$ une r-partition de S de nombre de croisements $O(\sqrt{r})$ relativement à L et qui a donc un nombre de croisements du même ordre relativement à toutes les droites du plan d'après le lemme 12.11.

Remarquons que pour r non constant, la preuve du théorème fournit une construction d'une r-partition en temps $O(nr \log r + r^{5/2})$. Il est possible, quitte à augmenter légèrement le nombre de croisements (en $O(r^{1/2+\varepsilon})$), de réduire ce temps à $O(n \log r)$ (cf. [Mat92]). L'idée est de fixer une constante r_0 , puis de calculer une r_0 -partition $\{(S_i, \Delta_i)\}_i$ pour S, puis une r_0^2 -partition pour chaque S_i , et de continuer récursivement $\log_{r_0} r$ fois.

Chapitre 13

Algorithmes randomisés incrémentaux

13.1 Introduction

Etant donnés un ensemble X d'objets (ex : des points du plan, des droites du plan, \ldots), on s'intéresse à la construction d'une certaine structure géométrique $\mathcal{S}(X)$ associée à cet ensemble (ex : l'enveloppe convexe des points, l'arrangement des droites,...). Le principe d'un algorithme incrémental est de construire $\mathcal{S}(X)$ récursivement en ajoutant les objets un à un. Plus précisément, si on ordonne les éléments x_1, \ldots, x_n de X, on construit récursivement $\mathcal{S}_i = \mathcal{S}(\{x_1, \ldots, x_i\})$ à partir de \mathcal{S}_{i-1} en insérant x_i dans la structure. Un algorithme incrémental est dit randomisé lorsque l'ordre sur les éléments de X est aléatoire. On peut alors considérer l'algorithme comme une variable aléatoire sur l'espace des permutations de X et estimer l'espérance de son coût en temps ou espace. Un tel algorithme est dit de type Las Veqas au sens où l'exécution est aléatoire mais pas le résultat final du calcul. La randomisation permet généralement d'obtenir des algorithmes plus simples que dans le cas déterministe, tout en conservant des complexités en moyenne aussi faibles (voire plus faibles) que les complexités dans le cas le pire des algorithmes déterministes. Le modèle probabiliste, c'est à dire le choix de la probabilité sur les permutations de X, est en général le modèle uniforme. On s'en tiendra ici à ce modèle. En particulier, l'ensemble $\{x_1, \ldots, x_i\}$ des *i* premiers objets constitue un *i*-échantillon aléatoire de X dont la probabilité de réalisation est $1/\binom{n}{i}$. Quicksort est l'archétype des algorithmes randomisés incrémentaux.

13.2 Le formalisme

Il se trouve que la plupart des structures classiques rencontrées en géométrie algorithmique, telles que les enveloppes convexes de points, les arrangements d'hyperplans, les diagrammes de Voronoi, les décompositions trapézoïdales de segments du plan, etc... peuvent être construites selon des algorithmes randomisés incrémentaux. Ses algorithmes possèdent tous les mêmes caractéristiques et peuvent être analysés de manière analogue. Un formalisme abstrait a ainsi pu être développé pour les algorithmes randomisés incrémentaux. Ce formalisme, introduit par Clarkson et Shor [CS89], s'applique aux différentes structures citées plus haut et évite des analyses spécifiques à chaque structure particulière.

De manière générale, on considère un univers d'objets et un ensemble de configurations. Par exemple, pour le problème de l'enveloppe convexe de points de \mathbb{R}^d , les objets sont les points de \mathbb{R}^d et les configurations correspondent intuitivement (cf. note ci-après) aux demi-espaces de \mathbb{R}^d . À toute configuration on associe un sous-ensemble d'objets et on dit que la configuration est déterminée par ce sous-ensemble. Dans l'exemple qui précède un sous-ensemble de points détermine un demi-espace si son enveloppe affine est précisément l'hyperplan qui borde ce demi-espace. Il faut ainsi noter qu'une configuration n'est pas à proprement parler un demi-espace mais plutôt la conjonction d'un ensemble de points dont l'enveloppe affine est un hyperplan et d'une orientation qui distingue l'un des deux demi-espaces bordés par cet hyperplan. Une configuration admet également un certain sous-ensemble d'objets, disjoint de l'ensemble qui la détermine, en *conflit* avec elle. Toujours pour le même exemple, un point est en conflit avec tout "demi-espace" qui le contient en son intérieur.

Étant donné un sous-ensemble fini X d'objets, on souhaite construire une certaine structure $\mathcal{S}(X)$ formée de l'ensemble des configurations déterminées par des sous-ensembles de X et sans conflit avec les objets de X. On note $\mathcal{C}^0(X)$ cet ensemble de configurations. Dans le cas du problème de l'enveloppe convexe, X est un ensemble de points de \mathbb{R}^d et $\mathcal{S}(X)$ est l'enveloppe convexe de ces points. Cette enveloppe est constituée d'un ensemble de facettes dont les hyperplans supports sont précisément les bords des demi-espaces s'appuyant sur des points de X et ne contenant aucun point en leur intérieur, c'est à dire des demi-espaces de $\mathcal{C}^0(X)$. On voit ainsi que, sous une hypothèse de position générale sur les objets de X, les éléments de la structure $\mathcal{S}(X)$ (ici les facettes) sont en bijection avec les configurations de $\mathcal{C}^0(X)$. Généralement, la structure $\mathcal{S}(X)$ que l'on souhaite construire est un peu plus riche que la simple collection des configurations de $\mathcal{C}^0(X)$. Dans notre exemple, on peut également vouloir calculer les adjacences entre facettes. Dans la plupart des cas rencontrés $\mathcal{S}(X)$ est simplement un graphe construit sur $\mathcal{C}^0(X)$.

Il est à noter que le formalisme exact des espaces de configurations n'est pas totalement fixé dans la littérature. On pourra comparer Clarkson [CS89], Mulmuley [Mul00] et Matoušek [SU00, chap. 13]. En particulier, la terminologie précédente varie d'un auteur à l'autre. Ainsi, une configuration est parfois appelée région ou domaine. L'ensemble déterminant une configuration est aussi son ensemble de définition ou d'adjacences (ou encore de 'triggers' en anglais) tandis qu'un objet en conflit avec une configuration est dit tuer ou stopper cette configuration. Les configurations déterminées par des sous-ensembles de X sont dites réalisables sur X. Les configurations réalisables et sans conflit sur X, i.e. de $C^0(X)$, sont dites actives sur X.

Voici d'autres exemples importants où le formalisme précédent s'applique.

Exemple 13.1 (Arrangement d'hyperplans) Les objets sont les hyperplans de \mathbb{R}^d . Un arrangement d'une famille d'hyperplans est constitué des cellules convexes de la subdivision de \mathbb{R}^d induite par les hyperplans de cette famille. On serait donc tenté de définir une configuration comme une cellule convexe déterminée par les hyperplans qui la borde. Mais ce nombre d'hyperplans n'est à priori pas borné. Or, pour des raisons d'efficacité algorithmique, il est important que les configurations réalisables soient déterminées par un nombre borné d'objets. Pour remédier à ce problème on enrichit l'arrangement par sa triangulation. Ceci peut être fait de manière canonique en triangulant récursivement les faces de l'arrangement par ordre croissant sur leur dimension; chaque face est triangulée par étoilement à partir de son sommet de coordonnées minimales pour l'ordre lexicographique¹. Ceci amène à définir une configuration comme un d-simplexe. Plus précisément, une configuration est un couple (σ , H) formé d'un d-simplexe σ et d'un ensemble H d'hyperplans incidents à σ (en fait à son bord) tels que σ est dans la triangulation canonique de l'arrangement de H. Une telle configuration est dite déterminée par H et en conflit avec les hyperplans qui rencontrent l'intérieur de σ . On voit maintenant que les d-simplexes de la triangulation canonique d'un arrangement d'hyperplans en position générale sont en bijection avec les configurations actives de l'ensemble des hyperplans de l'arrangement. L'hypothèse de position générale est aussi importante pour borner la taille des ensembles déterminant les configurations réalisables.

