Algorithme LLL et applications à la recherche de racines et à la factorisation de polynômes à coefficients entiers

Matthieu Agogué et Martial Hue

Sujet proposé par Phong Nguyen

Table des matières

1 Rappels et définitions

On rappelle ici des résultats élémentaires sur les réseaux ainsi que le procédé d'orthogonalisation de Schmidt.

1.1 Réseaux

Définition 1 *Un réseau de* $\mathbf{R}^n$ *est un sous-groupe discret de* $\mathbf{R}^n$.

Proposition 1 *Un sous-groupe additif* H *de* $\mathbf{R}^n$ *est discret si et seulement si pour tout compact* K *de* $\mathbf{R}^n$, *l'intersection* $H \cap K$ *est finie.*

Un exemple typique de sous-groupe discret de $\mathbf{R}^n$ est $\mathbf{Z}^n$. On va voir que c'est à peu de choses près le seul.

Théorème 1 *Soit* H *un sous-groupe discret de* $\mathbf{R}^n$. *Alors* H *est engendré en tant que* $\mathbf{Z}$*-module par* r *vecteurs linéairement indépendants sur* $\mathbf{R}$ *(d'où* $r \leqslant n$*).*

$\diamondsuit$ On choisit une famille $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r)$ d'éléments de H linéairement indépendants tel que r soit maximal. On note :

$$P = \{\sum_{i=1}^{r} \alpha_i \boldsymbol{b}_i \mid 0 \leqslant \alpha_i \leqslant 1\}$$

Comme P est compact, $H \cap P$ est fini. Soit $\boldsymbol{x} \in H$. Vu le caractère maximal de $(\boldsymbol{b}_i)_{i=1,\ldots,r}$, on a :

$$\boldsymbol{x} = \sum_{i=1}^{r} \lambda_i \boldsymbol{b}_i$$

avec $\lambda_i \in \mathbf{R}$. On considère pour $j \in \mathbf{Z}$, l'élément de $H \cap P$:

$$x_j = jx - \sum_{i=1}^{r} [j\lambda_i]\boldsymbol{b}_i = \sum_{i=1}^{r} (j\lambda_i - [j\lambda_i])\boldsymbol{b}_i$$

On a donc que H est engendré comme $\mathbf{Z}$-module par $H \cap P$, et est donc de type fini. D'autre part, comme $H \cap P$ est fini, les λ_i sont rationnels. Ainsi H est engendré par un nombre fini d'éléments qui sont combinaisons linéaires à coefficients rationnels des $(\boldsymbol{b}_i)_{i=1,\ldots,r}$. On a donc $d \in \mathbf{Z}$ tel que $dH \subset \sum_{i=1}^{r} \mathbf{Z}\boldsymbol{b}_i$. Ainsi il existe une base $(\boldsymbol{c}_i)_{i=1,\ldots,r}$ du $\mathbf{Z}$-module $\sum_{i=1}^{r} \mathbf{Z}\boldsymbol{b}_i$ et des $\alpha_i \in \mathbf{Z}$ tels que $(\alpha_1\boldsymbol{c}_1, \ldots, \alpha_r\boldsymbol{c}_r)$ engendre dH. De plus, on a $rg(dH) = rg(H)$, et $rg(dH) \geqslant r$, donc $rg(dH) = r$ et les α_i sont tous non-nuls. Donc dH, et par conséquent H, est engendré par r vecteurs linéairement indépendants sur $\mathbf{R}$. $\diamondsuit$

Définition 2 *On appelle pour* $L = \sum_{j=1}^{n} \mathbf{R}\boldsymbol{b}_j$ *un réseau de* $\mathbf{R}^n$ *le déterminant ou volume* $d(L)$ *de* L, *la racine du déterminant de la matrice de Gram de* $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_j)$.

Proposition 2 $d(L)$ *ne dépend pas de la base* $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_j)$ *choisie.*

Théorème 2 *Soit* L *un réseau de* $\mathbf{R}^n$ *et* S *un sous-ensemble intégrable de* $\mathbf{R}^n$ *tels que* $\mu(S) > d(L)$. *Il existe deux éléments* x, y *de* S *tels que* $x - y \in L$.

$\diamondsuit$ Soit $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ une base de L et P le parallèlotope semi-ouvert construit sur cette base. Alors S est la réunion disjointe des $S \cap (\boldsymbol{h} + P)$, pour $h \in L$, d'où :

$$\mu(S) = \sum_{h \in L} \mu(S \cup (h + P))$$

Or par l'invariance par translation de μ, on a $\mu(S \cap (\boldsymbol{h} + P)) = \mu((S - \boldsymbol{h}) \cap P)$, et par suite ces ensembles ne peuvent être deux à deux disjoints sinon, on aurait $\mu(S) \leqslant d(L)$. Par suite, il existe $\boldsymbol{h}, \boldsymbol{h}' \in L$ tels que $((S - \boldsymbol{h}) \cap P) \cap ((S - \boldsymbol{h}') \cap P) \not\varnothing$, d'où $\boldsymbol{x}, \boldsymbol{y} \in S$ tels que $\boldsymbol{x} - \boldsymbol{h} = \boldsymbol{y} - \boldsymbol{h}'$. $\diamondsuit$

1.2 Orthogonalisation de Gram-Schmidt

Soit $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ une famille libre de $\mathbf{R}^n$. On a la proposition suivante :

Proposition 3 *Si l'on définit par récurrence :*

$$\boldsymbol{b}_i^\star = \boldsymbol{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \boldsymbol{b}_j^\star \quad avec\ \mu_{i,j} = \frac{\langle \boldsymbol{b}_i, \boldsymbol{b}_j^\star \rangle}{\|\boldsymbol{b}_j^\star\|^2}$$

on obtient une famille orthogonale de $\mathbf{R}^n$, qui est une base de $\sum_{j=1}^{d} \mathbf{R}\boldsymbol{b}_j$. De plus la matrice donnant les $\boldsymbol{b}_i^\star$ en fonction des $\mathbf{b}_i$ est unipotente. En particulier, si $d(L)$ désigne le déterminant du réseau L, on a :

$$d(L)^2 = \prod_{i=1}^{n} \|\boldsymbol{b}_i^\star\|^2$$

Corollaire 1 *(inégalité d'Hadamard)*
Soit L un réseau de déterminant $d(L)$, $(\boldsymbol{b}_i)_{1 \leqslant i \leqslant n}$ une $\mathbf{Z}$-base. Alors :

$$d(L) \leqslant \prod_{i=1}^{n} \|\boldsymbol{b}_i\|$$

$\diamondsuit$ L'orthogonalité des $\boldsymbol{b}_i^\star$ donne :

$$\|\boldsymbol{b}_i\|^2 = \|\boldsymbol{b}_i^\star\|^2 + \sum_{j=1}^{i-1} \mu_{i,j}^2 \|\boldsymbol{b}_j^\star\|$$

d'où :

$$d(L)^2 = \prod_{i=1}^{n} \|\boldsymbol{b}_i^\star\|^2 \leqslant \prod_{i=1}^{n} \|\boldsymbol{b}_i\|^2 \diamondsuit$$

