Une introduction à la réduction de réseaux

Damien Stehlé

ÉNS de Lyon

JNCF, Mai 2013

Les réseaux euclidiens en informatique

- Un objet mathématique très classique.
- Mais toujours assez mal compris.

- Ses aspects calculatoires ont été étudiés depuis env. 30 ans.
- Mais ne nombreuses questions calculatoires restent ouvertes.
  $\Rightarrow$ Peu d'algorithmes, souvent pas très bien compris.

- Utile dans de nombreux domaines
  - théorie des communications
  - cryptographie
  - arithmétique des ordinateurs
  - théorie algorithmique des nombres
  - etc

Les réseaux euclidiens en informatique

- Un objet mathématique très classique.
- Mais toujours assez mal compris.

- Ses aspects calculatoires ont été étudiés depuis env. 30 ans.
- Mais ne nombreuses questions calculatoires restent ouvertes.
  $\Rightarrow$ Peu d'algorithmes, souvent pas très bien compris.

- Utile dans de nombreux domaines
  - théorie des communications
  - cryptographie
  - arithmétique des ordinateurs
  - théorie algorithmique des nombres
  - etc

Les réseaux euclidiens en informatique

- Un objet mathématique très classique.
- Mais toujours assez mal compris.

- Ses aspects calculatoires ont été étudiés depuis env. 30 ans.
- Mais ne nombreuses questions calculatoires restent ouvertes.
  $\Rightarrow$ Peu d'algorithmes, souvent pas très bien compris.

- Utile dans de nombreux domaines
  - théorie des communications
  - cryptographie
  - arithmétique des ordinateurs
  - théorie algorithmique des nombres
  - etc

Objectifs

- Une introduction aux aspects algorithmiques de la réduction de réseaux
- Quelques exemples d'applications

1er cours :

- Définitions mathématiques
- Problème du vecteur le plus court
- L'algorithme LLL

2nd cours :

- L'algorithme BKZ
- Applications en cryptanalyse
- Algorithmes algébriques-numériques de réduction

Objectifs

- Une introduction aux aspects algorithmiques de la réduction de réseaux
- Quelques exemples d'applications

1er cours :

- Définitions mathématiques
- Problème du vecteur le plus court
- L'algorithme LLL

2nd cours :

- L'algorithme BKZ
- Applications en cryptanalyse
- Algorithmes algébriques-numériques de réduction

Objectifs

- Une introduction aux aspects algorithmiques de la réduction de réseaux
- Quelques exemples d'applications

1er cours :

- Définitions mathématiques
- Problème du vecteur le plus court
- L'algorithme LLL

2nd cours :

- L'algorithme BKZ
- Applications en cryptanalyse
- Algorithmes algébriques-numériques de réduction

Mes sources préférées

Introductions aux aspects calculatoires des réseaux, avec sensibilité marquée vers la cryptographie et la théorie de la complexité.

- Notes de cours d'Oded Regev :
  `http://www.cims.nyu.edu/~regev/teaching/`
- Notes de cours de Daniele Micciancio :
  `http://cseweb.ucsd.edu/~daniele/classes.html/`

Plan du 1er cours

1. Définitions et propriétés élémentaires
2. Problèmes calculatoires
3. Résoudre SVP
4. Concept de réduction de réseau
5. L'algorithme LLL

Plan du 1er cours

Première définition

Définition algébrique

Un réseau L est un sous-groupe additif discret d'un $\mathbb{R}^n$.

- Sous-groupe additif :

  L est stable par combinaisons linéaires entières

- Discret : aucun point d'accumulation.

  Pour tout $\mathbf{b} \in L$, il existe une boule ouverte autour de $\mathbf{b}$ qui ne contient que $\mathbf{b}$

Exemple : Tout sous-groupe de $\mathbb{Z}^d$ $\Rightarrow$ on parle de réseau entier.

Contre-exemple : $S = \mathbb{Z} + \alpha\mathbb{Z}$ avec $\alpha \notin \mathbb{Q}$:
si $(p_k/q_k)_k$ est la suite des fractions continues de $\sqrt{2}$, alors

$$\begin{aligned} p_k - q_k\sqrt{2} &\to_k 0, \\ p_k - q_k\sqrt{2} &\in S \setminus 0. \end{aligned}$$

Première définition

Définition algébrique

Un réseau L est un sous-groupe additif discret d'un $\mathbb{R}^n$.

- Sous-groupe additif :

  L est stable par combinaisons linéaires entières

- Discret : aucun point d'accumulation.

  Pour tout $\mathbf{b} \in L$, il existe une boule ouverte autour de $\mathbf{b}$ qui ne contient que $\mathbf{b}$

Exemple : Tout sous-groupe de $\mathbb{Z}^d$ $\Rightarrow$ on parle de réseau entier.

Contre-exemple : $S = \mathbb{Z} + \alpha\mathbb{Z}$ avec $\alpha \notin \mathbb{Q}$:
si $(p_k/q_k)_k$ est la suite des fractions continues de $\sqrt{2}$, alors

$$\begin{aligned} p_k - q_k\sqrt{2} &\to_k 0, \\ p_k - q_k\sqrt{2} &\in S \setminus 0. \end{aligned}$$

Première définition

Définition algébrique

Un réseau L est un sous-groupe additif discret d'un $\mathbb{R}^n$.

- Sous-groupe additif :

  L est stable par combinaisons linéaires entières

- Discret : aucun point d'accumulation.

  Pour tout $\mathbf{b} \in L$, il existe une boule ouverte autour de $\mathbf{b}$ qui ne contient que $\mathbf{b}$

Exemple : Tout sous-groupe de $\mathbb{Z}^d$ $\Rightarrow$ on parle de réseau entier.

Contre-exemple : $S = \mathbb{Z} + \alpha\mathbb{Z}$ avec $\alpha \notin \mathbb{Q}$:
si $(p_k/q_k)_k$ est la suite des fractions continues de $\sqrt{2}$, alors

$$\begin{aligned} p_k - q_k\sqrt{2} &\to_k 0, \\ p_k - q_k\sqrt{2} &\in S \setminus 0. \end{aligned}$$

Première définition

Définition algébrique

Un réseau L est un sous-groupe additif discret d'un $\mathbb{R}^n$.

- Sous-groupe additif :

  L est stable par combinaisons linéaires entières

- Discret : aucun point d'accumulation.

  Pour tout $\mathbf{b} \in L$, il existe une boule ouverte autour de $\mathbf{b}$ qui ne contient que $\mathbf{b}$

Exemple : Tout sous-groupe de $\mathbb{Z}^d$ $\Rightarrow$ on parle de réseau entier.

Contre-exemple : $S = \mathbb{Z} + \alpha\mathbb{Z}$ avec $\alpha \notin \mathbb{Q}$:
si $(p_k/q_k)_k$ est la suite des fractions continues de $\sqrt{2}$, alors

$$\begin{array}{rcl} p_k - q_k\sqrt{2} & \to_k & 0, \\ p_k - q_k\sqrt{2} & \in & S \setminus 0. \end{array}$$

Première définition

Définition algébrique

Un réseau L est un sous-groupe additif discret d'un $\mathbb{R}^n$.

- Sous-groupe additif :

  L est stable par combinaisons linéaires entières

- Discret : aucun point d'accumulation.

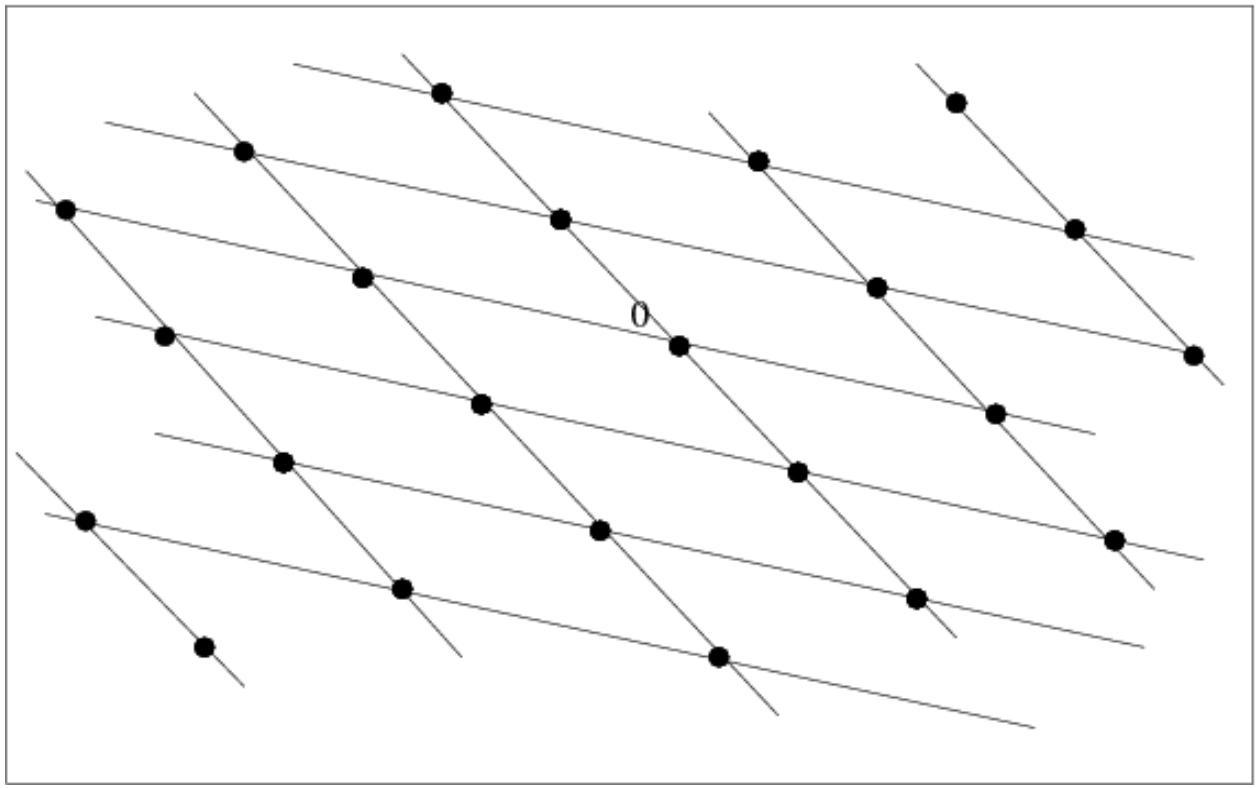
  Pour tout $\mathbf{b} \in L$, il existe une boule ouverte autour de $\mathbf{b}$ qui ne contient que $\mathbf{b}$

Exemple : Tout sous-groupe de $\mathbb{Z}^d$ $\Rightarrow$ on parle de réseau entier.