Exemple 13.2 (Décomposition trapézoïdale de segments du plan) Les objets sont les segments du plan. La structure est la décomposition trapézoïdale obtenue par cloisonnement vertical. Une configuration est un couple (τ, S) où τ est un trapèze du plan et Sun ensemble de segments incidents à τ et tel que τ soit l'un des trapèzes de la décomposition trapézoïdale de S. Cette configuration est déterminée par S et en conflit avec les segments qui intersectent l'intérieur de τ . Sous une hypothèse simple de position générale sur une famille de segments, les trapèzes de la décomposition trapézoïdale de cette famille sont en bijection avec les configurations actives sur cette famille.

Exemple 13.3 (Intersection de demi-espaces) Les objets sont les demi-espaces de \mathbb{R}^d . Une configuration est un couple (p, D) formé d'un point p de \mathbb{R}^d et d'un ensemble de demi-espaces dont l'intersection des hyperplans bordant est p. Une telle configuration est dite déterminée par D et en conflit avec tout demi-espace qui ne contient pas p. Sous une hypothèse de position générale, les sommets du polytope intersection d'une famille de demi-espaces sont en bijection avec les configurations actives sur cette famille.

Exemple 13.4 (Diagramme de Voronoi du plan) Les objets sont les points du plan (appelés sites). Pour des raisons analogues à l'emploi d'une triangulation canonique d'un arrangement d'hyperplans (cf. exemple 13.1) on considère la triangulation radiale du diagramme de Voronoi de tout sous-ensemble fini de sites. Elle est obtenue en étoilant chaque cellule du diagramme par rapport à son site. Ce qui amène à définir une configuration comme une paire (t, S) où t est un triangle et S un ensemble de sites tels que t apparaît dans la triangulation radiale du diagramme de Voronoi de S et que toute cellule de ce diagramme rencontre (la clôture de) t. Une telle configuration est dite déterminée par S et en conflit avec tout point p tel que t ne soit pas un triangle de la triangulation du diagramme de Voronoi de $S \cup \{p\}$, i.e. tel qu'il existe un point de t qui soit plus proche de p que des points de S.

Exemple 13.5 (Diagramme de Delaunay du plan) Les objets sont les points du plan. Les configurations sont les triangles du plan. Un triangle est déterminé par ses sommets

^{1.} En fait, puisque l'arrangement comporte des faces non bornées, le bon cadre est de plonger \mathbb{R}^d de manière canonique dans l'espace projectif, puis de définir un ordre total sur l'espace projectif qui étende l'ordre lexicographique.

et en conflit avec tout point intérieur à son cercle circonscrit. Sous une hypothèse de position générale, les triangles de la triangulation de Delaunay d'un ensemble fini X de points sont les triangles actifs sur X.

Exercice 13.6 Explicitez dans chacun des exemples ci-dessus la condition de position générale sur les objets de X afin que chaque configuration soit déterminée par un nombre borné d'objets, indépendant de la taille de X. Explicitez ces bornes.

On note $\mathcal{P}(E)$ l'ensemble des parties d'un ensemble E.

Définition 13.7 Un espace de configurations est un quadruplet $(\mathcal{O}, \mathcal{C}, \delta, \kappa)$ où

- 1. O est un ensemble (on dit aussi univers) dont les éléments sont appelés objets,
- 2. C est un ensemble, dont les éléments sont appelés configurations,
- 3. $\delta : \mathcal{C} \to \mathcal{P}(O)$ associe à chaque configuration un sous-ensemble d'objets. Une configuration c est dite déterminée par les objets de $\delta(c)$.
- 4. $\kappa : \mathcal{C} \to \mathcal{P}(O)$ associe à chaque configuration un sous-ensemble d'objets appelé son ensemble de conflits. Une configuration c est dite en conflit avec tout objet de $\kappa(c)$.
- 5. $\forall c \in \mathcal{C} : \delta(c) \cap \kappa(c) = \emptyset$.

On appelle degré d'un espace de configurations $(\mathcal{O}, \mathcal{C}, \delta, \kappa)$ le cardinal maximal de $\delta(c)$ pour toute configuration c.

Soit X une partie de \mathcal{O} . L'espace de configurations $(\mathcal{O}, \mathcal{C}, \delta, \kappa)$ induit sur X un espace de configurations $(X, \mathcal{C}(X), \delta_X, \kappa_X)$ où

- C(X) est l'ensemble des configurations de C déterminées par des parties de X (i.e. telles que $\delta(c) \subset X$),
- δ_X est la restriction de δ à $\mathcal{C}(X)$,

- et $\kappa_X : \mathcal{C}(X) \to \mathcal{P}(X)$ est définie par $\kappa_X(c) = \kappa(c) \cap X$.

Une configuration $c \in C$ est dite active sur X si $c \in C(X)$ et $\kappa(c) \cap X$ est vide. On note $C^0(X)$ l'ensemble des configurations actives sur X.

13.3 Algorithmes statiques

13.3.1 Formalisme et analyse randomisée

Un algorithme est dit statique ou hors ligne s'il s'applique à des données connues à l'avance. Dit autrement, les données sont chargées en mémoire une fois pour toute et ne seront plus modifiées. Relativement au formalisme des espaces de configurations, cela signifie qu'un ensemble fini X d'objets d'un univers \mathcal{O} est donné, et que l'on cherche à construire la structure $\mathcal{S}(X)$ et donc en particulier les configurations actives de l'espace de configurations induit sur X. On notera $(X, \mathcal{C}(X), \delta, \kappa)$ cet espace de configurations.

Un algorithme incrémental et randomisé fonctionne de la manière suivante :

(1) On construit une permutation aléatoire $(x_1, x_2, ..., x_n)$ des éléments de X. On pose $X_i = \{x_1, x_2, ..., x_i\}.$

(2) On construit récursivement $\mathcal{S}(X_{i+1})$ (et donc l'ensemble des configurations actives sur X_{i+1}) en ajoutant x_{i+1} à $\mathcal{S}(X_i)$. Pour cela on doit supprimer les configurations actives à l'étape i (de $\mathcal{C}^0(X_i)$) en conflit avec x_{i+1} et ajouter les configurations *activées* par x_{i+1} , c'est-à-dire les configurations $c \in \mathcal{C}^0(X_{i+1})$ telles que $x_{i+1} \in \delta(c)$.

Pour faciliter la construction récursive, on introduit le graphe des conflits \mathcal{G}_i entre les configurations actives courantes et les objets non-traités, défini par

$$\mathcal{G}_i = \left\{ (c, x) : c \in \mathcal{C}^0(X_i) \land x \in \kappa(c) \right\}$$

En particulier, $\mathcal{G}_i \subset \mathcal{C}^0(X_i) \times (X \setminus X_i)$ est un graphe biparti. On appellera *liste d'adjacences* ou *liste de conflits* d'un noeud de \mathcal{G}_i , la liste des voisins de ce noeud dans \mathcal{G}_i . On note $\ell_i(c)$ (resp. $\ell_i(x)$) la liste de conflits de la configuration c (resp. de l'objet x) dans \mathcal{G}_i . En particulier, $\ell_i(c) = \kappa(c)$. Ces listes permettent de retrouver rapidement les configurations actives de la *i*-ème étape qui vont disparaître suite à l'ajout de x_{i+1} : il s'agit précisément des configurations de $\ell_i(x_{i+1})$.