2 Algorithme LLL

2.1 Bases réduites et propriétés

Parmi les $\mathbf{Z}$-bases du réseau L, il en existe certaines qui sont plus intéressantes que les autres. Ce sont celles dont les éléments sont les plus courts, et elles

sont appelées réduites. Comme les bases ont toutes le même déterminant, le fait d'être réduite implique de ne pas être très loin d'être orthogonale.
On introduit ici la notion donnée par A.K.Lenstra, H.W.Lenstra et L.Lovász dans [1] (appelée maintenant réduction LLL).

Définition 3 *Avec les notations précédentes, la base* $(\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n)$ *est LLL-réduite si :*

$$|\mu_{i,j}| \leqslant \frac{1}{2} \quad \textit{pour } 1 \leqslant j < i \leqslant n$$

et

$$\|\boldsymbol{b}_i^\star + \mu_{i,i-1}\boldsymbol{b}_{i-1}^\star\|^2 \geqslant \frac{3}{4}\|\boldsymbol{b}_{i-1}^\star\|^2 \quad \textit{pour } 1 < i \leqslant n$$

ou de manière équivalente

$$\|\boldsymbol{b}_i^\star\| \geqslant (\frac{3}{4} - \mu_{i,i-1})\|\boldsymbol{b}_{i-1}^\star\|^2$$

On peut remarquer que les vecteurs $\boldsymbol{b}_i^\star + \mu_{i,i-1}\boldsymbol{b}_{i-1}^\star$ et $\boldsymbol{b}_{i-1}^\star$ sont les projections de $\boldsymbol{b}_i$ et $\boldsymbol{b}_{i-1}$ sur le supplémentaire orthogonal de $\sum_{j=1}^{i-2} \mathbb{R}\boldsymbol{b}_j$.

Théorème 3 *Soit* $(\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n)$ *une base LLL-réduite du réseau L. Alors on a les inégalités suivantes :*

1. $$d(L) \leqslant \prod_{i=1}^{n} \|\boldsymbol{b}_i\| \leqslant 2^{n(n-1)/4} d(L)$$

2. *Pour* $1 \leqslant j \leqslant i \leqslant n$, *on a :*
$$\|\boldsymbol{b}_j\| \leqslant 2^{(i-1)/2}\|\boldsymbol{b}_i^\star\|$$

3. $$\|\boldsymbol{b}_1\| \leqslant 2^{(n-1)/4} d(L)^{1/n}$$

4. *Pour tout* $\boldsymbol{x} \in L$, $\boldsymbol{x} \neq \boldsymbol{0}$, *on a :*
$$\|\boldsymbol{b}_1\| \leqslant 2^{(n-1)/2}\|\boldsymbol{x}\|$$

5. *Pour tout famille* $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d)$ *de L de vecteurs linéairement indépendants, on a pour* $1 \leqslant j \leqslant d$
$$\|\boldsymbol{b}_j\| \leqslant 2^{(n-1)/2} max(\|\boldsymbol{x}_1\|, \ldots, \|\boldsymbol{x}_d\|)$$

   ♢ La première inégalité de 1. est l'inégalité d'Hadamard. Pour la seconde, le fait que $(\boldsymbol{b}_i)_{1 \leqslant i \leqslant n}$ soit LLL-réduite implique que :

$$\|\boldsymbol{b}_i^\star\|^2 \geqslant \frac{1}{2}\|\boldsymbol{b}_{i-1}^\star\|^2$$

   c'est-à-dire pour $1 \leqslant j \leqslant i \leqslant n$, on a :

$$(*) \quad \|\boldsymbol{b}_j^\star\|^2 \leqslant 2^{i-j}\|\boldsymbol{b}_i^\star\|^2$$

   et par suite, il vient :

$$\|\boldsymbol{b}_i\|^2 \leqslant \frac{2^{i-1}+1}{2}\|\boldsymbol{b}_i^\star\|^2 \leqslant 2^{i-1}\|\boldsymbol{b}_i^\star\|^2$$

L'inégalité voulue s'obtient alors en multipliant ces inégalités.
Par $(*)$, on a aussi pour $1 \leqslant j \leqslant i \leqslant n$:

$$\|\boldsymbol{b}_j\|^2 \leqslant (2^{i-2} + 2^{i-j-1})\|\boldsymbol{b}_i^\star\|^2$$

et $(2^{i-2} + 2^{i-j-1}) \leqslant 2^{i-1}$, d'où 2.
On prend dans 2., $j = 1$ et on multiplie alors les inégalités obtenues pour $i = 1, \ldots, n$ pour obtenir 3.
Soit $\boldsymbol{x} \neq \boldsymbol{0}$ dans L. On écrit :

$$\boldsymbol{x} = \sum_{j=1}^{i} r_j \boldsymbol{b}_j \quad (r_i \neq 0)$$

Alors $\boldsymbol{x}$ s'écrit aussi :

$$\boldsymbol{x} = \sum_{j=1}^{i} s_j \boldsymbol{b}_j^\star \quad (r_i = s_i)$$

Par suite, comme $r_i \in \mathbb{Z}$, il vient :

$$\|\boldsymbol{x}\|^2 \geqslant |s_i|^2 \|\boldsymbol{b}_i^\star\|^2 \geqslant \|\boldsymbol{b}_i^\star\|^2 \geqslant 2^{1-i}\|\boldsymbol{b}_1\|^2 \geqslant 2^{1-n}\|\boldsymbol{b}_1\|^2$$