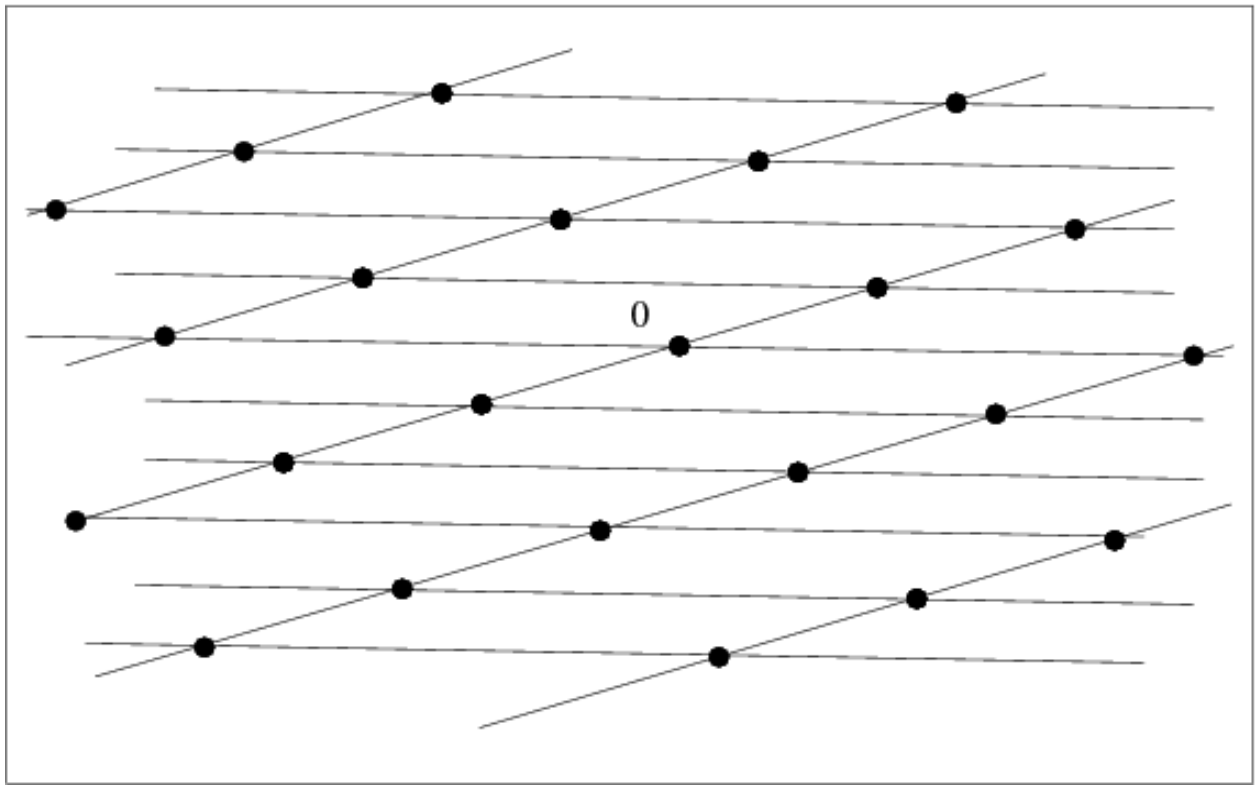
Contre-exemple : $S = \mathbb{Z} + \alpha\mathbb{Z}$ avec $\alpha \notin \mathbb{Q}$:
si $(p_k/q_k)_k$ est la suite des fractions continues de $\sqrt{2}$, alors

$$\begin{aligned} p_k - q_k\sqrt{2} &\to_k 0, \\ p_k - q_k\sqrt{2} &\in S \setminus 0. \end{aligned}$$

Un réseau de dimension 2



Le même réseau



Définition équivalente

Définition géométrique

Un réseau L est l'ensemble des combinaisons linéaires entières de vecteurs linéairement indépendants de $\mathbb{R}^n$.

$$L = \sum_{1 \le i \le d} \mathbb{Z}\mathbf{b}_i = \{ \sum_{1 \le i \le d} x_i \mathbf{b}_i,\ x_i \in \mathbb{Z} \} = B \cdot \mathbb{Z}^d,$$

où les $\mathbf{b}_i \in \mathbb{R}^n$ sont linéairement indépendants,

et $B \in \mathbb{R}^{n \times d}$ est la matrice dont les colonnes sont les $\mathbf{b}_i$.

- $\mathbf{b}_1, \ldots, \mathbf{b}_d$ est une base de L. Elle n'est pas unique.
- Dimension de l'espace ambiant : n.
- Dimension du réseau : d.

Si $d = n$, on parle de réseau de plein rang.

Définition équivalente

Définition géométrique

Un réseau L est l'ensemble des combinaisons linéaires entières de vecteurs linéairement indépendants de $\mathbb{R}^n$.

$$L = \sum_{1 \leq i \leq d} \mathbb{Z}\mathbf{b}_i = \{ \sum_{1 \leq i \leq d} x_i \mathbf{b}_i,\ x_i \in \mathbb{Z} \} = B \cdot \mathbb{Z}^d,$$

où les $\mathbf{b}_i \in \mathbb{R}^n$ sont linéairement indépendants,

et $B \in \mathbb{R}^{n \times d}$ est la matrice dont les colonnes sont les $\mathbf{b}_i$.

- $\mathbf{b}_1, \ldots, \mathbf{b}_d$ est une base de L. Elle n'est pas unique.
- Dimension de l'espace ambiant : n.
- Dimension du réseau : d.

Si $d = n$, on parle de réseau de plein rang.

Définition équivalente

Définition géométrique

Un réseau L est l'ensemble des combinaisons linéaires entières de vecteurs linéairement indépendants de $\mathbb{R}^n$.

$$L = \sum_{1 \le i \le d} \mathbb{Z}\mathbf{b}_i = \{ \sum_{1 \le i \le d} x_i \mathbf{b}_i,\ x_i \in \mathbb{Z} \} = B \cdot \mathbb{Z}^d,$$

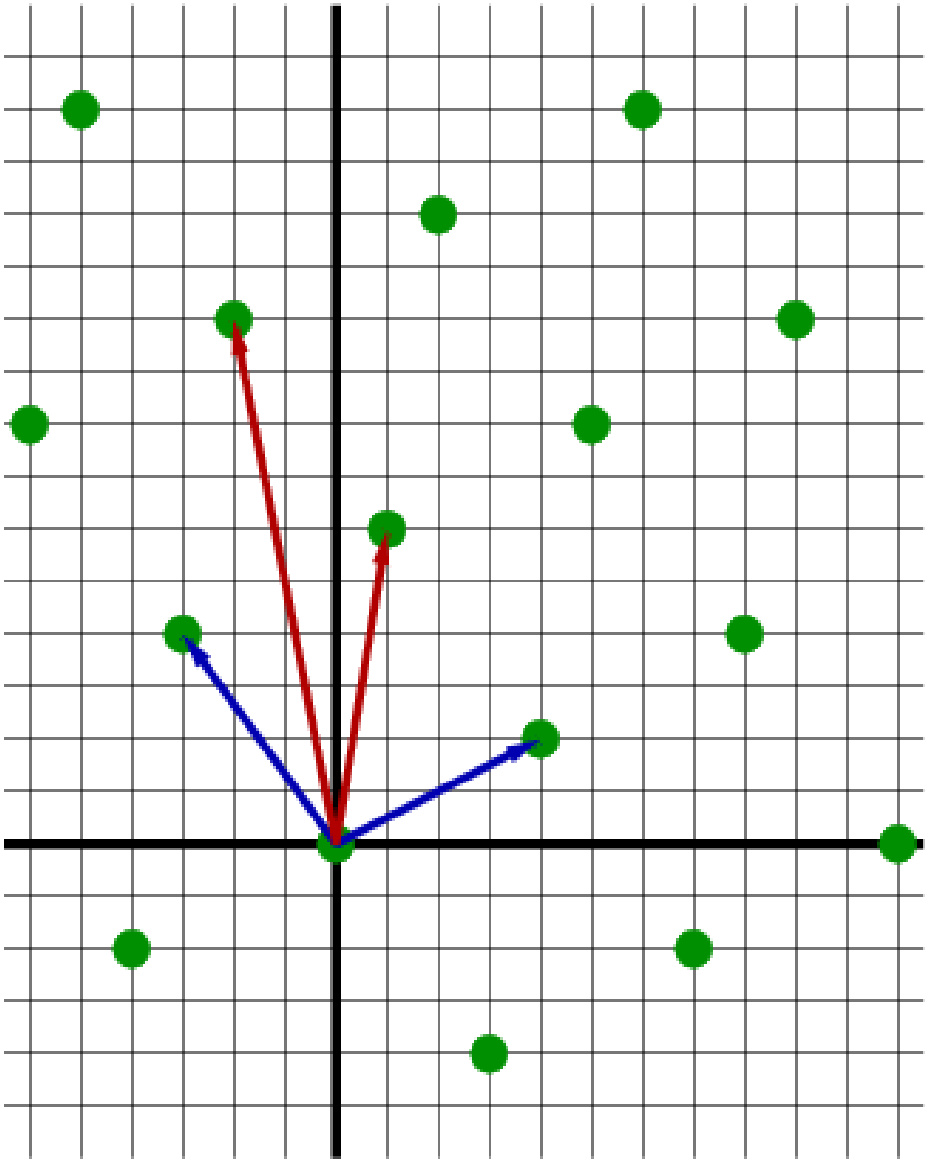
où les $\mathbf{b}_i \in \mathbb{R}^n$ sont linéairement indépendants,

et $B \in \mathbb{R}^{n \times d}$ est la matrice dont les colonnes sont les $\mathbf{b}_i$.

- $\mathbf{b}_1, \ldots, \mathbf{b}_d$ est une base de L. Elle n'est pas unique.
- Dimension de l'espace ambiant : n.
- Dimension du réseau : d.