Le caractère statique des données permet de maintenir, en plus de la structure $S(X_i)$, tout ou partie de ce graphe des conflits au cours de l'algorithme. L'objectif est de pouvoir reconstituer rapidement la liste $\ell_i(x_{i+1})$. Les détails propres au maintien (d'une partie) du graphe des conflits et de la structure $S(X_i)$ dépendent du problème spécifique à traiter (décomposition trapézoïdale, enveloppe convexe, etc...). On peut néanmoins analyser de manière générique un algorithme incrémental randomisé et statique en supposant que le coût des diverses mises à jour à l'étape *i* vérifie

la condition de mise à jour statique : le coût total de l'étape i lors de l'ajout de x_i est proportionnel au nombre de différences entre \mathcal{G}_{i-1} et \mathcal{G}_i .

Ce nombre de différences inclue

- 1. le nombre a_i de configurations activées par x_i , i.e. de $\mathcal{C}^0(X_i) \setminus \mathcal{C}^0(X_{i-1})$,
- 2. le nombre a'_i de conflits activés avec ces configurations (ce sont les nouvelles arêtes de \mathcal{G}_i qui n'étaient pas dans \mathcal{G}_{i-1}),
- 3. le nombre t_{i-1} de configurations tuées par x_i , i.e. de $\mathcal{C}^0(X_{i-1}) \setminus \mathcal{C}^0(X_i)$,
- 4. le nombre t'_{i-1} de conflits tués avec ces configurations (ce sont les arêtes de \mathcal{G}_{i-1} qui ne sont plus dans \mathcal{G}_i).

Puisque toute configuration ou tout conflit est nécessairement activé avant d'être tué et ne peut être tué qu'une fois au plus, on a compte tenu de $t_0 = t'_0 = t_n = t'_n = 0$:

$$\forall j \in [1,n] : \sum_{i=1}^{j} t_i \le \sum_{i=1}^{j} a_i \text{ et } \sum_{i=1}^{j} t'_i \le \sum_{i=1}^{j} a'_i.$$
(13.1)

Il en résulte que sous la condition de mise à jour statique, la complexité d'un algorithme incrémental statique est proportionnelle au nombre total de configurations et conflits activés au cours de l'algorithme. Pour tenir compte de l'aspect randomisé, on analyse la complexité en moyenne, relativement à la distribution uniforme sur les permutations des objets de X.

Lemme 13.8 Soit un espace de configurations $(X, \mathcal{C}(X), \delta, \kappa)$ de n = |X| objets et de degré $d = \max_{c \in \mathcal{C}(X)} |\delta(c)|$. On note $e_i = E(|\mathcal{C}^0(X_i)|)$ l'espérance du nombre de configurations activées à l'étape i. L'espérance du nombre total de configurations activées au cours de l'algorithme est majorée par

$$d\sum_{i=1}^{n} \frac{e_i}{i}$$

L'espérance du nombre total de conflits activés au cours de l'algorithme est majorée par

$$d^2 \sum_{i=1}^n \frac{n-i}{i^2} e_i$$

Preuve de la majoration de l'espérance du nombre de configurations actives : On pose $A_i = C^0(X_i) \setminus C^0(X_{i-1})$ l'ensemble des configurations activées à l'étape *i* et $a_i = |A_i|$. Fixons un sous-ensemble $X_i \subset X$ de *i* objets. L'évènement $\{X_i \text{ fixé }\}$ désigne l'ensemble des permutations de X dont l'*ensemble* des *i* premiers éléments coïncide avec X_i . On a ainsi

$$E(a_i \mid X_i \text{ fixé}) = \sum_{c \in \mathcal{C}^0(X_i)} P(c \in A_i \mid X_i \text{ fixé}) = \sum_{c \in \mathcal{C}^0(X_i)} P(x_i \in \delta(c) \mid X_i \text{ fixé})$$

La dernière égalité traduisant le fait qu'une configuration active à l'étape i ne l'était pas à l'étape i - 1 si et seulement si $x_i \in \delta(c)$. Or $P(x_i \in \delta(c) \mid X_i \text{ fixe}) \leq d/i$ car chaque objet de X_i a la même probabilité 1/i de se retrouver en *i*-ème position et, par définition du degré, au plus d objets sont dans $\delta(c)$. D'où

$$E(a_i \mid X_i \text{ fixé}) \le \frac{d}{i} |\mathcal{C}^0(X_i)|$$

et le lemme 1.18 permet de conclure.

Preuve de la majoration de l'espérance du nombre de conflits activés : On garde les notations précédentes et on note a'_i le nombre de conflits activés à l'étape i. On a ainsi

$$a'_i = \sum_{c \in A_i} |\kappa(c)| = \sum_{j > i} a^j_i$$

où $a_i^j = |A_i \cap \{c \in \mathcal{C}(X) : x_j \in \kappa(c)\}|$ désigne le nombre de configurations activées à l'étape *i* et en conflit avec x_j . Puisque x_j prend toutes les valeurs de $X \setminus X_i$ de manière équiprobable, l'espérance $E(a_i^j)$ ne dépend pas de *j* (cf. exercice 13.9 ci-dessous), i.e. $E(a_i^j) = E(a_i^{i+1})$ pour j > i. D'où

$$E(a'_i) = (n-i)E(a^{i+1}_i)$$

On note $T_i = \mathcal{C}^0(X_i) \setminus \mathcal{C}^0(X_{i+1})$ l'ensemble des configurations actives tuées à l'étape i+1. Comme $a_i^{i+1} = |A_i \cap T_i|$, on a

$$E(a_i^{i+1}) = \sum_{c \in \mathcal{C}(X)} P(c \in A_i \land c \in T_i) = \sum_{c \in \mathcal{C}(X)} P(c \in A_i \mid c \in T_i) P(c \in T_i)$$

Soit X_i un sous-ensemble de *i* objets de X et $x \in X \setminus X_i$. On note $B(X_i, x)$ l'ensemble des permutations de X dont l'ensemble des *i* premiers éléments est X_i et dont le i + 1-ième élément est x. On a égalité entre les événements $\{c \in T_i\}$ et $\bigcup B(X_i, x)$, l'union portant sur les (X_i, x) tels que $c \in C^0(X_i)$ et $x \in \kappa(c)$. Or $P(c \in A_i \mid B(X_i, x)) \leq d/i$ par un raisonnement analogue à la preuve précédente. D'où $P(c \in A_i \mid c \in T_i) \leq d/i$ (cf. exercice 1.21). On en déduit

$$E(a_i') \le d\frac{n-i}{i}E(t_i)$$

où l'on a posé $t_i = |T_i|$. Par conséquent $E(\sum_i a'_i) \leq dE(\sum_i \frac{n-i}{i}t_i)$. On vérifie aisément que la décroissance de la fonction $i \mapsto \frac{n-i}{i}$ et les inégalités (13.1) impliquent (cf. exercice 13.10 ci-dessous)

$$E(\sum_{i} a'_{i}) \le dE(\sum_{i} \frac{n-i}{i}a_{i})$$

La première majoration du lemme sur l'espérance du nombre a_i de configurations activées à l'étape i permet de conclure.

Exercice 13.9 Soient X un ensemble de n objets, $k \in \mathbb{N}$ et $f: X^{k+1} \to \mathbb{R}$. On fixe un k-uplet d'indices distincts $I = (i_1, \ldots, i_k) \in [1, n]^k$ et on définit n - k variables aléatoires $\{f_j\}_{j \notin I}$ sur l'ensemble S_X des permutations de X (muni de la probabilité uniforme) par

 $\forall j \in [1, n] \setminus I, \forall \sigma = (x_1, \dots, x_n) \in \mathcal{S}_X : \qquad f_j(\sigma) = f(x_{i_1}, \dots, x_{i_k}, x_j)$

Montrer que $E(f_i)$ ne dépend pas de j.