ce qui montre 4.
Soit $j \in \{1, \ldots, d\}$. Il existe $k \in \{1, \ldots, d\}$ tel que si

$$\boldsymbol{x}_k = \sum_{i=1}^{n} r_{k,i} \boldsymbol{b}_i$$

il existe $l \geqslant j$ tel que $r_{k,l} \neq 0$. On a alors :

$$max(\|\boldsymbol{x}_1\|, \ldots, \|\boldsymbol{x}_d\|)^2 \geqslant \|\boldsymbol{x}_k\|^2 \geqslant \|\boldsymbol{b}_j^\star\|^2 \geqslant 2^{j-l}\|\boldsymbol{b}_j\|^2 \geqslant 2^{1-n}\|\boldsymbol{b}_j\|^2$$

ce qui est bien 5. $\diamondsuit$

Remarque 1 *Ici on n'a pas utilisé à fond les conditions de réduction LLL. On peut affaiblir les inégalités vérifiées par les $\mu_{i,j}$ et prendre comme conditions :*

$$\mu_{i,j}^2 \leqslant \frac{1}{2} \textit{ pour } j < i - 1$$

$$|\mu_{i,i}| \leqslant \frac{1}{2}$$

En fait le facteur $\frac{3}{4}$ introduit ici peut être remplacé par $\delta \in]\frac{1}{4}, 1[$, mais c'est le facteur de réduction $\frac{3}{4}$ qui fut initialement utilisé dans l'algorithme LLL.

2.2 Description de l'algorithme LLL

On va maintenant pouvoir décrire l'algorithme LLL.
Supposons que les vecteurs $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{k-1}$ soient initialement LLL-réduits (i.e. ils forment une base LLL-réduite du réseau qu'ils engendrent). Le premier vecteur devant être réduit est $\boldsymbol{b}_k$, cela afin que l'on ait :

$$|\mu_{k,j}| \leqslant \frac{1}{2} \textit{ pour } j < k$$

Ceci se fait en remplaçant $\boldsymbol{b}_k$ par $\boldsymbol{b}_k - \sum_{j<k} a_j \boldsymbol{b}_j$ pour des $a_j \in \mathbf{Z}$ de la manière suivante : supposons que

$$|\mu_{k,j}| \leqslant \frac{1}{2} \text{ pour } l < j < k$$

(initialement avec l=k-1). On prend $q = \lfloor \mu_{k,l} \rceil$ l'entier le plus proche de $\mu_{k,l}$ et on remplace $\boldsymbol{b}_k$ par $\boldsymbol{b}_k - q\boldsymbol{b}_l$. Alors les $\mu_{k,j}$ ne sont pas modifiés pour $j > l$ (car $\boldsymbol{b}_j^\star$ est orthogonal aux $\boldsymbol{b}_l$ pour $l < j$). Le nouveau $\mu_{k,j}$ vérifie alors :

$$|\mu_{k,j}| \leqslant \frac{1}{2} \text{ pour } l - 1 < j < k$$

Il faut aussi satisfaire la condition de Lovász :

$$\|\boldsymbol{b}_i^\star\| \geqslant (\frac{3}{4} - \mu_{i,i-1})\|\boldsymbol{b}_{i-1}^\star\|^2$$

Si cette condition est satisfaite, on incrémente k de 1 et on s'occupe du vecteur suivant $\boldsymbol{b}_k$ (si il existe). Si cette condition n'est pas satisfaite, on echange les vecteurs $\boldsymbol{b}_k$ et $\boldsymbol{b}_{k-1}$, et on décrémente k de 1, car on sait seulement que $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{k-2}$ est LLL-réduite.
D'autre part, si on pose $B_k = \|\boldsymbol{b}_k^\star\|^2$, il est efficace de calculer les B_k et les $\mu_{k,j}$ une fois pour toute au début de l'algorithme et de les modifier au fur et à mesure. Cependant, il est plus intéressant de calculer les coefficients de Gram-Schmidt quand on en a besoin, en conservant dans une variable k_{max} la valeur maximale de k qui a été atteinte.
Une autre amélioration est de calculer uniquement $\mu_{k,k-1}$ puisque c'est le seul coefficient qui intervient pour tester la condition de Lovász.
On en déduit l'algorithme suivant :

Algorithme 1 *(Algorithme LLL)*
Étant donné une base $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ du réseau L, cet algorithme transforme les $\boldsymbol{b}_i$ afin que lorsque l'algorithme termine, les $\boldsymbol{b}_i$ forment une base LLL-réduite du réseau. En sortie, l'algorithme fournit la matrice H des coordonnées de la base finale LLL-réduite dans la base initiale, dont on notera H_i les colonnes :

1 *Initialisation*
Poser $k \leftarrow 2$, $k_{max} \leftarrow 1$, $\boldsymbol{b}_1^\star \leftarrow \boldsymbol{b}_1$, $B_1 \leftarrow \|\boldsymbol{b}_1^\star\|^2$ et $H \leftarrow I_n$.

2 *Gram-Schmidt*
Si $k \leqslant k_max$, aller à l'étape **3**.
Sinon, poser $k_{max} \leftarrow k$, puis pour $j = 1, \ldots, k-1$, poser :

$$\mu_{k,j} \leftarrow \frac{\langle \boldsymbol{b}_i, \boldsymbol{b}_j^\star \rangle}{B_j} \text{ et } \boldsymbol{b}_k^\star \leftarrow \boldsymbol{b}_k^\star - \mu_{k,j}\boldsymbol{b}_j^\star$$

Finalement, poser $B_k \leftarrow \|\boldsymbol{b}_k^\star\|^2$ et si $B_k = 0$ sortir un message d'erreur disant que les $\boldsymbol{b}_i$ ne forment pas une famille libre, et terminer l'algorithme.

3 *Condition LLL*
Executer le sous-algorithme RED(k,k-1).
Si $B_k < (0,75 - \mu_{k,k-1}^2)B_{k-1}$, executer le sous-algorithme ECH(k), poser $k \leftarrow max(2, k-1)$ et aller à l'étape **3**.
Sinon, pour $l = k-2, k-3, \ldots, 1$, executer le sous-algorithme RED(k,l), puis poser $k \leftarrow k+1$.