Si $d = n$, on parle de réseau de plein rang.

Deux bases d'un réseau de dimension 2



Ensemble des bases d'un même réseau

Matrices unimodulaires

Une matrice $U \in \mathbb{Z}^{d \times d}$ est unimodulaire si elle est inversible dans $\mathbb{Z}^{d \times d}$, i.e., $U \in GL_d(\mathbb{Z})$.
De manière équivalente : $\det U = \pm 1$.

Unimodularité et bases d'un réseau

Deux bases $(\mathbf{b}_i)_{i \leq d}$ et $(\mathbf{c}_i)_{i \leq d}$ engendrent le même réseau

$$\Downarrow$$

$$\exists U \in \mathrm{GL}_d(\mathbb{Z}) : \quad (\mathbf{b}_i)_{i \leq d} \cdot U = (\mathbf{c}_i)_{i \leq d}.$$

Conséquences directes :

- Tout réseau de dimension ≥ 2 a une infinité de bases.
- Un réseau est un élément de $GL_d(\mathbb{R})/GL_d(\mathbb{Z})$.
- On peut se demander si des bases sont meilleures que d'autres.

Ensemble des bases d'un même réseau

Matrices unimodulaires

Une matrice $U \in \mathbb{Z}^{d\times d}$ est unimodulaire si elle est inversible dans $\mathbb{Z}^{d\times d}$, i.e., $U \in GL_d(\mathbb{Z})$.
De manière équivalente : $\det U = \pm 1$.

Unimodularité et bases d'un réseau

Deux bases $(\mathbf{b}_i)_{i\leq d}$ et $(\mathbf{c}_i)_{i\leq d}$ engendrent le même réseau

$$\Downarrow$$

$$\exists U \in \mathrm{GL}_d(\mathbb{Z}) : \quad (\mathbf{b}_i)_{i\leq d} \cdot U = (\mathbf{c}_i)_{i\leq d}.$$

Conséquences directes :

- Tout réseau de dimension ≥ 2 a une infinité de bases.
- Un réseau est un élément de $GL_d(\mathbb{R})/GL_d(\mathbb{Z})$.
- On peut se demander si des bases sont meilleures que d'autres.

Ensemble des bases d'un même réseau

Matrices unimodulaires

Une matrice $U \in \mathbb{Z}^{d\times d}$ est unimodulaire si elle est inversible dans $\mathbb{Z}^{d\times d}$, i.e., $U \in GL_d(\mathbb{Z})$.
De manière équivalente : $\det U = \pm 1$.

Unimodularité et bases d'un réseau

Deux bases $(\mathbf{b}_i)_{i\leq d}$ et $(\mathbf{c}_i)_{i\leq d}$ engendrent le même réseau

$$\Downarrow$$

$$\exists U \in \mathrm{GL}_d(\mathbb{Z}) : \quad (\mathbf{b}_i)_{i\leq d} \cdot U = (\mathbf{c}_i)_{i\leq d}.$$

Conséquences directes :

- Tout réseau de dimension ≥ 2 a une infinité de bases.
- Un réseau est un élément de $GL_d(\mathbb{R})/GL_d(\mathbb{Z})$.
- On peut se demander si des bases sont meilleures que d'autres.

Minima d'un réseau

Minimum d'un réseau

Tout réseau $L \neq 0$ contient un $\mathbf{b} \neq \mathbf{0}$ de norme euclidienne minimale.
La norme de ce vecteur est le minimum $\lambda_1(L)$:

$$\lambda_1(L) = \min\left(r : \mathcal{B}(\mathbf{0}, r) \cap L \neq \{\mathbf{0}\}\right).$$

- Le minimum peut être atteint un nombre exponentiel de fois

Minima d'un réseau

Minimum d'un réseau

Tout réseau $L \neq 0$ contient un $\mathbf{b} \neq \mathbf{0}$ de norme euclidienne minimale.
La norme de ce vecteur est le minimum $\lambda_1(L)$:

$$\lambda_1(L) = \min\left(r : \mathcal{B}(\mathbf{0}, r) \cap L \neq \{\mathbf{0}\}\right).$$

- Le minimum peut être atteint un nombre exponentiel de fois

Minima d'un réseau

Minimum d'un réseau

Tout réseau $L \neq 0$ contient un $\mathbf{b} \neq \mathbf{0}$ de norme euclidienne minimale.
La norme de ce vecteur est le minimum $\lambda_1(L)$:

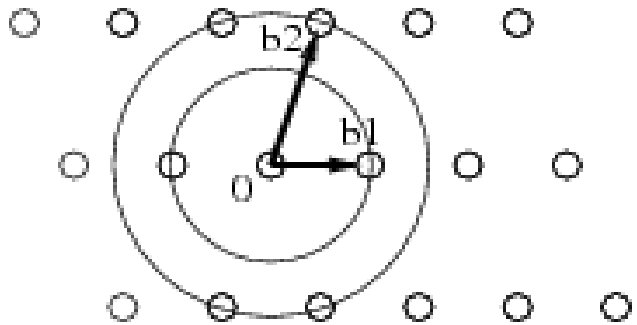
$$\lambda_1(L) = \min\left(r : \mathcal{B}(\mathbf{0}, r) \cap L \neq \{\mathbf{0}\}\right).$$

- Le minimum peut être atteint un nombre exponentiel de fois

Minimas successifs

Pour $i \leq d$, le i-ème minimum d'un réseau L de dimension d est :

$$\lambda_i(L) = \min\left(r : \dim \mathrm{Vec}(\mathcal{B}(\mathbf{0}, r) \cap L) \geq i\right).$$



Déterminant d'un réseau

La matrice de Gram d'une base $(\mathbf{b}_i)_{i \leq d}$ est $G = (\langle \mathbf{b}_i, \mathbf{b}_j \rangle)_{i,j}$.

Déterminant d'un réseau

Soit $\mathbf{b}_1, \ldots, \mathbf{b}_d$ une base d'un réseau L. On définit :

$$\det(L) = \sqrt{\det(G(\mathbf{b}_1, \ldots, \mathbf{b}_d))}.$$

Quelques propriétés simples :

- $\det(L)$ ne dépend pas du choix de la base de L.
- Si L est de plein rang, alors $\det(L) = |\det B|$.
- Hadamard: $\det(L) \leq \prod_i \|\mathbf{b}_i\|$, pour toute base.

Déterminant d'un réseau

La matrice de Gram d'une base $(\mathbf{b}_i)_{i\leq d}$ est $G = (\langle \mathbf{b}_i, \mathbf{b}_j \rangle)_{i,j}$.

Déterminant d'un réseau

Soit $\mathbf{b}_1, \ldots, \mathbf{b}_d$ une base d'un réseau L. On définit :

$$\det(L) = \sqrt{\det(G(\mathbf{b}_1, \ldots, \mathbf{b}_d))}.$$

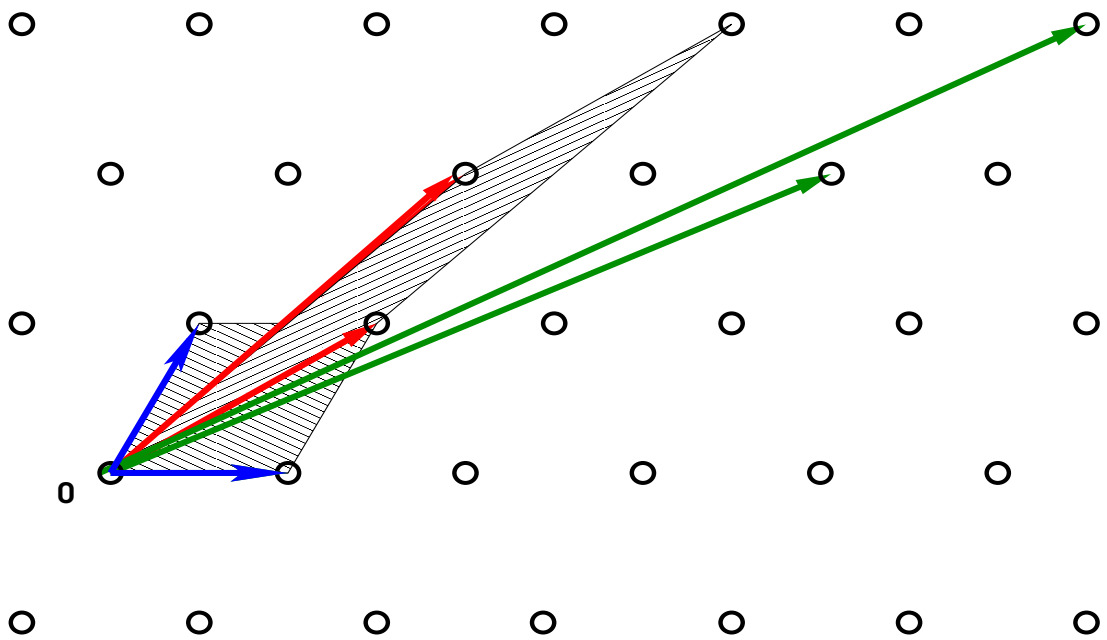
Quelques propriétés simples :

- $\det(L)$ ne dépend pas du choix de la base de L.
- Si L est de plein rang, alors $\det(L) = |\det B|$.
- Hadamard: $\det(L) \leq \prod_i \|\mathbf{b}_i\|$, pour toute base.

Interprétation géometrique du déterminant

Le déterminant d'un réseau admettant $(\mathbf{b}_i)_{i \leq d}$ comme base est le volume du parallélépipède engendré par les $\mathbf{b}_i$.

Il quantifie lui aussi l'aspect clairsemé du réseau suivant d dimensions.



Théorèmes de Minkowski

Fournissent un lien entre les minimas et le déterminant.

Théorème de Minkowski

Soient: $L \subseteq \mathbb{R}^n$ un réseau de plein rang
$S \subseteq \mathbb{R}^n$ convexe et symétrique t.q. $\mathrm{vol}(S) > 2^n \cdot \det(L)$.
Alors il existe $\mathbf{x} \in (L \setminus \mathbf{0}) \cap S$.

Si S est fermé, il suffit que $\mathrm{vol}(S) \geq 2^n \cdot \det(L)$.