Exercice 13.10 Soient deux familles de n nombres $(a_i)_{1 \le i \le n}$ et $(t_i)_{1 \le i \le n}$ telles que

$$\forall j \in [1, n], \qquad \sum_{i=1}^{j} t_i \le \sum_{i=1}^{j} a_i$$

et soit $(\alpha_i)_{1 \leq i \leq n}$ une suite positive décroissante. Montrer que

$$\sum_{i=1}^{n} \alpha_i t_i \le \sum_{i=1}^{n} \alpha_i a_i$$

Références

- L'état de l'art [Mul00] et le livre [Mul94] de Mulmuley.
- Les articles fondateurs [CS89, CMS93].
- Le livre de Boissonnat et Yvinec [BY95].

La notion de graphe d'historique (ou d'influence) permet d'étendre le formalisme des algorithmes randomisés statiques (ou hors-ligne) aux algorithmes semi-dynamiques (ou en ligne) ou dynamiques. La version semi-dynamique permet d'ajouter au cours du temps des objets non-spécifiés au début de l'algorithme, ce qui n'est pas compatible avec la notion de graphe des conflits. Un exemple est fourni par l'algorithme de décomposition trapézoïdale de la section 10.2. Dans cet algorithme le graphe d'historique est donné par la structure \mathcal{R} de recherche. Un algorithme dynamique gère également la suppression d'objets au cours du temps. Les références ci-dessus développent ces aspects.
13.3.2 Applications

On montre ici comment appliquer le formalisme précédent à des exemples de la section 13.2.

Décomposition trapézoïdale

On reprend le cadre de l'exemple 13.2. Soit une famille $X = \{x_1, x_2, \dots, x_n\}$ de *n* segments du plan telle que deux segments sont soit disjoints soit s'intersectent en une extrémité commune. On considère ici que $\mathcal{S}(X)$ est un graphe d'adjacence sur $\mathcal{C}^0(X)$ où deux configurations sont adjacentes si les trapèzes correspondants partagent une paroi verticale. On note que sous une hypothèse raisonnable de position générale, le degré de l'espace de configurations est majoré par 4. Compte tenu de la taille linéaire de la décomposition trapézoïdale (i.e. de chaque $\mathcal{S}(X_i)$), le lemme 13.8 montre que le nombre moyen de configurations activées au cours de l'algorithme est en O(n) et que le nombre moyen de conflits activés est un $O(n \log n)$: il suffit de remplacer e_i par O(i) dans les formules du lemme. Afin d'en déduire la même complexité pour tout l'algorithme de décomposition il faut valider la condition de mise à jour statique. En plus de la décomposition trapézoïdale $\mathcal{S}(X_i)$, on maintient ici le graphe des conflits \mathcal{G}_i dans sa totalité. Ce graphe est codé sous forme de listes d'adjacences (de conflits) augmentées de pointeurs bidirectionnels, de sorte que si le trapèze τ est en conflit avec le segment x, l'occurrence de x dans $\ell_i(\tau)$ pointe vers l'occurrence de τ dans $\ell_i(x)$ et vice versa. Voyons les opérations à effectuer pour mettre à jour $\mathcal{S}(X_{i-1})$ et \mathcal{G}_{i-1} suite à l'ajout du segment x_i . Il faut

- Supprimer les trapèzes courants en conflit avec x_i , c.-à-d. traversés par x_i , et supprimer les listes de conflits de ces trapèzes. Comme noté dans la section 13.3.1, ces trapèzes sont précisément ceux de la liste de conflits $\ell_{i-1}(x_i)$.
- Ajouter les trapèzes activés par x_i et les nouvelles relations d'adjacences liés à ces trapèzes et créer les listes de conflits pour ces trapèzes.

Pour effectuer ces opérations, on procède en deux étapes. Dans une première étape on se limite à subdiviser (en au plus quatre morceaux) chaque trapèze courant traversé par x_i ou par les parois verticales aux extrémités de x_i , quitte à conserver des parois verticales inutiles (typiquement toute paroi verticale courante coupée par x_i est scindée en deux parois dont une seule doit être conservée). On modifie $S(X_{i-1})$ et \mathcal{G}_{i-1} en accord avec ces subdivisions. Pour cela on remplace chaque trapèze τ qui doit être subdivisé par les (au plus quatre) trapèzes qui le partitionnent et on répartie la liste de conflits $\ell_{i-1}(\tau) = \kappa(\tau)$ entre ces trapèzes. Ceci prend un temps $O(|\kappa(\tau)|)$. De plus, pour chaque segment $x \in \kappa(\tau)$, on remplace τ dans $\ell_{i-1}(x)$ par la sous-liste (de longueur trois au plus) ordonnée de gauche à droite des nouveaux trapèzes qui partitionnent τ et qui sont traversés par x. Ce qui prend à nouveau un temps $O(|\kappa(\tau)|)$. La condition sur l'ordre des trapèzes lors du remplacement permet de supposer que les trapèzes de la liste de conflits de tout segment sont ordonnés de gauche à droite le long du segment. Notons que les relations d'adjacences entre et avec les nouveaux trapèzes s'obtiennent aisément en temps proportionnel à leur nombre.

Dans une seconde étape, on supprime les morceaux de parois verticales inutiles, c.-à-d. qui ne passent pas par une extrémité de segment. Ceci revient à fusionner certain des trapèzes introduits à la première étape. Si l'on doit par exemple fusionner les trapèzes



FIGURE 13.1 – Mise à jour de l'enveloppe convexe $S(X_{i-1})$ en dimension d = 2 lors de l'insertion de x_i . Sur la figure $f_{i-1}(x) = f_{i-1}(y) = c$ tandis que $f_{i-1}(z) = NULL$.

 $\tau_1, \tau_2, \ldots, \tau_k$ en un trapèze τ' il faut fusionner les listes de conflits $\ell(\tau_1), \ell(\tau_2), \ldots, \ell(\tau_k)$ en une liste $\ell(\tau')$ et remplacer chaque τ_j par τ' dans toutes les listes $\ell(x)$ pour $x \in \ell(\tau_j)$. La difficulté consiste à supprimer en temps linéaire les répétitions lors de la fusion des listes et lors des remplacements. On utilise pour cela la propriété d'ordonnancement de gauche à droite des listes de conflits des segments. Plus précisément, on crée une liste $\ell(\tau')$ vide au départ. Puis, on parcourt chaque $\ell(\tau_j)$. Pour $x \in \ell(\tau_j)$, on place x dans $\ell(\tau')$ et on remplace dans $\ell(x)$ la sous-liste ℓ' de $\tau_1, \tau_2, \ldots, \tau_k$ (contenant τ_j) par τ' en temps proportionnel à $|\ell'|$. On supprime de plus x des listes de conflits des trapèzes de la sousliste ℓ' . Ces suppressions évitent les répétitions dans $\ell(\tau')$. Le temps total pris pour cette seconde étape est clairement proportionnel à la somme des tailles des listes de conflits des trapèzes créés à la première étape. Et cette somme est elle-même proportionnel à la taille des listes de conflits tuées par x_i . On valide ainsi la condition de mise à jour statique.

Enveloppe convexe en dimension d = 2 ou 3

On rappelle que pour le problème de l'enveloppe convexe les objets sont les points de \mathbb{R}^d et les configurations sont les paires (S, h) où h est un demi-espace et S est un ensemble de points tels que ∂h est l'enveloppe affine de S. Une configuration (S, h) est déterminée par S et en conflit avec les points contenus dans l'intérieur de h.