4 *Fin*

Si $k \leqslant n$ aller à l'étape **2**.
Sinon, sortir la base LLL-réduite $\boldsymbol{b}_i$, la matrice $H \in GL_n(\mathbf{Z})$ et terminer l'algorithme.

Sous-algorithme *RED(k,l)*

Si $|\mu_{k,l}| \leqslant 0,5$, terminer l'algorithme.
Sinon soit q l'entier le plus proche de $\mu_{k,l}$, i.e.

$$q \leftarrow \lfloor \mu_{k,l} \rceil = \lfloor 0,5 + \mu_{k,l} \rfloor$$

Poser $\boldsymbol{b}_k \leftarrow \boldsymbol{b}_k - q\boldsymbol{b}_l$, $H_k \leftarrow H_k - qH_l$, $\mu_{k,l} \leftarrow \mu_{k,l} - q$ et pour $i = 1, \ldots, l-1$, poser $\mu_{k,i} \leftarrow \mu_{k,i} - q\mu_{l,i}$ et terminer le sous-algorithme.

Sous-algorithme *ECH(k)*

Echanger les vecteurs $\boldsymbol{b}_k$ et $\boldsymbol{b}_{k-1}$, ainsi que H_k et H_{k-1}, et si $k > 2$, pour $j = 1, \ldots, k-2$, echanger $\mu_{k,j}$ et $\mu_{k-1,j}$.
Puis poser (dans cet ordre) $\mu \leftarrow \mu_{k,k-1}$, $B \leftarrow B_k + \mu^2 B_{k-1}$, $\mu_{k,k-1} \leftarrow \mu B_{k-1}/B$, $\boldsymbol{b} \leftarrow \boldsymbol{b}^\star_{k-1}$, $\boldsymbol{b}^\star_{k-1} \leftarrow \boldsymbol{b}^\star_k + \mu\boldsymbol{b}$, $\boldsymbol{b}^\star_k \leftarrow -\mu_{k,k-1}\boldsymbol{b}^\star_k + (B_k/B)\boldsymbol{b}$ et $B_{k-1} \leftarrow B$.
Finalement, pour $i = k+1, \ldots, k_{max}$, poser (dans cet ordre) $t \leftarrow \mu_{i,k}$, $\mu_{i,k} \leftarrow \mu_{i,k-1} - \mu t$, $\mu_{i,k} \leftarrow \mu_{i,k}\mu_{k,k-1} + t$ et terminer le sous-algorithme

Il est facile de voir qu'au début de l'étape 4 de l'algorithme LLL, les conditions LLL de sont satisfaites pour tout $i \leqslant k-1$. Il en résulte que si $k > n$ (c'est-à-dire si k=n+1) les $\boldsymbol{b}_i$ forment une base LLL-réduite du réseau. De plus les opérations effectuées sur les $\boldsymbol{b}_i$ sont des opérations élémentaires, et donc la base finalement obtenue est bien une base du réseau L.
Cependant, il faut vérifier la terminaison de l'algorithme, et que le résultat est obtenu en un temps polynomial.
Pour $0 \leqslant i \leqslant n$, posons

$$d_i = det((\boldsymbol{b}_r.\boldsymbol{b}_s)_{1 \leqslant r,s \leqslant i})$$

On a alors

$$d_i = \prod_{1 \leqslant j \leqslant i} B_j$$

En particulier, on a $d_0 = 1$, $d_i > 0$ et $d_n = d(L)^2$.

Proposition 4 *Les d_i sont bornés inférieurement.*

$\diamondsuit$ Introduisons

$$m(L) = min\{\|\boldsymbol{x}\|^2 \mid x \in L,\ \boldsymbol{x} \neq \mathbf{0}\}$$

On interprète alors d_i comme le carré du déterminant du réseau engendré par les vecteurs $\boldsymbol{b}_1 \ldots \boldsymbol{b}_i$ dans l'espace vectoriel $\sum_{j=1}^{i} \mathbf{R}\boldsymbol{b}_j$.
Ce réseau contient alors un vecteur non nul $\boldsymbol{x}$ tel que :

$$\|\boldsymbol{x}\|^2 \leqslant (i^2/\pi) d_i^{1/i}$$

d'où

$$d_i \geqslant (\pi/i^2)^i m(L)^i \diamondsuit$$

Posons :

$$D = \prod_{1 \leqslant i \leqslant n} d_i$$

Cette quantité ne peut être modifiée que si l'un des B_i est modifié, c'est-à-dire dans l'algorithme ECH. Mais dans cet algorithme, les d_i sont inchangés pour $i < k-1$ et $i \geqslant k$. De plus, d'après la condition testée dans l'étape 3, d_{k-1} est multiplié par un facteur compris entre 0 et 3/4. D'autre part, D est borné inférieurement et donc le sous-algorithme ECH n'est exécuté qu'un nombre fini de fois, et donc k ne peut décroître qu'un nombre fini de fois et l'algorithme se termine.

Théorème 4 *Soit $L \subset \mathbf{Z}^n$ un réseau de base $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$, et soit $B \in \mathbf{R}$, $B \geqslant 2$ tel que pour $1 \leqslant i \leqslant n$, on ait $\|\boldsymbol{b}_i\|^2 \leqslant B$. Alors le nombre d'opérations nécessaires à la réduction de la base initiale en une base LLL-réduite par l'algorithme LLL est en $O(n^4 log(B))$.*

3 Le théorème de Coppersmith

On ne sait pas extraire des racines e-ièmes sans factoriser n. Plus généralement, on ne sait pas résoudre une équation polynomiale modulo n sans factoriser n (et utiliser ensuite les restes chinois). En 1996, Coppersmith a démontré, à l'aide de l'algorithme LLL, que l'on pouvait trouver efficacement toutes les petites solutions d'équations polynomiales sans factoriser n :

Théorème 5 *(Coppersmith) : Soient $P(x)$ un polynôme unitaire de degré δ à coefficients entiers, et n un entier de factorisation inconnue. Alors on peut trouver en temps polynomial en $(logn, \delta)$ tous les entiers x_0 tels que $P(x_0) \equiv 0(\mathrm{mod} n)$ et $|x_0| \leqslant n^{1/\delta}$*