Corollaire 1 : $\lambda_1(L) \leq \sqrt{n} \cdot \det(L)^{1/n}$

Corollaire 2 : $\prod_{i \leq n} \lambda_i(L) \leq \sqrt{n}^n \cdot \det(L)$

Mauvaise nouvelle : les preuves ne sont pas effectives

Théorèmes de Minkowski

Fournissent un lien entre les minimas et le déterminant.

Théorème de Minkowski

Soient: $L \subseteq \mathbb{R}^n$ un réseau de plein rang
$S \subseteq \mathbb{R}^n$ convexe et symétrique t.q. $\mathrm{vol}(S) > 2^n \cdot \det(L)$.
Alors il existe $\mathbf{x} \in (L \setminus \mathbf{0}) \cap S$.

Si S est fermé, il suffit que $\mathrm{vol}(S) \geq 2^n \cdot \det(L)$.

Corollaire 1 : $\lambda_1(L) \leq \sqrt{n} \cdot \det(L)^{1/n}$

Corollaire 2 : $\prod_{i \leq n} \lambda_i(L) \leq \sqrt{n}^n \cdot \det(L)$

Mauvaise nouvelle : les preuves ne sont pas effectives

Théorèmes de Minkowski

Fournissent un lien entre les minimas et le déterminant.

Théorème de Minkowski

Soient: $L \subseteq \mathbb{R}^n$ un réseau de plein rang
$S \subseteq \mathbb{R}^n$ convexe et symétrique t.q. $\mathrm{vol}(S) > 2^n \cdot \det(L)$.
Alors il existe $\mathbf{x} \in (L \setminus \mathbf{0}) \cap S$.

Si S est fermé, il suffit que $\mathrm{vol}(S) \geq 2^n \cdot \det(L)$.

Corollaire 1 : $\lambda_1(L) \leq \sqrt{n} \cdot \det(L)^{1/n}$

Corollaire 2 : $\prod_{i \leq n} \lambda_i(L) \leq \sqrt{n}^n \cdot \det(L)$

Mauvaise nouvelle : les preuves ne sont pas effectives

Théorèmes de Minkowski

Fournissent un lien entre les minimas et le déterminant.

Théorème de Minkowski

Soient: $L \subseteq \mathbb{R}^n$ un réseau de plein rang
$S \subseteq \mathbb{R}^n$ convexe et symétrique t.q. $\mathrm{vol}(S) > 2^n \cdot \det(L)$.
Alors il existe $\mathbf{x} \in (L \setminus \mathbf{0}) \cap S$.

Si S est fermé, il suffit que $\mathrm{vol}(S) \geq 2^n \cdot \det(L)$.

Corollaire 1 : $\lambda_1(L) \leq \sqrt{n} \cdot \det(L)^{1/n}$

Corollaire 2 : $\prod_{i \leq n} \lambda_i(L) \leq \sqrt{n}^n \cdot \det(L)$

Mauvaise nouvelle : les preuves ne sont pas effectives

Plan du cours

Problèmes faciles

Étant donnée une base de $L \subseteq \mathbb{Z}^n$, on peut efficacement :

- tester si un vecteur $\mathbf{b}$ appartient à L
- calculer le déterminant de L

Étant données des bases de $L_1 \subseteq \mathbb{Z}^n$ et $L_2 \subseteq \mathbb{Z}^n$, on peut efficacement :

- tester si $L_1 \subseteq L_2$
- tester si $L_1 = L_2$

$\rightsquigarrow$ algèbre linéaire sur des matrices à coefficients entiers.

Problèmes faciles

Étant donnée une base de $L \subseteq \mathbb{Z}^n$, on peut efficacement :

- tester si un vecteur $\mathbf{b}$ appartient à L
- calculer le déterminant de L

Étant données des bases de $L_1 \subseteq \mathbb{Z}^n$ et $L_2 \subseteq \mathbb{Z}^n$, on peut efficacement :

- tester si $L_1 \subseteq L_2$
- tester si $L_1 = L_2$

$\rightsquigarrow$ algèbre linéaire sur des matrices à coefficients entiers.

Le problème du vecteur le plus court

Admet de multiples variantes et généralisations

SVP calculatoire

Étant donnée une base de L, trouver $\mathbf{b} \in L$ avec $\|\mathbf{b}\| = \lambda_1(L)$.

SVP décisionnel

Étant donnée une base de L et un rationnel d, répondre
OUI si $\lambda_1(L) \leq d$, et NON sinon.

- On s'intéresse surtout à ces problèmes pour $d \to +\infty$.
- [Van Emde Boas'81]: DecSVP est NP-difficile pour la norme infinie.
- [Ajtai'98]: DecSVP est NP-difficile sous des réductions probabilistes.

Le problème du vecteur le plus court

Admet de multiples variantes et généralisations

SVP calculatoire

Étant donnée une base de L, trouver $\mathbf{b} \in L$ avec $\|\mathbf{b}\| = \lambda_1(L)$.

SVP décisionnel

Étant donnée une base de L et un rationnel d, répondre
OUI si $\lambda_1(L) \leq d$, et NON sinon.

- On s'intéresse surtout à ces problèmes pour $d \to +\infty$.
- [Van Emde Boas'81]: DecSVP est NP-difficile pour la norme infinie.
- [Ajtai'98]: DecSVP est NP-difficile sous des réductions probabilistes.

Le problème du vecteur le plus court

Admet de multiples variantes et généralisations

SVP calculatoire

Étant donnée une base de L, trouver $\mathbf{b} \in L$ avec $\|\mathbf{b}\| = \lambda_1(L)$.

SVP décisionnel

Étant donnée une base de L et un rationnel d, répondre
OUI si $\lambda_1(L) \leq d$, et NON sinon.

- On s'intéresse surtout à ces problèmes pour $d \to +\infty$.
- [Van Emde Boas'81]: DecSVP est NP-difficile pour la norme infinie.
- [Ajtai'98]: DecSVP est NP-difficile sous des réductions probabilistes.

Relations linéaire entières entre des réels

Formule BBP : $\pi = \sum_{i \geq 0} \frac{1}{16^i} \left(\frac{4}{8i+1} - \frac{2}{8i+4} - \frac{1}{8i+5} - \frac{1}{8i+6} \right)$

Si l'on cherche une relation entière entre $y_1, \ldots, y_d \in \mathbb{R}$, on peut :

Prendre $L := L[(\mathbf{b}_i)_i]$, avec $(\mathbf{b}_i)_i = \begin{pmatrix} y_1 & y_2 & \cdots & y_d \\ 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{pmatrix}$

$\mathbb{Z}$-relation $\sum x_i \cdot y_i = 0 \quad \Rightarrow \quad (0, x_1, \ldots, x_d)^T \in L \setminus 0$

Relations linéaire entières entre des réels

Formule BBP : $\mathbf{1}\cdot\pi = \sum_{i\geq 0} \frac{1}{16^i}\left(\frac{\mathbf{4}}{8i+1} + \frac{\mathbf{-2}}{8i+4} + \frac{\mathbf{-1}}{8i+5} + \frac{\mathbf{-1}}{8i+6}\right)$

Si l'on cherche une relation entière entre $y_1, \ldots, y_d \in \mathbb{R}$, on peut :

Prendre $L := L[(\mathbf{b}_i)_i]$, avec $(\mathbf{b}_i)_i = \begin{pmatrix} y_1 & y_2 & \cdots & y_d \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$

$\mathbb{Z}$-relation $\sum x_i \cdot y_i = 0 \quad \Rightarrow \quad (0, x_1, \ldots, x_d)^T \in L \setminus \mathbf{0}$

Pour un C grand :
$\mathbf{b} \in L_C$ et $\|\mathbf{b}\| = \lambda_1(L_C) \Rightarrow$ plus petite $\mathbb{Z}$-relation non triviale

Relations linéaire entières entre des réels

$$\text{Formule BBP : } \mathbf{1}\cdot\pi = \sum_{i\geq 0} \frac{1}{16^i}\left(\frac{\mathbf{4}}{8i+1} + \frac{\mathbf{-2}}{8i+4} + \frac{\mathbf{-1}}{8i+5} + \frac{\mathbf{-1}}{8i+6}\right)$$

Si l'on cherche une relation entière entre $y_1, \ldots, y_d \in \mathbb{R}$, on peut :

$$\text{Prendre } L := L[(\mathbf{b}_i)_i], \quad \text{avec } (\mathbf{b}_i)_i = \begin{pmatrix} y_1 & y_2 & \cdots & y_d \\ 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{pmatrix}$$

$\mathbb{Z}$-relation $\sum x_i \cdot y_i = 0 \quad \Rightarrow \quad (0, x_1, \ldots, x_d)^T \in L \setminus \mathbf{0}$

Pour un C grand :
$\mathbf{b} \in L_C$ et $\|\mathbf{b}\| = \lambda_1(L_C) \Rightarrow$ plus petite $\mathbb{Z}$-relation non triviale

Relations linéaire entières entre des réels

Formule BBP : $\mathbf{1}\cdot\pi = \sum_{i\geq 0} \frac{1}{16^i}\left(\frac{\mathbf{4}}{8i+1} + \frac{\mathbf{-2}}{8i+4} + \frac{\mathbf{-1}}{8i+5} + \frac{\mathbf{-1}}{8i+6}\right)$

Si l'on cherche une relation entière entre $y_1, \ldots, y_d \in \mathbb{R}$, on peut :

Prendre $L_C := L[(\mathbf{b}_i)_i]$, avec $(\mathbf{b}_i)_i = \begin{pmatrix} Cy_1 & Cy_2 & \ldots & Cy_d \\ 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{pmatrix}$

$\mathbb{Z}$-relation $\sum x_i \cdot y_i = 0 \quad \Rightarrow \quad (0, x_1, \ldots, x_d)^T \in L \setminus \mathbf{0}$

Pour un C grand :
$\mathbf{b} \in L_C$ et $\|\mathbf{b}\| = \lambda_1(L_C) \Rightarrow$ plus petite $\mathbb{Z}$-relation non triviale

Approximations de SVP

SVP_γ pour un facteur d'approximation $\gamma \geq 1$

Étant donnée une base de L, trouver $\mathbf{b} \in L$ t.q. $0 < \|\mathbf{b}\| \leq \gamma \cdot \lambda_1(L)$.