Soit alors une famille $X \subset \mathbb{R}^d$ de *n* points en position générale, c.-à-d. telle que toute sousfamille de d+1 points est affinement indépendante. On a vu que les hyperplans supports des facettes de l'enveloppe convexe de X correspondent aux régions actives sur X. Plus précisément on considère ici que $\mathcal{S}(X)$ est un graphe d'adjacence sur $\mathcal{C}^0(X)$ où deux configurations sont adjacentes si les facettes correspondantes partagent une (d-2)-face. En plus de l'enveloppe convexe $\mathcal{S}(X_i)$ on se contente ici de maintenir un petit sous-graphe du graphe des conflits \mathcal{G}_i . Plus précisément on maintient pour chaque sommet $x \in X \setminus X_i$, une facette $f_i(x)$ de $\mathcal{S}(X_i)$ en conflit avec x (donc $f_i(x) \in \ell_i(x)$). On pose $f_i(x) = NULL$ si x est intérieur à $\mathcal{S}(X_i)$ (cf. figure 13.1). On maintient également pour chaque facette active, c, une liste de conflits $\ell'_i(c) \subset \ell_i(c)$ restreinte aux sommets x tels que $f_i(x) = c$. On maintient de plus un pointeur bidirectionnel entre x et son occurrence dans $\ell'_i(f_i(x))$. Puisque l'ensemble des facettes tuées par x forme un sous-graphe connexe de $\mathcal{S}(X_i)$, on



FIGURE 13.2 – Mise à jour de l'enveloppe convexe $S(X_{i-1})$ en dimension d = 3 lors de l'insertion de x_i . Sur la figure $f_{i-1}(x) = f_{i-1}(y) = c$ et $f_i(x) = f_i(y) = c'$ tandis que $f_i(z) = c_j$.

peut identifier, par un simple parcours à partir de la facette $f_i(x)$, toutes les facettes en conflits avec x en un temps proportionnel à leur nombre (le degré des sommets de ce graphe est borné par hypothèse de position générale).

Afin d'appliquer les résultats du lemme 13.8 il faut valider la condition de mise à jour statique. Voyons les opérations à effectuer suite à l'ajout du sommet x_i pour mettre à jour $\mathcal{S}(X_{i-1})$ et les listes de conflits restreintes.

- Il faut supprimer les facettes courantes en conflit avec x_i . Comme décrit ci-dessus ces facettes s'obtiennent en temps proportionnel à leur nombre à partir de $f_{i-1}(x_i)$.
- Il faut ajouter les nouvelles facettes incidentes au sommet x_i . Soit $\{e_j\}_{j\in J}$ l'ensemble des (d-2)-faces bordantes, c.-à-d. incidentes à la fois à une facette supprimée et à une facette conservée (cf. figure 13.2). Notons que les faces bordantes e_j peuvent être déterminées lors du parcours des faces à supprimer. Les facettes activées par x_i sont de la forme $Conv(\{x_i\} \cup e_j)$. Pour d = 2, $\{e_j\}_{j\in J}$ est réduit à deux sommets et il est aisé d'établir les nouvelles relations d'adjacences entre arêtes actives. Pour d = 3, les arêtes dans $\{e_j\}_{j\in J}$ forment un cycle et leurs adjacences induisent celles entre les triangles activés correspondants.
- Pour chaque facette c supprimée, il faut mettre à jour les conflits avec les sommets de $\ell'_i(c)$. Soit $x \in \ell'_i(c)$. On parcourt les facettes actives en conflits avec x à partir de $f_{i-1}(x) = c$ en temps constant par facette. Si l'on rencontre une facette c' qui n'est pas tuée par x_i , alors on pose $f_i(x) = c'$. Sinon, toutes les facettes en conflits avec x sont tuées par x_i , c.-à-d. $\ell_{i-1}(x) \subset \ell_{i-1}(x_i)$, et ou bien x est intérieur à $\mathcal{S}(X_i)$, et on pose $f_i(x) = NULL$, ou bien x est en conflit avec l'une des facettes activées par x_i . Soit $c_j = Conv(\{x_i\} \cup e_j)$ une telle face. La face de $\mathcal{S}(X_{i-1})$ incidente à e_j et tuée par x_i est nécessairement en conflit avec x. Pour identifier c_j , il suffit donc, lors du parcours de $\ell_{i-1}(x)$, de tester si la face parcourue est incidente à une (d-2)-faces bordante e_j et dans ce cas de vérifier si $Conv(\{x_i\} \cup e_j)$ est en conflit avec x. Si aucun test n'aboutie, c'est que x est intérieur à $\mathcal{S}(X_i)$ et $f_i(x) = NULL$. Clairement, la mise à jour de $f_i(x)$ prend un temps proportionnel au nombre de faces tuées par x_i et en conflit avec x.

Comme x n'apparaît que dans une unique liste $\ell'_i(c)$, on en déduit que la mise à jour de toutes les listes (restreintes) de conflits prend un temps proportionnel au nombre de conflits tués par x_i .

Finalement, l'ensemble des opérations à effectuer à l'étape i vérifie bien la condition de mise à jour statique. L'hypothèse de position générale implique que le degré de l'espace de configurations est précisément d. La complexité moyenne de l'algorithme ci-dessus est donc donnée par le lemme 13.8, ou encore par

$$O(d\sum_{i=1}^{n} \frac{e_i}{i} + d^2 \sum_{i=1}^{n} \frac{n-i}{i^2} e_i)$$

Or, en dimension 2 ou 3 la taille de l'enveloppe convexe de X_i est un O(i) (pour la dimension 3, on pourra utiliser la formule d'Euler). On en déduit que la complexité moyenne est un $O(n \log n)$ en dimension 2 ou 3. Notons que cette complexité est optimale en dimension 2 (et donc 3) par réduction du problème du tri au calcul de l'enveloppe convexe : le tri de n nombres t_i s'obtient en temps linéaire à partir de l'enveloppe convexe des n points du plan (t_i, t_i^2) .

Chapitre 14

Transversaux, ϵ -net et dimension de Vapnik-Chervonenkis

14.1 Transversaux et couplages

Soit X un ensemble et $\mathcal{F} \subset \mathcal{P}(X)$ une famille de parties de X. Le couple (X, \mathcal{F}) est appelé selon les auteurs un système de parties ou un hypergraphe ou encore un espace de domaines (range space). Une partie T de X est dite transverse à \mathcal{F} si T intersecte tout élément de \mathcal{F} . Le nombre transversal, $\tau(\mathcal{F})$, de \mathcal{F} est le cardinal minimal de toute partie transverse. Un transversal fractionnaire pour \mathcal{F} est une fonction réelle $\phi : X \to [0, 1]$ telle que

$$\forall S \in \mathcal{F} : \sum_{x \in S} \phi(x) \ge 1$$

On définit la taille du transversal fractionnaire ϕ par $\sum_{x \in X} \phi(x)$. La taille minimale de tout transversal fractionnaire est appelée le nombre transversal fractionnaire. Il est noté $\tau^*(\mathcal{F})$. La fonction caractéristique de toute partie transverse est un transversal fractionnaire à valeurs dans $\{0, 1\}$.

Un sous-ensemble d'éléments de \mathcal{F} deux à deux disjoints est appelé un *couplage (packing* en anglais) pour \mathcal{F} . Le cardinal maximal de tout couplage est appelé le *nombre de couplage* de \mathcal{F} . Il est noté $\nu(\mathcal{F})$. Un *couplage fractionnaire* pour \mathcal{F} est une fonction $\psi : \mathcal{F} \to [0, 1]$ telle que

$$\forall x \in X : \sum_{S \ni x} \psi(S) \leq 1$$

La taille du couplage fractionnaire ψ est par définition $\sum_{S \in \mathcal{F}} \psi(S)$. Le nombre de couplage fractionnaire est la taille maximale de tout couplage fractionnaire. Il est noté $\nu^*(\mathcal{F})$. À tout couplage correspond un couplage fractionnaire à valeurs dans $\{0, 1\}$.

Théorème 14.1 Soit (X, \mathcal{F}) un système de parties. On suppose X fini. le nombre de couplage fractionnaire est rationnel et égal au nombre transversal fractionnaire. Il est de plus atteint par la taille d'un transversal fractionnaire ainsi que par la taille d'un couplage fractionnaire.