♢ L'idée de Coppersmith est de réduire le problème de la recherche des petites racines modulaires au problème (facile) de la résolution d'équations polynomiales sur $\mathbb{Z}$. Plus précisément, Coppersmith utilise la réduction de réseau pour trouver une équation polynomiale (sur $\mathbb{Z}$) satisfaite par toutes les petites racines modulaires de P. Intuitivement, on essaie de linéariser toutes les équations de la forme $x^i P(x)^j \equiv 0(\mathrm{mod} n)$ pour des valeurs entières appropriées de i et j. De telles équations sont satisfaites par n'importe quelle solution de $P(x) \equiv 0(\mathrm{mod} n)$. Les petites solutions x_0 donnent des solutions inhabituellement petites du système linéaire obtenu. Pour transformer des équations modulaires en des équations entières, on utilise le lemme élémentaire suivant, avec la notation $||r(x)|| = \sqrt{\sum a_i^2}$:

Lemme : *Soit $r(x) \in \mathbb{Q}[x]$ un polynôme de degré $< m$ et X un nombre positif. Supposons $||r(xX)|| < 1/\sqrt{m}$. Si $r(x_0) \in \mathbb{Z}$, avec $|x_0| < X$ alors $r(x_0) = 0$ sur $\mathbb{Z}$.*

Ce lemme provient du fait que tout entier suffisamment petit est nécessairement nul. Fixons à présent un paramètre h et considérons les $m = (h+1)\delta$ polynômes $q_{u,v}(x) = x^u (P(x/n))^v$ où $0 \leqslant u \leqslant \delta - 1$ et $0 \leqslant v \leqslant h$. Remarquons que l'évaluation de $q_{u,v}(x)$ en n'importe quelle racine x_0 de $P(x)$ modulo n est nécessairement un entier. Et cela reste vrai pour toute combinaison linéaire à coefficients entiers des $q_{u,v}(x)$. Si une telle combinaison satisfait en outre

$||r(xX)|| < 1/\sqrt{m}$ alors le lemme nous assure que la résolution de l'équation $r(x) = 0$ sur $\mathbb{Z}$ fournira toutes les racines de $P(x)$ modulo n inférieures à X en valeur absolue. Cela suggère la recherche d'un vecteur court dans le réseau correspondant aux $q_{u,v}(xX)$. Plus précisément, définissons la matrice $m \times m$ M dont la $i$ème ligne est constituée des coefficients de $q_{u,v}(xX)$, en commençant par les termes de plus bas degré, avec $v = [(i-1)/\delta]$ et $u = (i-1) - \delta v$. M est triangulaire inférieure et un calcul élémentaire montre que $vol(M) = X^{m(m-1)/2} n^{-mh/2}$. Appliquons une réduction LLL au réseau total engendré par les lignes de M. Le premier vecteur de la base réduite obtenue correspond un polynôme de la forme $r(xX)$, qui vérifie

$$||r(xX)|| \leqslant 2^{(m-1)/4} vol(M)^{1/m} = 2^{(m-1)/4} X^{(m-1)/2} n^{-h/2}$$

Rappelons qu'il faut $||r(xX)|| < 1/\sqrt{n}$ pour appliquer le lemme. Par conséquent, pour tout h donné, la méthode garantit de trouver toutes les racines modulaires inférieures à X si :

$$X \leqslant \frac{1}{\sqrt{2}} n^{h/(m-1)} m^{-1/(m-1)}$$

La limite de cette borne supérieure , lorsque h tend vers l'infini est $\frac{1}{\sqrt{2}} n^{1/\delta}$. On en déduit le résultat par un choix approprié de h. $\diamondsuit$

4 Factorisation dans $\mathbf{Z}[X]$

Dans cette partie, nous allons exposer une méthode de factorisation dans $\mathbf{Z}[X]$. Cet algorithme va utiliser l'algorithme de Berlekamp, ou tout autre algorithme pour factoriser dans un $(\mathbf{Z}/p\mathbf{Z})[X]$ et ensuite on utillisera LLL pour pouvoir à partir d'un facteur irréductible dans $(\mathbf{Z}/p\mathbf{Z})[X]$ obtenir un facteur irréductible dans $\mathbf{Z}[X]$. On va donc tout d'abord voir comment cette remontée fonctionne et comment elle peut être effectuée par l'algorithme LLL, et ensuite présenter l'algorithme complet de factorisation.

4.1 Lien avec les réseaux

On désigne dans toute cette partie par p un nombre premier et k un entier positif. Soit $f \in \mathbf{Z}[X]$ de degré $n > 0$, et $h \in \mathbf{Z}[X]$ ayant les propriétés suivantes :

$$h \text{ est unitaire,} \tag{1}$$

$$(h \bmod p^k) \text{ divise } (f \bmod p^k) \text{ dans } (\mathbf{Z}/p^k\mathbf{Z})[X], \tag{2}$$

$$(h \bmod p) \text{ est irréductible dans } (\mathbf{Z}/p\mathbf{Z})[X], \tag{3}$$

$$(h \bmod p)^2 \text{ ne divise pas } (f \bmod p) \textit{dans } (\mathbf{Z}/p\mathbf{Z})[X]. \tag{4}$$

On désigne par l le degré de h. On a alors $0 < l \leqslant n$.

Proposition 5 *Le polynôme f a un facteur irréductible, unique au signe près, h_0 dans $\mathbf{Z}[X]$ tel que $(h \bmod p^k)$ divise $(h_0 \bmod p^k)$. D'autre part, si g divise f dans $\mathbf{Z}[X]$, les trois assertions suivantes sont équivalentes :*

– $(h \bmod p)$ *divise* $(g \bmod p)$ *dans* $(\mathbf{Z}/p\mathbf{Z})[X]$
– $(h \bmod p^k)$ *divise* $(g \bmod p^k)$ *dans* $(\mathbf{Z}/p^k\mathbf{Z})[X]$
– h_0 *divise* g *dans* $\mathbf{Z}[X]$
En particulier, $(h \bmod p^k)$ *divise* $(h_0 \bmod p^k)$ *dans* $(\mathbf{Z}/p^k\mathbf{Z})[X]$.

