

Chapitre 6

Algorithmes randomisés incrémentaux

6.1 Introduction

Étant donné un ensemble X d'objets (ex : des points du plan, des droites du plan, ...), 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, \dots, x_n de X , on construit récursivement $\mathcal{S}_i = \mathcal{S}(\{x_1, \dots, x_i\})$ 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 Vegas* 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. En particulier, la suite des i premiers objets constitue un i -échantillon aléatoire de X , au sens où sa probabilité de réalisation est $1/\binom{n}{i}$. Quicksort est l'archétype des algorithmes randomisés incrémentaux.

6.2 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 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 parlé 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 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 $\mathcal{C}^0(X)$, sont dites *actives* sur X .

Voici d'autres exemples importants où le formalisme précédent s'applique.

Exemple 6.1 (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 en faisant 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*

lexicographique. 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 6.2 (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 S un 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 6.3 (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 6.4 (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 6.1) 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 $S \cup \{p\}$, i.e. tel qu'il existe un point de t qui soit plus proche de p que des points de S .

Exemple 6.5 (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 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 6.6 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 6.7 Un espace de configurations est un quadruplet $(\mathcal{O}, \mathcal{C}, \delta, \kappa)$ où

1. \mathcal{O} est un ensemble (on dit aussi univers) dont les éléments sont appelés objets,
2. \mathcal{C} est un ensemble, dont les éléments sont appelés configurations,
3. $\delta : \mathcal{C} \rightarrow \mathcal{P}(\mathcal{O})$ associe à chaque configuration un sous-ensemble d'objets. Une configuration c est dite déterminée par les objets de $\delta(c)$.
4. $\kappa : \mathcal{C} \rightarrow \mathcal{P}(\mathcal{O})$ 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

- $\mathcal{C}(X)$ est l'ensemble des configurations de \mathcal{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 $\kappa_X : \mathcal{C}(X) \rightarrow \mathcal{P}(X)$ est définie par $\kappa_X(c) = \kappa(c) \cap X$.

Une configuration $c \in \mathcal{C}$ est dite active sur X si $c \in \mathcal{C}(X)$ et $\kappa(c) \cap X$ est vide. On note $\mathcal{C}^0(X)$ l'ensemble des configurations actives sur X .

6.3 Algorithmes statiques

6.3.1 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, \dots, x_n) des éléments de X . On pose $X_i = \{x_1, x_2, \dots, x_i\}$.
- (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

$$\mathcal{G}_i = \{(c, x) : c \in \mathcal{C}^0(X_i) \wedge x \in \kappa(c)\}$$

En particulier, $\mathcal{G}_i \subset \mathcal{C}^0(X_i) \times (X \setminus X_i)$ est un graphe bipartite. 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 $\mathcal{S}(X_i)$, 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 $\mathcal{S}(X_i)$ 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 $t_0 = t'_0 = t_n = t'_n = 0$:

$$\forall j \in [1, n] : \quad \sum_{i=1}^j t_i \leq \sum_{i=1}^j a_i \text{ et } \sum_{i=1}^j t'_i \leq \sum_{i=1}^j a'_i. \quad (6.1)$$

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 6.8 *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 actives 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 $A_i = \mathcal{C}^0(X_i) \setminus \mathcal{C}^0(X_{i-1})$ l'ensemble des configurations activées à l'étape i et $a_i = |A_i|$. Soit c une configuration active à l'étape i . Alors c n'était pas active à l'étape $i - 1$ si et seulement si $x_i \in \delta(c)$. On en déduit

$$\begin{aligned} E(a_i) &= \sum_{c \in \mathcal{C}(X)} P(c \in A_i) = \sum_{c \in \mathcal{C}(X)} P(c \in \mathcal{C}^0(X_i) \wedge x_i \in \delta(c)) \\ &= \sum_{c \in \mathcal{C}(X)} P(x_i \in \delta(c) \mid c \in \mathcal{C}^0(X_i)) P(c \in \mathcal{C}^0(X_i)) \end{aligned}$$

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 . Pour majorer $P(x_i \in \delta(c) \mid c \in \mathcal{C}^0(X_i))$ on note que

$$P(x_i \in \delta(c) \mid c \in \mathcal{C}^0(X_i) \wedge X_i \text{ fixé}) \leq \frac{d}{i}$$

En effet, 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 $\delta(c)$. Cette majoration subsiste donc inconditionnellement (cf. exercice 1.20), i.e. $P(x_i \in \delta(c) \mid c \in \mathcal{C}^0(X_i)) \leq d/i$. On en déduit

$$E(a_i) \leq \frac{d}{i} \sum_{c \in \mathcal{C}(X)} P(c \in \mathcal{C}^0(X_i))$$