Preuve : Soit $A \in \{0,1\}^{X \times \mathcal{F}}$ la matrice définie par $a_{x,s} = 1$ si $x \in S$ et $a_{x,s} = 0$ sinon. On a par définition

$$\nu^*(\mathcal{F}) = \max\{\mathbb{I}\psi \mid \psi \ge \mathbf{0}, A\psi \le \mathbf{1}\} \text{ et } \tau^*(\mathcal{F}) = \min\{\phi\mathbf{1} \mid \phi \ge \mathbf{0}, \phi A \ge \mathbb{I}\}$$

L'égalité $\nu^*(\mathcal{F}) = \tau^*(\mathcal{F})$ apparaît alors comme une application directe du corollaire 7.2 du théorème de dualité de la programmation linéraire. Par ailleurs $\nu^*(\mathcal{F})$ (resp. $\tau^*(\mathcal{F})$) est un extremum d'une forme linéaire à coefficients rationnels (égaux à 1) définie sur le polytope { $\psi \geq 0, A\psi \leq 1$ } (resp. { $\phi \geq 0, \phi A \geq 1$ }). Un tel extremum est atteint en un sommet du polytope. Ce sommet est à coordonnées rationnelles puisque solution d'un système définit par un sous-ensemble des équations du polytope (cf. théorème 7.7). \Box

Dans certaines applications il peut être utile de prouver qu'un système de parties particulier admet un petit transversal. La théorie de la dimension de Vapnik-Chervonenkis permet dans un cadre assez général de majorer le nombre transversal d'un système de parties.

14.2 Dimension de Vapnik-Chervonenkis

Définition 14.2 Soit (X, \mathcal{F}) un système de parties. Pour toute partie $Y \subset X$ on note $\mathcal{F}_Y = \{S \cap Y : S \in \mathcal{F}\}$ la trace de \mathcal{F} sur Y. On dit que Y est éclatée par \mathcal{F} si $\mathcal{F}_Y = \mathcal{P}(Y)$. La dimension de Vapnik-Chervonenkis (VC pour abréger) du système (X, \mathcal{F}) est le cardinal maximal (possiblement infini) de toute partie de X éclatée par \mathcal{F} . On la note $\dim_{VC}(\mathcal{F})$.

- Exemple 1. Le système $(\mathbb{R}^d, \mathcal{H})$ où \mathcal{H} est l'ensemble des demi-espaces fermés de \mathbb{R}^d a une dimension de VC égale à d+1. En effet, les d+1 sommets d'un simplexe non-dégénéré peuvent clairement être éclatés par \mathcal{H} . Par ailleurs, soit un ensemble fini $S \subset \mathbb{R}^d$ de d+2 sommets au moins. Par le lemme 6.6 de Radon il existe deux parties disjointes S_1 et S_2 de S dont les enveloppes convexes s'intersectent. On en déduit que S_1 ne peut être séparé de S_2 par un hyperplan et donc que $S_1 \notin \mathcal{H}_S$.
- Exemple 2. Le système $(\mathbb{R}^d, \mathcal{P})$ où \mathcal{P} est l'ensemble des polytopes de \mathbb{R}^d a une dimension de VC infinie. En effet, pour tout entier n on considère un ensemble S de n points sur la sphère unité de \mathbb{R}^d . L'ensemble S est éclaté par \mathcal{P} : pour toute partie $T \subset S$, on a $T = S \cap Conv(T)$, d'où $T \in \mathcal{P}_S$.
- Exemple 3. Soit X un ensemble de n demi-espaces dans \mathbb{R}^d . Soit H l'arrangement des hyperplans supports de ces demi-espaces. On associe à toute d-cellule de H l'ensemble des demi-espaces de X contenant cette cellule. On considère le système \mathcal{D} constitué des parties de X ainsi obtenues. On montre que $\dim_{VC}(\mathcal{D}) = d$ et que \mathcal{D} est formé d'au plus $\sum_{i=0}^{d} {n \choose i}$ parties de X (cf. lemme 9.13).

Exercice 14.3 Soit un système de parties (X, \mathcal{F}) et $Y \subset X$. Vérifier que $\dim_{VC}(\mathcal{F}_Y) \leq \dim_{VC}(\mathcal{F})$.

On pose $\Phi_d(n) = \sum_{i=0}^d {n \choose i}$.

Théorème 14.4 (Vapnik-Chervonenkis, Sauer, Sheah, 1971) Soient deux entiers *n* et *d*, et un système (X, \mathcal{F}) avec |X| = n et $\dim_{VC}(\mathcal{F}) \leq d$. Alors $|\mathcal{F}| \leq \Phi_d(n)$.

Preuve : Soit $x \in X$. On note \mathcal{F}_{X-x} la trace de \mathcal{F} sur $X - \{x\}$ et on pose

$$\mathcal{F}_1 = \{ S \in \mathcal{F} \mid x \notin S \land S \cup \{x\} \in \mathcal{F} \}$$

$$\mathcal{F}_2 = \{ S \in \mathcal{F} \mid x \in S \land S - \{x\} \in \mathcal{F} \}$$

On a

$$|\mathcal{F}| = |\mathcal{F}_1| + |\mathcal{F}_{X-x}| \tag{14.1}$$

En effet, $S \mapsto S - \{x\}$ établit une bijection entre \mathcal{F}_2 et \mathcal{F}_1 d'une part et entre $\mathcal{F} \setminus \mathcal{F}_2$ et \mathcal{F}_{X-x} d'autre part. Par ailleurs, si une partie $Y \subset X - \{x\}$ est éclatée par \mathcal{F}_1 alors clairement $Y \cup \{x\}$ est éclatée par \mathcal{F} . On en déduit $\dim_{VC}(\mathcal{F}_1) \leq d-1$. On a de plus (cf. exercice 14.3) $\dim_{VC}(\mathcal{F}_{X-x}) \leq d$.

On en déduit par récurrence sur d + n (par convention $\binom{0}{0} = 1$ et $\binom{n}{d} = 0$)

$$|\mathcal{F}| \le \Phi_{d-1}(n-1) + \Phi_d(n-1) = \Phi_d(n)$$

La dernière égalité étant une simple conséquence de la formule de Pascal.

Remarquons en particulier que le théorème fournit la majoration $|\mathcal{F}| \leq (en/d)^d = O(n^d)$ (cf. section 1.6.2). C'est dans la pratique cette majoration qui rend la notion de dimension de VC intéressante et permet de démontrer l'existence de petits transversaux.

Le théorème suivant peut s'avérer utile pour donner une borne sur la dimension de VC d'un système.

Proposition 14.5 Soit \mathcal{F} un système de parties avec $\dim_{VC}(\mathcal{F}) = d$. Soit \mathcal{T} un autre système dont les parties peuvent se déduirent de \mathcal{F} par une formule ensembliste fixée (utilisant les unions, intersections et différences) de k opérations au plus (par exemple de la forme $(S_1 \cap (S_2 \cup \ldots \cup S_{k-1})) \setminus S_k$ avec $S_1, \ldots S_k \in \mathcal{F}$). Alors

$$\dim_{VC}(\mathcal{T}) \le 2dk \log(2dk)$$

La preuve est laissée à titre d'exercice et peut aussi être trouvée dans [PA95, p. 254] ou Matousek, [Mat02, p. 245].

Comme application directe on déduit de l'exemple 1 ci-dessus que la famille des polytopes à au plus k sommets dans \mathbb{R}^{d-1} a une dimension de VC finie et majorée par la formule de la proposition.

En fait l'existence de petits transversaux est difficile à obtenir même avec une dimension de VC finie. Par contre si on relâche la notion de transversal en exigeant uniquement d'intersecter toutes les parties *suffisamment grosses* du système considéré on peut obtenir des quasi-transversaux de tailles indépendantes du système considéré !