$\diamondsuit$ L'existence de h_0 découle de (2) et (3), l'unicité au signe près, de (4). Les implications $(ii) \implies (i)$ et $(iii) \implies (i)$ sont claires. Maintenant supposons (i), montrons (iii) et (ii) Il résulte de (i) et (4) que $(h \bmod p)$ ne divise pas $(f/g \bmod p)$ dans $\mathbf{F}_p[X]$, donc h_0 ne divise pas $(f/g \bmod p)$ dans $\mathbf{Z}[X]$, donc divise g. Ceci prouve (iii). D'après (3), les polynômes $(h \bmod p)$ et $(f/g \bmod p)$ sont premiers entre eux dans $\mathbf{F}_p[X]$, donc on a :

$$(\lambda_1 \bmod p).(h \bmod p) + (\mu_1 \bmod p).(f/g \bmod p) = 1$$

où $\lambda_1, \mu_1 \in \mathbf{Z}[X]$. Par conséquent $\lambda_1 h + \mu_1 f/g = 1 - pv_1$, avec $v_1 \in \mathbf{Z}[X]$. En multipliant par $1 + pv_1 + p^2 v_1^2 + \ldots + p^{k-1} v_1^{k-1}$ et par g, on obtient :

$$\lambda_2 h + \mu_2 f \equiv g \bmod p^k \mathbf{Z}[X]$$

où $\lambda_2, \mu_2 \in \mathbf{Z}[X]$. Le terme de gauche, pris modulo p^k est divisible par $(h \bmod p^k)$,c'est vrai pour le terme de droite. Ceci prouve (ii). La dernière assertion en découle en prenant $g = h_0$. $\diamondsuit$

Pour la suite, on fixe un entier $m \geqslant l$, et soit L l'ensemble des polynômes de $\mathbf{Z}[X]$, de degré inférieur ou égal à m, dont l'image canonique dans $(\mathbf{Z}/p^k\mathbf{Z})[X]$ est divisible par $(h \bmod p^k)$. L est alors un réseau de $\mathbf{R}^{m+1}$ en identifiant $\sum_0^m a_i X^i$ et $(a_0, \ldots, a_m)$, dont une base est :

$$\{p^k X^i \mid 0 \leqslant i < l\} \cup \{hX^j \mid 0 \leqslant j \leqslant m - l\}$$

d'où $d(L) = p^{kl}$. On définit de plus la norme $\| \sum_0^m a_i X^i \|$ comme la norme euclidienne de $(a_0, \ldots, a_m)$.

Proposition 6 *Soit* $b \in L$ *tel que*

$$p^{kl} > \|f\|^m \|b\|^n \tag{5}$$

Alors b *est divisible par* h_0 *dans* $\mathbf{Z}[X]$*, et en particulier* $f \wedge b \neq 1$.

$\diamondsuit$ On peut supposer $b \neq 0$. Posons $g = pgcd(f, b)$. D'après la proposition 5, il suffit de montrer que $(h \bmod p)$ divise $(g \bmod p)$. Supposons que ce ne soit pas le cas. Alors par (3), on a :

$$\lambda_3 h + \mu_3 g = 1 - pv_3 \tag{6}$$

où $\lambda_3, \mu_3, v_3 \in \mathbf{Z}[X]$. Nous allons en tirer une contradiction. Soit $e = deg(g)$ et $m' = deg(b)$. On a $0 \leqslant e \leqslant m' \leqslant m$. Posons

$$M = \{\lambda f + \mu b : \lambda, \mu \in \mathbf{Z}[X], deg(\lambda) < m' - e, deg(\mu) < n - e\}$$

$$\subset \mathbf{Z} + \mathbf{Z} \cdot X^e + \mathbf{Z} \cdot X^{e+1} + \ldots + \mathbf{Z} \cdot X^{n+m'-e-1}$$

Soit M' la projection de M sur

$$\mathbf{Z} \cdot X^e + \mathbf{Z} \cdot X^{e+1} + \ldots + \mathbf{Z} \cdot X^{n+m'-e-1}$$

Supposons que $\lambda f + \mu g$ se projette sur 0 dans M', avec λ et μ comme dans la définition de M. Alors $deg(\lambda f + \mu b) < e$, or g divise $\lambda f + \mu b$, donc $\lambda f + \mu b = 0$ De $\lambda \cdot (f/g) = -\mu \cdot (b/g)$ et $pgcd(f/g, b/g) = 1$, on déduit que f/g divise μ. Or $deg(\mu) < n - e = deg(f/g)$, donc $\mu = 0$, et aussi $\lambda = 0$.
Ceci montre que les projections des

$$\{X^i f : 0 \leqslant i < m' - e\} \cup \{X^j b : 0 \leqslant j < n - e\}$$

sont linéairement indépendantes. Comme ces projections engendrent M', il résulte que M' est un réseau de rang $n + m' - 2e$. D'après l'inégalité d'Hadamard et (5), on obtient

$$d(M') \leqslant \|f\|^{m'-e} \cdot \|b\|^{n-e} \leqslant \|f\|^m \cdot \|b\|^n < p^{kl}. \tag{7}$$

On déduit de (6) que

$$\{v \in M : deg(v) < e + l\} \subset p^k \mathbf{Z}[X] \tag{8}$$

Ainsi, si on choisit une base $b_e, b_{e+1}, \ldots, b_{n+m'-e-1}$, de M' avec $deg(b_j) = j$, alors les coefficients dominants de $b_e, b_{e+1}, \ldots, b_{n+m'-e-1}$ sont divisibles par p^k. [Remarquer que $e + l - 1 \leqslant n + m' - e - 1$ car g divise b et $(h \bmod p)$ divise $(f/g \bmod p)$]. Comme $d(M')$ est égal à la valeur absolue du produit des coefficients dominants de $b_e, b_{e+1}, \ldots, b_{n+m'-e-1}$ on a $d(M') \geqslant p^{kl}$. En combinant avec (7), on a la contradiction voulue. Pour montrer (8), soit $v \in M, deg(v) < e + l$. Alors g divise v. En multipliant (6) par v/g et par $1 + pv_3 + p^2 v_3^2 + \ldots + p^{k-1} v_3^{k-1}$ on obtient

$$\lambda_4 h + \mu_4 v \equiv v/g \bmod p^k \mathbf{Z}[X] \tag{9}$$

où $\lambda_4, \mu_4 \in \mathbf{Z}[X]$. De $v \in M$ et $b \in L$ on déduit que $(v \bmod p^k)$ est divisible par $(h \bmod p^k)$, donc d'après (9), on a aussi $(v/g \bmod p^k)$ divisible par $(h \bmod p^k)$. Or $(h \bmod p^k)$ est de degré l de coefficient 1, alors que $(v/g \bmod p^k)$ est de degré $< e + l - e = l$, donc $v/g \equiv 0 \bmod p^k \mathbf{Z}[X]$, et $v \equiv 0 \bmod p^k \mathbf{Z}[X]$, d'où (8).
$\diamondsuit$