GapSVP_γ pour un facteur d'approximation $\gamma \geq 1$

Étant donnée une base de L et $d \in \mathbb{Q}$, répondre OUI si $\lambda_1(L) \leq d$ et NON si $\lambda_1(L) \geq \gamma \cdot d$.

- [HavReg'07]: GapSVP_γ est NP-difficile pour tout $\gamma \leq 2^{(\log d)^{1-\varepsilon}}$, sous des réductions probabilistes.
- [AhaReg'04]: GapSVP_γ est dans $\mathrm{NP} \cap \mathrm{coNP}$ pour $\gamma \geq \sqrt{d}$.
  $\Rightarrow$ Selon toute vraisemblance, GapSVP_γ n'est pas NP-difficile pour un tel γ.
- SVP_γ est dans P pour $\gamma \geq 2^{O(d \log \log d / \log d)}$.

Approximations de SVP

SVP_γ pour un facteur d'approximation $\gamma \geq 1$

Étant donnée une base de L, trouver $\mathbf{b} \in L$ t.q. $0 < \|\mathbf{b}\| \leq \gamma \cdot \lambda_1(L)$.

GapSVP_γ pour un facteur d'approximation $\gamma \geq 1$

Étant donnée une base de L et $d \in \mathbb{Q}$, répondre OUI si $\lambda_1(L) \leq d$ et NON si $\lambda_1(L) \geq \gamma \cdot d$.

- [HavReg'07]: GapSVP_γ est NP-difficile pour tout $\gamma \leq 2^{(\log d)^{1-\varepsilon}}$, sous des réductions probabilistes.
- [AhaReg'04]: GapSVP_γ est dans NP $\cap$ coNP pour $\gamma \geq \sqrt{d}$.
  $\Rightarrow$ Selon toute vraisemblance, GapSVP_γ n'est pas NP-difficile pour un tel γ.
- SVP_γ est dans P pour $\gamma \geq 2^{\Omega(\frac{d \log\log d}{\log d})}$.

Approximations de SVP

SVP_γ pour un facteur d'approximation $\gamma \geq 1$

Étant donnée une base de L, trouver $\mathbf{b} \in L$ t.q. $0 < \|\mathbf{b}\| \leq \gamma \cdot \lambda_1(L)$.

GapSVP_γ pour un facteur d'approximation $\gamma \geq 1$

Étant donnée une base de L et $d \in \mathbb{Q}$, répondre OUI si $\lambda_1(L) \leq d$ et NON si $\lambda_1(L) \geq \gamma \cdot d$.

- [HavReg'07]: GapSVP_γ est NP-difficile pour tout $\gamma \leq 2^{(\log d)^{1-\varepsilon}}$, sous des réductions probabilistes.
- [AhaReg'04]: GapSVP_γ est dans NP $\cap$ coNP pour $\gamma \geq \sqrt{d}$.
  $\Rightarrow$ Selon toute vraisemblance, GapSVP_γ n'est pas NP-difficile pour un tel γ.
- SVP_γ est dans P pour $\gamma \geq 2^{\Omega(\frac{d \log\log d}{\log d})}$.

Approximations de SVP

SVP_γ pour un facteur d'approximation $\gamma \geq 1$

Étant donnée une base de L, trouver $\mathbf{b} \in L$ t.q. $0 < \|\mathbf{b}\| \leq \gamma \cdot \lambda_1(L)$.

GapSVP_γ pour un facteur d'approximation $\gamma \geq 1$

Étant donnée une base de L et $d \in \mathbb{Q}$, répondre OUI si $\lambda_1(L) \leq d$ et NON si $\lambda_1(L) \geq \gamma \cdot d$.

- [HavReg'07]: GapSVP_γ est NP-difficile pour tout $\gamma \leq 2^{(\log d)^{1-\varepsilon}}$, sous des réductions probabilistes.
- [AhaReg'04]: GapSVP_γ est dans $\mathrm{NP} \cap \mathrm{coNP}$ pour $\gamma \geq \sqrt{d}$.
  $\Rightarrow$ Selon toute vraisemblance, GapSVP_γ n'est pas NP-difficile pour un tel γ.
- SVP_γ est dans P pour $\gamma \geq 2^{\Omega(\frac{d \log\log d}{\log d})}$.

Factoriser les polynômes à coefficients entiers

Application à l'origine de l'algorithmique des réseaux euclidiens

Factoring polynomials with rational coefficients,
Lenstra, Lenstra and Lovász. Math. Ann., 1982

$\Rightarrow$ Premier algorithme polynomial de factorisation dans $\mathbb{Z}[x]$

Pour $P \in \mathbb{Z}[x]$:

0- Si $\deg P \leq 1$, s'arrêter

1- Calculer une racine $\alpha \in \mathbb{C}$ de P

2- Déterminer le polynôme minimal P_α de α, en cherchant des $\mathbb{Z}$-combinaisons entre $1, \alpha, \ldots, \alpha^i \in \mathbb{C}$, pour valeur de i croissante

3- Diviser P par P_α et recommencer

Factoriser les polynômes à coefficients entiers

Application à l'origine de l'algorithmique des réseaux euclidiens

Factoring polynomials with rational coefficients,
Lenstra, Lenstra and Lovász. Math. Ann., 1982

$\Rightarrow$ Premier algorithme polynomial de factorisation dans $\mathbb{Z}[x]$

Pour $P \in \mathbb{Z}[x]$:

0- Si $\deg P \leq 1$, s'arrêter

1- Calculer une racine $\alpha \in \mathbb{C}$ de P

2- Déterminer le polynôme minimal P_α de α, en cherchant des $\mathbb{Z}$-combinaisons entre $1, \alpha, \ldots, \alpha^i \in \mathbb{C}$, pour valeur de i croissante

3- Diviser P par P_α et recommencer

Zoom sur l'étape 2

2- Déterminer le polynôme minimal P_α de α, en cherchant des $\mathbb{Z}$-combinaisons entre $1, \alpha, \ldots, \alpha^i \in \mathbb{C}$, pour caleur de i croissante

Difficultés :

- $\mathbb{Z}$-relations entre des complexes et non des réels. Fausse difficulté.
- On a un algo polynomial qui résout seulement SVP_γ, pour $\gamma \gg 1$.

Solution :

- On peut montrer (bornes de Landau-Mignotte) que $\lambda_2(L_C) \gg \gamma \cdot \lambda_1(L_C)$ pour C suffisamment grand et explicite.

$\Rightarrow$ Tous les vecteurs courts de L_C donnent des multiples d'une $\mathbb{Z}$-relation minimale.

Zoom sur l'étape 2

2- Déterminer le polynôme minimal P_α de α, en cherchant des $\mathbb{Z}$-combinaisons entre $1, \alpha, \ldots, \alpha^i \in \mathbb{C}$, pour caleur de i croissante

Difficultés :

- $\mathbb{Z}$-relations entre des complexes et non des réels.
  Fausse difficulté.
- On a un algo polynomial qui résout seulement SVP_γ, pour $\gamma \gg 1$.

Solution :

- On peut montrer (bornes de Landau-Mignotte) que $\lambda_2(L_C) \gg \gamma \cdot \lambda_1(L_C)$ pour C suffisamment grand et explicite.

$\Rightarrow$ Tous les vecteurs courts de L_C donnent des multiples d'une $\mathbb{Z}$-relation minimale.

Zoom sur l'étape 2

2- Déterminer le polynôme minimal P_α de α, en cherchant des $\mathbb{Z}$-combinaisons entre $1, \alpha, \ldots, \alpha^i \in \mathbb{C}$, pour caleur de i croissante

Difficultés :

- $\mathbb{Z}$-relations entre des complexes et non des réels.
  Fausse difficulté.
- On a un algo polynomial qui résout seulement SVP_γ, pour $\gamma \gg 1$.

Solution :

- On peut montrer (bornes de Landau-Mignotte) que $\lambda_2(L_C) \gg \gamma \cdot \lambda_1(L_C)$ pour C suffisamment grand et explicite.

$\Rightarrow$ Tous les vecteurs courts de L_C donnent des multiples d'une $\mathbb{Z}$-relation minimale.

Variantes de SVP

uSVP_γ (Unique-SVP)

Étant donnée une base de L t.q. $\lambda_2(L) \geq \gamma \cdot \lambda_1(L)$,
trouver $\mathbf{b} \in L$ t.q. $\|\mathbf{b}\| = \lambda_1(L)$.

HSVP_γ (Hermite SVP)

Étant donnée une base de L, trouver $\mathbf{b} \in L$ t.q. $\|\mathbf{b}\| \leq \gamma \cdot (\det L)^{1/d}$.

HSVP_γ est essentiellement équivalent à SVP_γ (réductions polynomiales).

- $\mathrm{HSVP}_{\gamma'} \leq \mathrm{SVP}_\gamma$, pour $\gamma' = \sqrt{d} \cdot \gamma$.
- $\mathrm{SVP}_{\gamma'} \leq \mathrm{HSVP}_\gamma$, pour $\gamma' = \gamma^2$.

uSVP_γ est essentiellement équivalent à GapSVP_γ.

- $\mathrm{uSVP}_\gamma \leq \mathrm{GapSVP}_\gamma$.
- $\mathrm{GapSVP}_{\gamma'} \leq \mathrm{uSVP}_\gamma$, pour $\gamma' = \sqrt{n}\gamma$.

Variantes de SVP

uSVP_γ (Unique-SVP)

Étant donnée une base de L t.q. $\lambda_2(L) \geq \gamma \cdot \lambda_1(L)$,
trouver $\mathbf{b} \in L$ t.q. $\|\mathbf{b}\| = \lambda_1(L)$.

HSVP_γ (Hermite SVP)

Étant donnée une base de L, trouver $\mathbf{b} \in L$ t.q. $\|\mathbf{b}\| \leq \gamma \cdot (\det L)^{1/d}$.

HSVP_γ est essentiellement équivalent à SVP_γ (réductions polynomiales).

- $\mathrm{HSVP}_{\gamma'} \leq \mathrm{SVP}_\gamma$, pour $\gamma' = \sqrt{d} \cdot \gamma$.
- $\mathrm{SVP}_{\gamma'} \leq \mathrm{HSVP}_\gamma$, pour $\gamma' = \gamma^2$.

uSVP_γ est essentiellement équivalent à GapSVP_γ.