Théorème 14.6 Soit (X, μ) un espace de probabilité de mesure μ . Soit \mathcal{F} une famille de parties mesurables de X. On note d sa dimension de VC. Soit un entier positif t. On considère un échantillon aléatoire x de X^t obtenu par t tirages indépendants suivant la répartition μ (i.e. selon la loi produit μ^t). On note $\{x\}$ l'union des composantes de x, c'est une partie de X à t éléments au plus. Alors pour tout entier T > t et tout $\epsilon > 0$ tel que tout $S \in \mathcal{F}$ est de mesure au moins ϵ , $\{x\}$ est transversal pour \mathcal{F} avec une probabilité au moins

$$1 - 2\Phi_d(T)(1 - \frac{t}{T})^{(T-t)\epsilon - 1}$$

Preuve : Soit y un échantillon aléatoire de X^{T-t} obtenu par T-t tirages indépendants suivant la répartition μ . On veut estimer la probabilité

 $P(\{x\} \text{ n'est pas un transversal}) = P(\exists S \in \mathcal{F}; I(S, x) = 0)$

où I(S, x) est le nombre de composantes de x appartenant à S.

Pour tout entier m tel que $\min_{S \in \mathcal{F}} P(I(S, y) \ge m)$ est positif, j'affirme que

$$P(\exists S \in \mathcal{F}; I(S, x) = 0) \le \frac{P(\exists S \in \mathcal{F}; I(S, x) = 0 \land I(S, y) \ge m)}{\min_{S \in \mathcal{F}} P(I(S, y) \ge m)}$$

Preuve de l'affirmation : On considère les probabilités conditionnelles

$$p_0 = P(\exists S \in \mathcal{F}; I(S, x) = 0 | x) \text{ et } p_1 = P(\exists S \in \mathcal{F}; I(S, x) = 0 \land I(S, y) \ge m | x)$$

- Si $\{x\}$ est un transversal alors p_0 est nul.

- Si $\{x\}$ n'est pas un transversal, on considère S' tel que I(S', x) = 0. On a alors

$$p_1 \geq P(I(S', y) \geq m \mid x)$$

= $P(I(S', y) \geq m)$ (car x et y sont indépendants)
 $\geq \min_{S \in \mathcal{F}} P(I(S, y) \geq m)$

D'où

$$p_1 \ge p_0 \min_{S \in \mathcal{F}} P(I(S, y) \ge m)$$

On en conclut que cette inégalité est vraie inconditionnellement.

On choisit maintenant m égal à la médiane d'une variable aléatoire $X_{T-t,\epsilon}$ suivant la loi binomiale de paramètres T et ϵ . En particulier $P(X_{T-t,\epsilon} \ge m) \ge 1/2$. Pour tout $S \in \mathcal{F}$, la fonction I(S, y) suit une loi binomiale de paramètres T et $\mu(S)$. Comme $\mu(S) \ge \epsilon$ et que la médiane de $X_{T-t,\epsilon}$ croît avec ϵ on en déduit $\min_{S \in \mathcal{F}} P(I(S, y) \ge m) \ge 1/2$, d'où

$$P(\exists S \in \mathcal{F}; I(S, x) = 0) \le 2P(\exists S \in \mathcal{F}; I(S, x) = 0 \land I(S, y) \ge m)$$

On pose $z = (x, y) \in X^T$ et on note [z] le multi-ensemble des composantes de z. Soit $S \in \mathcal{F}$. J'affirme que

$$P(I(S,x) = 0 \land I(S,y) \ge m \mid [z]) = \chi(I(S,z) \ge m) \binom{T-t}{I(S,z)} / \binom{T}{I(S,z)}$$

où χ désigne la fonction caractéristique d'une propriété. En effet, si I(S, z) < m alors la probabilité ci-dessus est nulle. Sinon, on remarque que chacune des $\binom{T}{I(S,z)}$ affectations des I(S, z) élément de [z] appartenant à S vers les composantes de z est équiprobable et qu'une telle affectation donne une probabilité égale à 1 ou 0 selon qu'elle affecte uniquement les T - t composantes de y dans z ou non. On en déduit

$$P(I(S,x) = 0 \land I(S,y) \ge m \mid [z]) \le \chi(I(S,z) \ge m)(1-\frac{t}{T})^s \le (1-\frac{t}{T})^m$$

Par le théorème 14.4, [z] est intersecté de au plus $\Phi_d(T)$ manières différentes par les éléments de \mathcal{F} . Or, si S et S' intersectent [z] de la même manière, on a évidemment

$$\{z\,;\, I(S,x)=0\,\wedge\, I(S,y)\geq m\}=\{z\,;\, I(S',x)=0\,\wedge\, I(S',y)\geq m\}$$

D'où finalement

$$P(\exists S \in \mathcal{F}; I(S, x) = 0 \land I(S, y) \ge m \mid [z]) \le \Phi_d(T)(1 - \frac{t}{T})^m$$

On conclut en utilisant le fait que la médiane m de la variable binomiale $X_{T-t,\epsilon}$ s'écarte au plus d'une unité de sa moyenne $(T-t)\epsilon$.

Définition 14.7 Soit (X, \mathcal{F}) un système de parties et soit $\epsilon > 0$. Un ϵ -net (ϵ -transversal?) pour (X, \mathcal{F}) est une partie de X qui est un transversal pour les éléments de \mathcal{F} de taille au moins $\epsilon |X|$. Dit autrement un ϵ -net pour (X, \mathcal{F}) est un transversal pour $(X, \mathcal{F}_{\epsilon})$ où $\mathcal{F}_{\epsilon} = \{S \in \mathcal{F} : |S| > \epsilon |X|\}.$

Corollaire 14.8 Soit (X, \mathcal{F}) un système de parties de dimensions de VC finie d. Alors pour tout ϵ positif, (X, \mathcal{F}) admet un ϵ -net de taille au plus

$$\frac{d}{\epsilon}(\ln\frac{1}{\epsilon} + 2\ln\ln\frac{1}{\epsilon} + 6)$$

Bibliographie

[AE99]	Pankaj K. Agarwal and Jeff Erickson. Geometric range searching and its rel- atives. In B. Chazelle, J.E. Goodman, and R. Pollack, editors, Advances in Discrete and Computational Geometry, volume 23 of Contemporary Math- ematics, pages 1–56. American Mathematical Society, 1999.
[Agg84]	A. Aggarwal. The art gallery problem : Its variations, applications, and algorithmic aspects. PhD thesis, Johns Hopkins University, 1984.
[AGR01]	N. Amato, M. Goodrich, and E. Ramos. A randomized algorithm for tri- angulating a simple polygon in linear time. <i>Discrete & Computational</i> <i>Geometry</i> , 26 :245–265, 2001.
[AK00]	F. Aurenhammer and R. Klein. Voronoi diagrams. In J. Sack and G. Urru- tia, editors, <i>Handbook of Computational Geometry</i> , chapter V, pages 201– 290. Elsevier Science Publishing, 2000.
[BLW98]	Norman L. Biggs, E. Keith Lloyd, and Robin J. Wilson. <i>Graph theory</i> 1736-1936. Oxford University Press, 1998.
[BY95]	J-D. Boissonnat and M. Yvinec. <i>Géométrie algorithmique</i> . Ediscience international, Paris, 1995. Version anglaise : Algorithmic geometry. Cambridge University Press, 1998.
[CdVPV01]	É. Colin de Verdière, M. Pocchiola, and G. Vegter. Tutte's barycenter method applied to isotopies. In <i>Proc. Canadian Conf. on Computational Geometry (CCCG)</i> , 2001.
[CdVPV03]	É. Colin de Verdière, M. Pocchiola, and G. Vegter. Tutte's barycenter method applied to isotopies. <i>Computational Geometry : Theory and Applications</i> , 26(1):81–97, 2003.
[CF90]	B. Chazelle and J. Friedman. A deterministic view of random sampling and its use in computational geometry. <i>Combinatorica</i> , 10:229–249, 1990.
[Cha82]	B. Chazelle. A theorem on polygon cutting with applications. 339–349. In 23rd Annu. IEEE Sympos. Found. Comput. Sci., pages 339–349. IEEE, 1982.
[Cha90]	Bernard Chazelle. Lower bounds for orthogonal range searching : I. the reporting case. Journal of the ACM, $37(2)$:200–212, 1990.
[Cha91]	B. Chazelle. Triangulating a simple polygon in linear time. Discrete & Computational Geometry, 6 :485–524, 1991.