Proposition 7 *Sous les hypothèses précédentes, soit $b_1, \ldots, b_{m+1}$ une base LLL-réduite de L. On suppose :*

$$p^{kl} > 2^{mn/2} (\mathrm{C}_{2m}^m)^{n/2} \|f\|^{m+n} \tag{10}$$

Alors on a $d^o(h_0) \leqslant m$ ssi

$$\|b_1\| < \left(\frac{p^{kl}}{\|f\|^m}\right)^{1/n}$$

$\diamondsuit$ La condition est suffisante d'après la proposition 6. Pour la condition nécessaire, voir [1]
$\diamondsuit$

Proposition 8 *Supposons les hypothèses de la proposition précédente vérifiées, et supposons qu'en plus, il existe* $j \in \{1, \ldots, m+1\}$ *pour lequel*

$$\|b_j\| < (\frac{p^{kl}}{\|f\|^m})^{1/n} \tag{11}$$

et soit t *le plus grand des tels* j*. On a alors*

$$d^o(h_0) = m + 1 - t$$

$$h_0 = pgcd(b_1, \ldots, b_t)$$

et (11) est vraie pour $1 \leqslant j \leqslant t$*.*

$\diamondsuit$ Soit $J = \{j \in \{1, 2, \ldots, m+1\} : \text{on a (11)}\}$. D'après la proposition 6, on sait que h_0 divise b_j pour tout $j \in J$, donc si on pose

$$h_1 = pgcd(\{b_j : j \in J\})$$

alors h_0 divise h_1. Chaque b_j, $j \in J$ est divisible par h_1 et est de degré $\leqslant m$, donc appartient à

$$\mathbf{Z} \cdot h_1 + \mathbf{Z} \cdot h_1 X + \ldots + \mathbf{Z} \cdot h_1 X^{m - deg(h_1)}$$

Comme les b_j sont linéairement indépendants, on a :

$$\#J \leqslant m + 1 - deg(h_1) \tag{12}$$

D'après un résultat dû à Mignotte, voir [1], on a $\|h_0 X\| = \|h_0\| \leqslant (\mathrm{C}_{2m}^m)^{1/2} \cdot \|f\|$ pour tout $i \geqslant 0$. Pour $i = 0, 1, \ldots, m - deg(h_0)$ on a $h_0 X \in L$, donc d'après le théorème 2, on a :

$$\|b_j\| \leqslant 2^{m/2} \cdot (\mathrm{C}_{2m}^m)^{1/2} \cdot \|f\|$$

pour $1 \leqslant j \leqslant m + 1 - deg(h_0)$. D'après (10), ceci implique

$$\{1, 2, \ldots, m + 1 - deg(h_0)\} \subset J \tag{13}$$

De (12) et (13) et le fait que h_0 divise h_1, on déduit qu'il y a égalité dans (12) et (13) et que

$$deg(h_0) = deg(h_1) = m + 1 - t$$

$$J = \{1, 2, \ldots, t\}$$

Il reste à montrer que h_0 est égal à h_1, au signe près, et pour cela il suffit que h_1 soit primitif. Choisissons $j \in J$, et soit d_j le contenu de b_j. Alors b_j/d_j est divisible par h_0 et $h_0 \in L$, donc $b_j/d_j \in L$. Or b_j appartient à une base de L, donc $d_j = 1$ et b_j est primitif, et ceci reste valable pour le facteur h_1 de b_j

$\diamondsuit$

4.2 Sous-algorithmes utiles

Nous allons maintenant pouvoir donner un algorithme permettant de factoriser un polynôme $f \in \mathbf{Z}[X]$ de degré $n > 0$, en facteurs irréductibles dans $\mathbf{Z}[X]$. Nous avons cependant besoin de trois algorithmes préliminaires.

4.2.1 Sous-algorithme 1

On se donne $f,\ n,\ p > 0$ premier, $k \geqslant 0$ et un polynôme $h \in \mathbf{Z}[X]$ de degré $l > 0$ qui vérifient les conditions de la proposition 3. On suppose que les coefficients de h sont réduits modulo p^k,afin que l'on ait :

$$\|h\|^2 \leqslant 1 + lp^{2k}$$

Soit $m \geqslant l$ fixé et supposons que l'on a :

$$p^{kl} > 2^{mn/2}(\mathbf{C}_{2m}^{m})^{(n/2)}\|f\|^{m+n}$$

On a un algorithme qui détermine h_0 comme auparavant si l'on a $d^o h_0 \leqslant m$. Pour cela, on introduit le réseau L déjà vu :

$$\{p^k X^i \mid 0 \leqslant i < l\} \cup \{hX^j \mid 0 \leqslant j \leqslant m - l\}$$

On réduit cette base par l'algorithme LLL et on a alors deux cas :
– $\|b_1\| \geqslant (p^{kl}/\|f\|^m)^{1/n}$. Alors par la proposition 7, on a $d^o h_0 > m$.
– $\|b_1\| < (p^{kl}/\|f\|^m)^{1/n}$. Alors par la proposition 7, on a $d^o h_0 \leqslant m$, et

$$h_0 = pgcd(b_1, \ldots, b_t)$$

avec t comme dans la proposition 8.