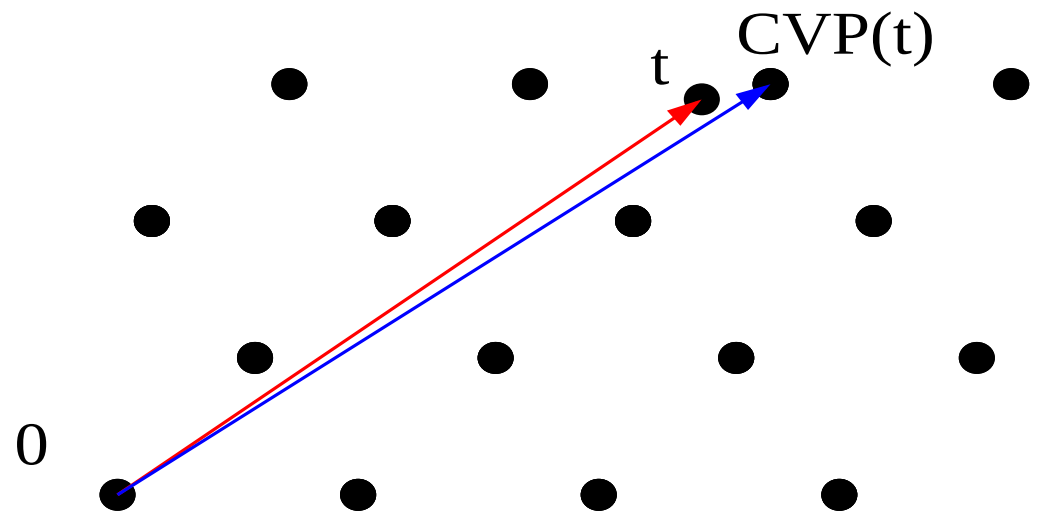
- $\mathrm{uSVP}_\gamma \leq \mathrm{GapSVP}_\gamma$.
- $\mathrm{GapSVP}_{\gamma'} \leq \mathrm{uSVP}_\gamma$, pour $\gamma' = \sqrt{n}\gamma$.

Le problème du décodage à distance bornée

BDD_γ pour $\gamma \geq 1$

Étant donnée une base de L et un vecteur $\mathbf{t}$ à distance $\leq \lambda_1(L)/\gamma$, trouver $\mathbf{b} \in L$ t.q.

$$\|\mathbf{b} - \mathbf{t}\| \leq \lambda_1(L)/\gamma.$$



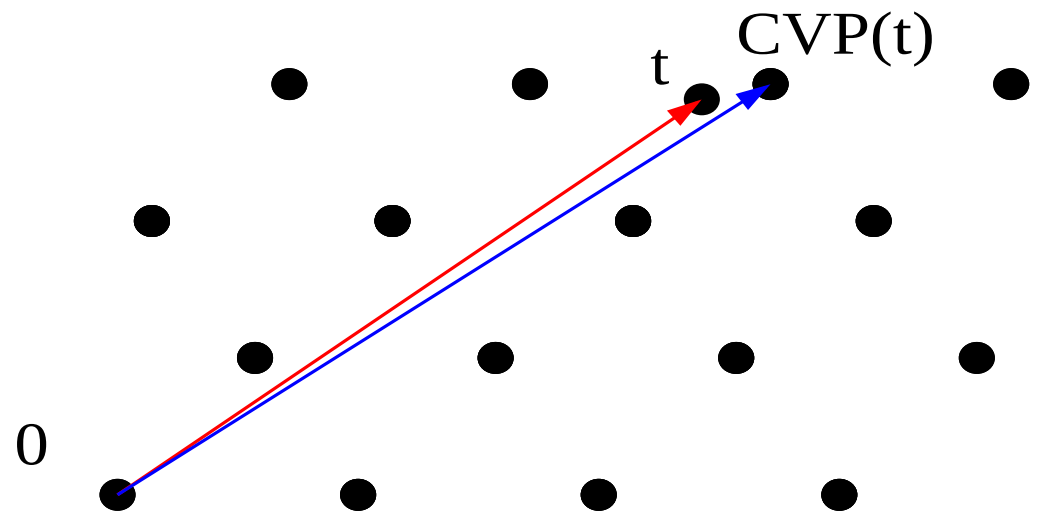
BDD_γ est essentiellement à uSVP_γ
(réductions avec pertes de petits facteur constants).

Le problème du décodage à distance bornée

BDD_γ pour $\gamma \geq 1$

Étant donnée une base de L et un vecteur $\mathbf{t}$ à distance $\leq \lambda_1(L)/\gamma$, trouver $\mathbf{b} \in L$ t.q.

$$\|\mathbf{b} - \mathbf{t}\| \leq \lambda_1(L)/\gamma.$$



BDD_γ est essentiellement à uSVP_γ
(réductions avec pertes de petits facteur constants).

Application de BDD : communications MIMO

Système de communications MIMO (Multi-Input Multi-Output) brut

- N_t antennes émettrices, N_r antennes réceptrices
- Le message est codé dans un vecteur $\mathbf{x} \in \mathbb{Z}^{N_t}$
- Le récepteur observe $\mathbf{y} = H \cdot \mathbf{x} + \mathbf{e} \in \mathbb{R}^{N_r}$, avec :
  - H la matrice du canal (codant les interférences entre les antennes)
  - $\mathbf{e}$ un petit vecteur (correspondant à du bruit physique)

Souvent, H est connue, et il s'agit de retrouver $\mathbf{x}$ à partir de $\mathbf{y}$.
$\Rightarrow$ C'est une instance de BDD !

Très utilisé en pratique :

- Normes Wifi et Wimax
- Freebox v5 (entre les 2 boîtiers, et pour le wifi)
- Téléphones portables : 3G dual carrier (3G+)

Application de BDD : communications MIMO

Système de communications MIMO (Multi-Input Multi-Output) brut

- N_t antennes émettrices, N_r antennes réceptrices
- Le message est codé dans un vecteur $\mathbf{x} \in \mathbb{Z}^{N_t}$
- Le récepteur observe $\mathbf{y} = H \cdot \mathbf{x} + \mathbf{e} \in \mathbb{R}^{N_r}$, avec :
  - H la matrice du canal (codant les interférences entre les antennes)
  - $\mathbf{e}$ un petit vecteur (correspondant à du bruit physique)

Souvent, H est connue, et il s'agit de retrouver $\mathbf{x}$ à partir de $\mathbf{y}$.
$\Rightarrow$ C'est une instance de BDD !

Très utilisé en pratique :

- Normes Wifi et Wimax
- Freebox v5 (entre les 2 boîtiers, et pour le wifi)
- Téléphones portables : 3G dual carrier (3G+)

Application de BDD : communications MIMO

Système de communications MIMO (Multi-Input Multi-Output) brut

- N_t antennes émettrices, N_r antennes réceptrices
- Le message est codé dans un vecteur $\mathbf{x} \in \mathbb{Z}^{N_t}$
- Le récepteur observe $\mathbf{y} = H \cdot \mathbf{x} + \mathbf{e} \in \mathbb{R}^{N_r}$, avec :
  - H la matrice du canal (codant les interférences entre les antennes)
  - $\mathbf{e}$ un petit vecteur (correspondant à du bruit physique)

Souvent, H est connue, et il s'agit de retrouver $\mathbf{x}$ à partir de $\mathbf{y}$.
$\Rightarrow$ C'est une instance de BDD !

Très utilisé en pratique :

- Normes Wifi et Wimax
- Freebox v5 (entre les 2 boîtiers, et pour le wifi)
- Téléphones portables : 3G dual carrier (3G+)

Plan du cours

1. Définitions et propriétés élémentaires
2. Problèmes calculatoires
3. **Résoudre SVP**
4. Concept de réduction de réseau
5. L'algorithme LLL

À partir de maintenant, les réseaux sont entiers (i.e., $L \subseteq \mathbb{Z}^n$) et de rangs pleins (i.e., $d = n$)

Résolution de SVP : état de l'art

	Temps	Espace
Par le calcul de la cellule de Voronoi [MicVou10b]	$2^{2n+o(n)}$	$2^{n+o(n)}$
Avec le principe de saturation [AjtKumSiv01,MicVou10a,PujSte10]	$2^{2.465n+o(n)}$	$2^{1.325n+o(n)}$
Algorithme d'énumération [Kannan83,FinPoh83,HanSte07]	$n^{n/(2e)+o(n)}$	$\mathcal{P}oly(n)$

- Il s'agit de bornes
- La complexité est polynomiale en $\log \|B\| := \log \max \|\mathbf{b}_i\|$, où les $\mathbf{b}_i$ sont les vecteurs donnés en entrée

Résolution de SVP : état de l'art

	Temps	Espace
Par le calcul de la cellule de Voronoi [MicVou10b]	$2^{2n+o(n)}$	$2^{n+o(n)}$
Avec le principe de saturation [AjtKumSiv01,MicVou10a,PujSte10]	$2^{2.465n+o(n)}$	$2^{1.325n+o(n)}$
Algorithme d'énumération [Kannan83,FinPoh83,HanSte07]	$n^{n/(2e)+o(n)}$	$\mathcal{P}oly(n)$

- Il s'agit de bornes
- La complexité est polynomiale en $\log \|B\| := \log \max \|\mathbf{b}_i\|$, où les $\mathbf{b}_i$ sont les vecteurs donnés en entrée

Le principe de saturation

Kabatyansky & Levenshtein

Soit $E \subseteq \mathbb{R}^n \setminus \mathbf{0}$. Si l'angle entre tous $\mathbf{u} \neq \mathbf{v}$ dans E est $\geq \phi_0$, alors $|E| \leq 2^{cn+o(n)}$ pour une certaine constante $c = c(\phi_0)$.

- Pour $\phi_0 = 60°$, on obtient $|E| \leq 2^{0.401 \cdot n}$.

Autre formulation :

Si des points sur la sphère unité sont à distances bornées inférieurement, alors ces points sont au plus $2^{O(n)}$.

Conséquence :

Le minimum d'un réseau de dimension n est atteint $\leq 2^{0.401 \cdot n+o(n)}$ fois.