[Che52]	Herman Chernoff. A measure of asymptotic efficiency for tests of a hypoth- esis based on the sum of observations. <i>Annals of Mathematical Statistics</i> , 23 :493–507, 1952.
[CLP11]	Timothy M Chan, Kasper Green Larsen, and Mihai Pătrașcu. Orthogonal range searching on the ram, revisited. In <i>Proceedings of the twenty-seventh annual symposium on Computational geometry</i> , pages 1–10. ACM, 2011. J'ai une version papier.
[CLRS02]	T. Cormen, C. Leiserson, R. Rivest, and C. Stein. Introduction à l'algorithmique 2nd edition. Dunod, 2002.
[CMS93]	K. L. Clarkson, K. Mehlhorn, and R. Seidel. Four results on randomized incremental constructions. <i>Comp. Geom. : Theory and Applications</i> , 3:185–212, 1993. J ai une version papier.
[CS89]	K. Clarkson and P. Shor. Applications of random sampling in computa- tional geometry, ii. <i>Discrete & Computational Geometry</i> , 4:387–421, 1989.
[Dan51]	George B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. In <i>Activity and Analysis of Production and Allocation</i> , number 13 in Cowles commission Monograph, pages 339–347. Chapman and Hall, 1951.
[dBCvKO08]	Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. <i>Computational Geometry : Algorithms and Applications</i> . Springer-Verlag, third edition (march 2008) edition, 2008.
[DL76]	D. P. Dobkin and R.J. Lipton. The complexity of searching lines in the plane. Technical report, Dept. Comput. Sci. Yale Univ., 1976.
[EG89]	H. Edelsbrunner and L. J. Guibas. Topologically sweeping an arrangement. J. Comput. Syst. Sci., 38 :165–194, 1989. Corrigendum in 42 (1991) 249–251.
[Fár48]	I. Fáry. On straight line representation of planar graphs. Acta sci. math. (Szeged), 1 :229–233, 1948.
[GH89]	L. J. Guibas and J. Hershberger. Optimal shortest path queries in a simple polygon. <i>Journal of Computer and System Sciences</i> , 39(2):126–152, 1989.
[GHL ⁺ 87]	Leonidas J. Guibas, John Hershberger, Daniel Leven, Micha Sharir, and Robert E. Tarjan. Linear-Time Algorithms for Visibility and Shortest Path Problems Inside Triangulated Simple Polygons. <i>Algorithmica</i> , 2:209–233, 1987.
[GJPT78]	M. R. Garey, D. S. Johnson, F. P. Preparata, and R. E. Tarjan. Triangulating a simple polygon. <i>Information Processing Letter</i> , 7 :175–179, 1978.
[GO04]	J. E. Goodman and J. O'Rourke, editors. <i>Handbook of Discrete and Computational Geometry</i> . CRC Press, second edition edition, 2004.
[GPP08]	J. E. Goodman, J. Pach, and R. Pollack, editors. <i>Surveys on Discrete and Computational Geometry. Twenty Years Later</i> , volume 453 of <i>Contemporary mathematics</i> . AMS, 2008.
[Her91]	J. Hershberger. A new data structure for shortest path queries in a simple polygon. <i>Information Processing Letters</i> , 38(5):231–235, 1991.

[HP00]	Sariel Har-Peled. Constructing planar cuttings in theory and practice. <i>SIAM J. Comput.</i> , 29(6) :2016–2039, 2000.
[Kir83]	David Kirkpartrick. Optimal search in planar subdivisions. SIAM J. Comput., $12(1)$:28–35, 1983.
[KS06]	Jonathan A. Kelner and Daniel A. Spielman. A randomized polynomial- time simplex algorithm for linear programming. In <i>Symposium on Theory</i> of Computing (STOC'06), pages $51 - 60$, 2006.
[Len11]	N. Lennes. Theorems on the simple finite polygon and polyhedron. Amer. J. Math., 33:36–62, 1911.
[LMS94]	Chi-Yuan Lo, Jiří Matoušek, and W. Steiger. Algorithms for ham-sandwich cuts. Discrete & Computational Geometry, 11:433–452, 1994.
[LP84]	D. T. Lee and F. P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. <i>Networks</i> , 14 :393–410, 1984.
[Mat91]	Jiří Matoušek. Cutting hyperplane arrangements. Discrete & Computa- tional Geometry, 6:385–406, 1991.
[Mat92]	Jiří Matoušek. Efficient partition trees. Discrete & Computational Geom- etry, 8:315–334, 1992.
[Mat94]	Jiří Matoušek. Geometric range searching. <i>ACM Computing Surveys</i> , 26(4) :421–461, 1994.
[Mat02]	Jiří Matoušek. <i>Lectures on Discrete Geometry</i> . Number 212 in GTM. Springer Verlag, 2002.
[Meg84]	N. Megiddo. Linear programming in linear time when the dimension is fixed. <i>JACM</i> , 31 :114–127, 1984.
[MG07]	Jiří Matoušek and Bernd Gärtner. Understanding and Using Linear Pro- gramming. Springer Verlag, 2007.
[MT01]	Bojan Mohar and Carsten Thomassen. <i>Graphs on Surfaces</i> . John Hopkins University Press, 2001.
[Mul94]	Ketan Mulmuley. Computational Geometry. An Introduction Through Ran- domized Algorithms. Prentice Hall, 1994.
[Mul00]	Ketan Mulmuley. Randomized Algorithms in Computational Geometry. In JR. Sack and J. Urrutia, editors, <i>Handbook of Computational Geometry</i> , chapter Chapter 16, pages 703–724. North-Holland, 2000.
[PA95]	János Pach and Pankaj Agarwal. <i>Combinatorial Geometry</i> . John Wiley, 1995.
[San12]	Francisco Santos. A counterexample to the hirsch conjecture. Annals of Mathematics, to appear, 2012.
[Sei91]	R. Seidel. Small-dimensional linear programming and convex hulls made easy. Discrete & Computational Geometry, 6 :423–434, 1991.
[Sei95]	R. Seidel. The upper bound theorem for polytopes : an easy proof of its asymptotic version. <i>Computational Geometry : Theory and Applications</i> , 5(2) :115–116, 1995.

[SU00]	JR. Sack and J. Urrutia, editors. <i>Handbook of Computational Geometry</i> . North-Holland, 2000.
[Tar72]	R. E. Tarjan. Depth-first search and linear graph algorithms. <i>SIAM Journal</i> on Computing, 1(2) :146–160, 1972.
[Tut63]	William T. Tutte. How to Draw a Graph. Proc. London Mathematical Society, 13:743–768, 1963.
[vEB90]	Peter van Emde Boas. <i>Handbook of theoretical computer science</i> , volume A, chapter Machine Models and Simulations, pages 3–66. Elsevier, 1990.
[Wil96]	R. J. Wilson. <i>Introduction to Graph Theory</i> . Prentice Hall, fourth edition, 1996.
[Zie95]	Günter M. Ziegler. <i>Lectures on polytopes</i> . Number 152 in Graduate texts in mathematics. Springer, rev. first ed edition, 1995.