Proposition 9 *Le nombre d'opérations effectuées par le sous-algorithme* 1 *est polynomial avec une complexité en* $O(m^4 k \log p)$

4.2.2 Sous-algorithme 2

On se donne maintenant un polynôme f de degré n, un nombre premier p. On suppose de plus que l'on a un polynôme $h \in \mathbf{Z}[X]$, de degré l, satisfaisant aux conditions de la proposition 5 avec $k = 1$, avec des coefficients réduits modulo p.
On va maintenant utiliser le sous-algorithme 1, pour calculer h_0, le facteur irréductible de f tel que $(h \bmod p)$ divise $(h_0 \bmod p)$. On va tout d'abord commencer à chercher une valeur de k convenable afin d'appliquer les résultats précédents.
On peut supposer $l < n$, sinon $f = h_0$ et l'algorithme termine. On cherche alors la plus petite valeur de k telle que l'on ait les hypothèses de la proposition avec $m = n - 1$, c'est-à-dire tel que l'on ait :

$$p^{kl} > 2^{(n-1)n/2}(\mathrm{C}_{2(n-1)}^{n-1})^{n/2}\|f\|^{2n-1}$$

On modifie alors h sans changer $(h \bmod p)$ pour que l'on ait dans $(\mathbf{Z}/p\mathbf{Z})[X]$:

$$(h \bmod p^k) \mid (f \bmod p^k)$$

en gardant toujours les coefficients de h réduits modulo p.
On va maintenant exécuter le sous-algorithme 1 pour les valeurs $m = [(n-1)/2^u], [(n-1)/2^u], \ldots, [(n-1)/2], (n-1)$ où u désigne le plus grand entier pour lequel on a $l \leqslant (n-1)/2^u$, et $[x]$ désigne la partie entière de x, et on s'arrête dès que le sous-algorithme 1 trouve h_0 pour une telle valeur de m.
Si ceci n'arrive pas, alors $d^o h_0 > (n-1)$ et donc $h_0 = f$.

Proposition 10 *Le nombre d'opérations effectuées par le sous-algorithme* 2 *est polynomial avec une complexité en* $O(m_0(n^5 + n^4 \log|f| + n^3 \log p))$ *où* $m_0 = d^o h_0$.

4.2.3 Sous-algorithme 3 : algorithme de Berlekamp

On redonne ici sans démonstration l'algorithme de Berlekamp qui permet pour des petites valeurs de p premier de factoriser $f \in (\mathbf{Z}/p\mathbf{Z})[X]$ de degré n. En pratique, cet algorithme reste efficace pour des valeurs de p inférieurs à 100. Au delà, il convient d'utiliser l'algorithme de Cantor-Zassenhaus.

Algorithme 2 *(Algorithme de Berlekamp pour p petit)*
Étant donné un polynôme $f \in (\mathbf{Z}/p\mathbf{Z})[X]$ de degré n, sans racine, l'algorithme suivant renvoie la factorisation de f en facteurs irréductibles.

1 *Calcul de Q*
Calculer inductivement pour $0 \leqslant k < n$, les éléments $q_{i,k} \in \mathbf{Z}/p\mathbf{Z}$ tels que l'on ait :

$$X^{pk} \equiv \sum_{0 \leqslant i < n} q_{i,k} X^i \pmod{A(X)}$$

2 *Calcul de* $\ker Q - I$
Calculer les vecteurs $V_1, \ldots, V_r$ de $\ker Q - I$. Alors r sera le nombre de facteurs irréductibles de f et on a $V_1 = (1, 0, \ldots, 0)^t$. Poser $E \leftarrow \{f\}$, $k \leftarrow 1$ et $j \leftarrow 1$ (E designe un ensemble de polynômes dont le produit est égal à f, k son cardinal et j l'indice du vecteur du noyau utilisé).

3 *Fin ?*
Si $k = r$, renvoyer E et terminer l'algorithme. Sinon, poser $j \leftarrow j + 1$ et soit $T(X)$ le polynôme correspondant au vecteur V_j (i.e. $T(X) = \sum_{0 \leqslant i < n} (V_j)_i X^i$.

4 *Scindage*
Pour chaque élément $B \in E$ de degré > 1, calculer

$$F = \{G(X) = (B(X) \wedge T(X) - s) \mid s \in \mathbf{Z}/p\mathbf{Z} \text{ et } d^o G(X) \geqslant 1\}$$

Poser $E \leftarrow (E \backslash \{B\}) \cup F$ et $k \leftarrow k - 1 + \sharp F$.
Si $k = r$, renvoyer E et terminer l'algorithme. Sinon aller à l'étape **3**.

4.3 Algorithme final de factorisation dans $\mathbf{Z}[X]$

On va maintenant décrire un algorithme qui étant donné un polynôme $f \in \mathbf{Z}[X]$ primitif de degré $n > 0$, retourne sa décomposition en facteurs irréductibles dans $\mathbf{Z}[X]$.
Pour cela, on commence par calculer le discriminant de f par le calcul du résultant de f et f'.

- On suppose que l'on a $R(f, f') \neq 0$
  Ensuite on détermine le plus nombre premier p tel que $p \not| R(f, f')$ et à l'aide de l'algorithme de Berlekamp, on décompose ($f \bmod p$) en facteurs irréductibles dans $(\mathbf{Z}/p\mathbf{Z})[X]$. On a alors la propriété (4) pour tout diviseur ($h \bmod p$) de ($f \bmod p$).
  Ensuite on va calculer les h_0 correspondants aux ($h \bmod p$), en utilisant le sous-algorithme 2, et en faisant bien attention au fur et à mesure de supprimer tous les ($h \bmod p$) qui divisent un même h_0.
- On suppose que l'on a $R(f, f') = 0$
  On calcule alors g le *pgcd* de f et f' dans $\mathbf{Z}[X]$. On pose $f_0 = f/g$. Alors f_0 n'a pas de facteurs multiples dans $\mathbf{Z}[X]$ et on se ramène au cas précédent.

Ensuite comme g et f_0 ont les mêmes facteurs irréductibles dans $\mathbf{Z}[X]$, un nombre limité de divisions permet de les trouver et de terminer la factorisation de f.

Théorème 6 *L'algorithme décrit ci-dessus factorise un polynôme f de degré $n > 0$ en produit de facteurs irréductibles avec une complexité $O(n^6 + n^5 \log |f|)$.*

Références

[1] A.K.Lenstra, H.W.Lenstra, L.Lovász, *Factoring Polynomials with Rationnals Coefficients*, Math. Ann. 261, 515-534, 1982

[2] P.Samuel, *Théorie algébrique des nombres*, Hermann

[3] H.Cohen, *A Course in Computational Algebric Number Theory*, Springer-Verlag, 1995

[4] M.Grötschel, L.Lovász, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, 1993

[5] J.Stern, L.Gramboulan, P.Nguyen, D.Pointcheval, *Conception et preuves d'algorithmes cryptographiques*, Cours de magistère MMFAI École Normale Supérieure, 2001-2002