Le principe de saturation

Kabatyansky & Levenshtein

Soit $E \subseteq \mathbb{R}^n \setminus \mathbf{0}$. Si l'angle entre tous $\mathbf{u} \neq \mathbf{v}$ dans E est $\geq \phi_0$, alors $|E| \leq 2^{cn+o(n)}$ pour une certaine constante $c = c(\phi_0)$.

- Pour $\phi_0 = 60°$, on obtient $|E| \leq 2^{0.401 \cdot n}$.

Autre formulation :

Si des points sur la sphère unité sont à distances bornées inférieurement, alors ces points sont au plus $2^{O(n)}$.

Conséquence :

Le minimum d'un réseau de dimension n est atteint $\leq 2^{0.401 \cdot n+o(n)}$ fois.

Le principe de saturation

Kabatyansky & Levenshtein

Soit $E \subseteq \mathbb{R}^n \setminus \mathbf{0}$. Si l'angle entre tous $\mathbf{u} \neq \mathbf{v}$ dans E est $\geq \phi_0$, alors $|E| \leq 2^{cn+o(n)}$ pour une certaine constante $c = c(\phi_0)$.

- Pour $\phi_0 = 60°$, on obtient $|E| \leq 2^{0.401 \cdot n}$.

Autre formulation :

Si des points sur la sphère unité sont à distances bornées inférieurement, alors ces points sont au plus $2^{O(n)}$.

Conséquence :

Le minimum d'un réseau de dimension n est atteint $\leq 2^{0.401 \cdot n+o(n)}$ fois.

Un premier algorithme

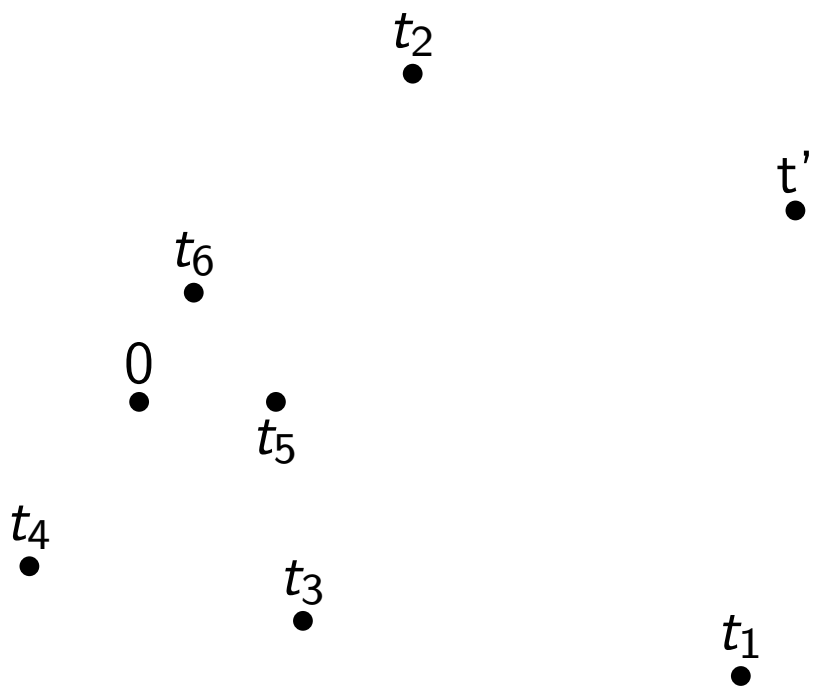
L'algorithme dépend de 2 paramètres, $N \in \mathbb{Z}$ et $\delta \in]0,1[$.

1. Échantillonner $\mathbf{t}_1, \ldots, \mathbf{t}_N \in L$.
2. Pour $i = 1..N$:
   Tant qu'il existe $j < i$ t.q. $\|\mathbf{t}_i - \mathbf{t}_j\| \leq \delta \cdot \|\mathbf{t}_i\|$, remplacer $\mathbf{t}_i$ par $\mathbf{t}_i - \mathbf{t}_j$.
3. Renvoyer un plus court des vecteurs obtenus.

Un premier algorithme

L'algorithme dépend de 2 paramètres, $N \in \mathbb{Z}$ et $\delta \in]0,1[$.

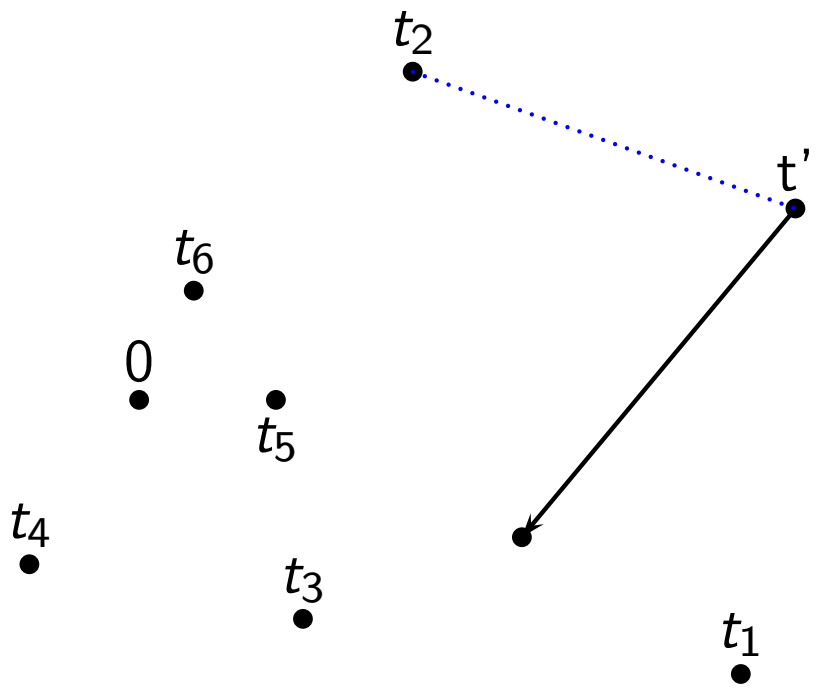
1. Échantillonner $\mathbf{t}_1, \ldots, \mathbf{t}_N \in L$.
2. Pour $i = 1..N$:
   Tant qu'il existe $j < i$ t.q. $\|\mathbf{t}_i - \mathbf{t}_j\| \leq \delta \cdot \|\mathbf{t}_i\|$, remplacer $\mathbf{t}_i$ par $\mathbf{t}_i - \mathbf{t}_j$.
3. Renvoyer un plus court des vecteurs obtenus.



Un premier algorithme

L'algorithme dépend de 2 paramètres, $N \in \mathbb{Z}$ et $\delta \in]0,1[$.

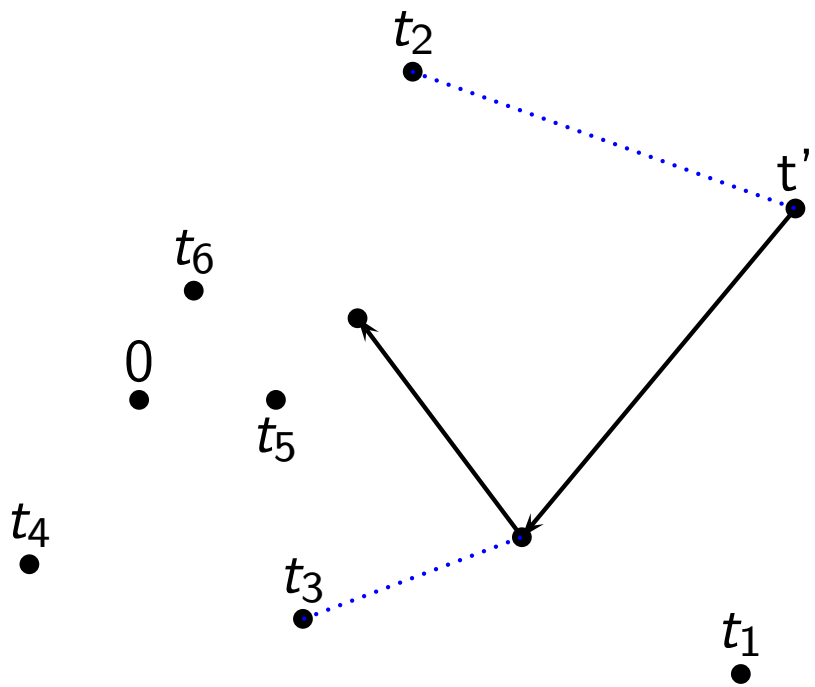
1. Échantillonner $\mathbf{t}_1, \ldots, \mathbf{t}_N \in L$.
2. Pour $i = 1..N$:
   Tant qu'il existe $j < i$ t.q. $\|\mathbf{t}_i - \mathbf{t}_j\| \leq \delta \cdot \|\mathbf{t}_i\|$, remplacer $\mathbf{t}_i$ par $\mathbf{t}_i - \mathbf{t}_j$.
3. Renvoyer un plus court des vecteurs obtenus.



Un premier algorithme

L'algorithme dépend de 2 paramètres, $N \in \mathbb{Z}$ et $\delta \in]0,1[$.

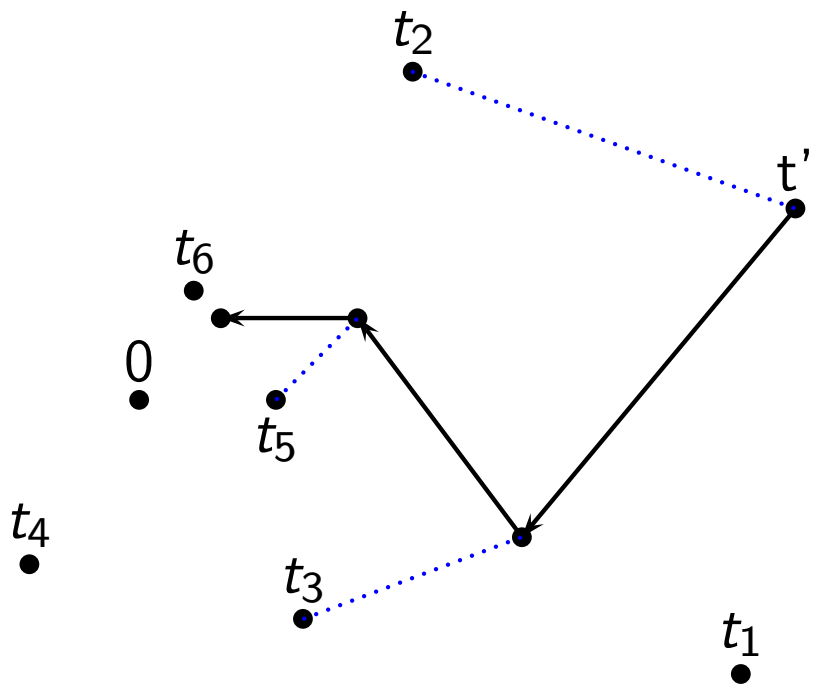
1. Échantillonner $\mathbf{t}_1, \ldots, \mathbf{t}_N \in L$.
2. Pour $i = 1..N$:
   Tant qu'il existe $j < i$ t.q. $\|\mathbf{t}_i - \mathbf{t}_j\| \leq \delta \cdot \|\mathbf{t}_i\|$, remplacer $\mathbf{t}_i$ par $\mathbf{t}_i - \mathbf{t}_j$.
3. Renvoyer un plus court des vecteurs obtenus.