Cette dernière somme n'est autre que e_i . □

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

$$a'_i = \sum_{c \in A_i} |\kappa(c)| = \sum_{j > i} a'_i(x_j)$$

où $a'_i(x_j) = |A_i \cap \{c \in \mathcal{C}(X) : x_j \in \kappa(c)\}|$ 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 $E(a'_i(x_j))$ ne dépend pas de j (cf. exercice 6.9 ci-dessous), i.e. $E(a'_i(x_j)) = E(a'_i(x_{i+1}))$ pour $j > i$. D'où

$$E(a'_i) = (n - i)E(a'_i(x_{i+1}))$$

Comme

$$a'_i(x_{i+1}) = |\{c \in \mathcal{C}(X) : x_i \in \delta(c) \wedge c \in \mathcal{C}^0(X_i) \setminus \mathcal{C}^0(X_{i+1})\}|$$

on a encore, en notant $T_i = \mathcal{C}^0(X_i) \setminus \mathcal{C}^0(X_{i+1})$ l'ensemble des configurations actives tuées à l'étape $i + 1$:

$$\begin{aligned} E(a'_i(x_{i+1})) &= \sum_{c \in \mathcal{C}(X)} P(x_i \in \delta(c) \wedge c \in T_i) \\ &= \sum_{c \in \mathcal{C}(X)} P(x_i \in \delta(c) \mid c \in T_i) P(c \in T_i) \end{aligned}$$

Par un raisonnement analogue à la preuve précédente, on majore $P(x_i \in \delta(c) \mid c \in T_i)$ par d/i . On en déduit

$$E(a'_i) \leq d \frac{n-i}{i} E(t_i)$$

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 (6.1) impliquent (cf. exercice 6.10 ci-dessous)

$$E(\sum_i a'_i) \leq dE(\sum_i \frac{n-i}{i} a_i)$$

La première majoration du lemme sur l'espérance du nombre a_i de configurations activées à l'étape i permet de conclure. \square

Exercice 6.9 Soient X un ensemble de n objets, $k \in \mathbb{N}$ et $f : X^{k+1} \rightarrow \mathbb{R}$. On fixe un k -uplet d'indices distincts $I = (i_1, \dots, i_k) \in [1, n]^k$ et on définit $n - k$ variables aléatoires $\{f_j\}_{j \notin I}$ sur l'ensemble \mathcal{S}_X des permutations de X (muni de la probabilité uniforme) par

$$\forall j \in [1, n] \setminus I, \forall \sigma = (x_1, \dots, x_n) \in \mathcal{S}_X : \quad f_j(\sigma) = f(x_{i_1}, \dots, x_{i_k}, x_j)$$

Montrer que $E(f_j)$ ne dépend pas de j .

Exercice 6.10 Soient deux familles de n nombres $(a_i)_{1 \leq i \leq n}$ et $(t_i)_{1 \leq i \leq n}$ telles que

$$\forall j \in [1, n], \quad \sum_{i=1}^j t_i \leq \sum_{i=1}^j a_i$$

et soit $(\alpha_i)_{1 \leq i \leq n}$ une suite positive décroissante. Montrer que

$$\sum_{i=1}^n \alpha_i t_i \leq \sum_{i=1}^n \alpha_i a_i$$

6.3.2 Applications

On montre ici comment appliquer le formalisme précédent à des exemples de la section 6.2.

Décomposition trapézoïdale

On reprend le cadre de l'exemple 6.2. Soit alors 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 6.8 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 $\ell(\tau)$ pointe vers l'occurrence de τ dans $\ell(x)$ 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 6.3.1, 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 donne lieu à deux parois dont une seule doit être conservée). On modifie $\mathcal{S}(X_{i-1})$ 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 découpent 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 découpent 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 obtenus 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 certains des trapèzes introduits à la première étape. Si l'on doit par exemple fusionner les trapèzes $\tau_1, \tau_2, \dots, \tau_k$ en un trapèze τ' il faut fusionner les listes de conflits $\ell(\tau_1), \ell(\tau_2), \dots, \ell(\tau_k)$ 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'ordonnement 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 $\tau_1, \tau_2, \dots, \tau_k$ (contenant τ_j) par τ' en temps proportionnel à $|\ell'|$. On supprime de plus x des listes de conflits des trapèzes de la sous-liste 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 $\partial h = \text{conv}(S)$. 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 sous-famille 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 de \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 , de sorte que $f_i(x) \in \ell_i(x)$. On pose $f_i(x) = \text{NULL}$ si x est intérieur à $\mathcal{S}(X_i)$. 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 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).

Afin d'appliquer les résultats du lemme 6.8 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 $\{e_j\}$ les $(d-2)$ -faces bordantes, c.-à-d. incidentes à la fois à une facette supprimée et à une facette conservée. 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 $\text{Conv}(\{x_i\} \cup e_j)$. Pour $d=2$, $\{e_j\}$ 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 $\{e_j\}$ 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 proportionnel à leur nombre. Si l'une c' de ces facettes n'est pas tuée par x_i , alors on pose $f_i(x) = c'$. Sinon, on a $\ell_{i-1}(x) \subset \ell_{i-1}(x_i)$ et ou bien x est intérieur à $\mathcal{S}(X_i)$, et alors $f_i(x) = \text{NULL}$, ou bien x est en conflit avec l'une des faces activées par x_i . Soit $c_j = \text{Conv}(\{x_i\} \cup e_j)$ 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 $\text{Conv}(\{x_i\} \cup e_j)$ est en conflit avec x . Si aucun test n'aboutie, c'est que x est intérieur à $\mathcal{S}(X_i)$ et $f_i(x) = \text{NULL}$. 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 6.8, 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 à 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) .