Un premier algorithme

L'algorithme dépend de 2 paramètres, $N \in \mathbb{Z}$ et $\delta \in]0,1[$.

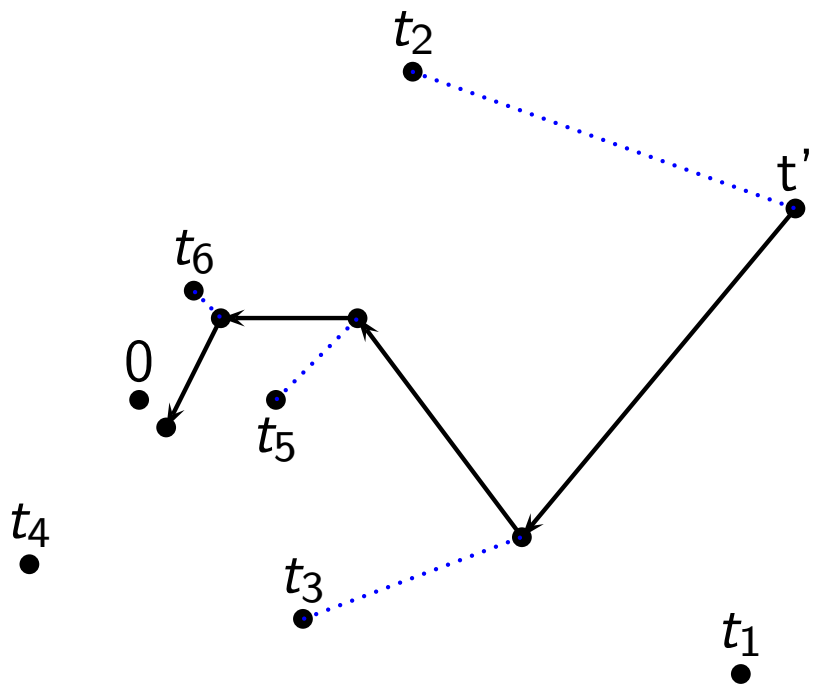
1. Échantillonner $\mathbf{t}_1, \ldots, \mathbf{t}_N \in L$.
2. Pour $i = 1..N$:
   Tant qu'il existe $j < i$ t.q. $\|\mathbf{t}_i - \mathbf{t}_j\| \leq \delta \cdot \|\mathbf{t}_i\|$, remplacer $\mathbf{t}_i$ par $\mathbf{t}_i - \mathbf{t}_j$.
3. Renvoyer un plus court des vecteurs obtenus.



Un premier algorithme

L'algorithme dépend de 2 paramètres, $N \in \mathbb{Z}$ et $\delta \in]0,1[$.

1. Échantillonner $\mathbf{t}_1, \ldots, \mathbf{t}_N \in L$.
2. Pour $i = 1..N$:
   Tant qu'il existe $j < i$ t.q. $\|\mathbf{t}_i - \mathbf{t}_j\| \leq \delta \cdot \|\mathbf{t}_i\|$, remplacer $\mathbf{t}_i$ par $\mathbf{t}_i - \mathbf{t}_j$.
3. Renvoyer un plus court des vecteurs obtenus.



Analyse de complexité

L'algorithme dépend de 2 paramètres, $N \in \mathbb{Z}$ et $\delta \in]0,1[$.

1. Échantillonner $\mathbf{t}_1, \ldots, \mathbf{t}_N \in L$.
2. Pour $i = 1..N$:
   Tant qu'il existe $j < i$ t.q. $\|\mathbf{t}_i - \mathbf{t}_j\| \leq \delta \cdot \|\mathbf{t}_i\|$, remplacer $\mathbf{t}_i$ par $\mathbf{t}_i - \mathbf{t}_j$.
3. Renvoyer un plus court des vecteurs obtenus.

Espace $\approx N$. Et le temps ?

- Soit B une borne sur la norme des vecteurs échantillonnés.
- Saturation $\Rightarrow$ au plus $2^{0.401 \cdot n}$ points dans chaque couronne $\mathcal{B}(R) \setminus \mathcal{B}(R(1-\varepsilon))$.
- Au plus $\log_{1/(1-\varepsilon)}(B/\lambda_1(L))$ couronnes.
  Cette quantité peut être rendue $\leq \mathcal{P}oly(n)$.

$\Rightarrow$ Temps $\lesssim 2^{0.401 \cdot n} \cdot N$.

Analyse de complexité

L'algorithme dépend de 2 paramètres, $N \in \mathbb{Z}$ et $\delta \in]0,1[$.

1. Échantillonner $\mathbf{t}_1, \ldots, \mathbf{t}_N \in L$.
2. Pour $i = 1..N$:
   Tant qu'il existe $j < i$ t.q. $\|\mathbf{t}_i - \mathbf{t}_j\| \leq \delta \cdot \|\mathbf{t}_i\|$, remplacer $\mathbf{t}_i$ par $\mathbf{t}_i - \mathbf{t}_j$.
3. Renvoyer un plus court des vecteurs obtenus.

Espace $\approx N$. Et le temps ?

- Soit B une borne sur la norme des vecteurs échantillonnés.
- Saturation $\Rightarrow$ au plus $2^{0.401 \cdot n}$ points dans chaque couronne $\mathcal{B}(R) \setminus \mathcal{B}(R(1-\varepsilon))$.
- Au plus $\log_{1/(1-\varepsilon)}(B/\lambda_1(L))$ couronnes.
  Cette quantité peut être rendue $\leq \mathcal{P}oly(n)$.

$\Rightarrow$ Temps $\lesssim 2^{0.401 \cdot n} \cdot N$.

Correction : Comment s'assurer que l'on résout SVP ?

Intuition : Si l'algo ne résout pas SVP, c'est qu'il "voit" le réseau.

Principe : Cacher le réseau à l'algo en ajoutant du bruit : $\mathbf{t}_i \to \mathbf{t}_i + \mathbf{e}_i$.

Modifications :

- On tire $\mathbf{e}_i$ dans une boule de rayon $\approx \lambda_1(L)$.
- On définit $\mathbf{t}'_i = \mathbf{e}_i \bmod L \quad (\mathbf{t}_i = \mathbf{t}'_i - \mathbf{e}_i)$.
- On crible $\mathbf{t}'_i$ à l'aide des $\mathbf{t}_j$ précédents.
- Après que $\mathbf{t}'_i$ a passé le crible : $\mathbf{t}_i := \mathbf{t}'_i - \mathbf{e}_i \in L$.

Ça fonctionne : Soit $\mathbf{s} \in L$ t.q. $\|\mathbf{s}\| = \lambda_1(L)$.

- Dans certains cas : $\mathbf{e}_i = (\mathbf{e}_i - \mathbf{s}) + \mathbf{s}$, avec $\mathbf{e}_i - \mathbf{s} \in \mathcal{B}(\lambda_1(L))$.
- Alors $\mathbf{t}'_i$ est le même, mais pas le $\mathbf{t}_i$ final.

Correction : Comment s'assurer que l'on résout SVP ?

Intuition : Si l'algo ne résout pas SVP, c'est qu'il "voit" le réseau.

Principe : Cacher le réseau à l'algo en ajoutant du bruit : $\mathbf{t}_i \rightarrow \mathbf{t}_i + \mathbf{e}_i$.

Modifications :

- On tire $\mathbf{e}_i$ dans une boule de rayon $\approx \lambda_1(L)$.
- On définit $\mathbf{t}'_i = \mathbf{e}_i \bmod L \quad (\mathbf{t}_i = \mathbf{t}'_i - \mathbf{e}_i)$.
- On crible $\mathbf{t}'_i$ à l'aide des $\mathbf{t}_j$ précédents.
- Après que $\mathbf{t}'_i$ a passé le crible : $\mathbf{t}_i := \mathbf{t}'_i - \mathbf{e}_i \in L$.

Ça fonctionne : Soit $\mathbf{s} \in L$ t.q. $\|\mathbf{s}\| = \lambda_1(L)$.

- Dans certains cas : $\mathbf{e}_i = (\mathbf{e}_i - \mathbf{s}) + \mathbf{s}$, avec $\mathbf{e}_i - \mathbf{s} \in \mathcal{B}(\lambda_1(L))$.
- Alors $\mathbf{t}'_i$ est le même, mais pas le $\mathbf{t}_i$ final.
- Si N était assez grand pour que l'on ait le même $\mathbf{t}'^{\text{mid}} \in L$ pour une vie de labo...
- ... on trouve $\mathbf{t}^{\text{mid}}$ et $\mathbf{t}^{\text{mid}} + \mathbf{s}$ avec grande probabilité.

Correction : Comment s'assurer que l'on résout SVP ?

Intuition : Si l'algo ne résout pas SVP, c'est qu'il "voit" le réseau.

Principe : Cacher le réseau à l'algo en ajoutant du bruit : $\mathbf{t}_i \rightarrow \mathbf{t}_i + \mathbf{e}_i$.

Modifications :

- On tire $\mathbf{e}_i$ dans une boule de rayon $\approx \lambda_1(L)$.
- On définit $\mathbf{t}'_i = \mathbf{e}_i \bmod L \quad (\mathbf{t}_i = \mathbf{t}'_i - \mathbf{e}_i)$.
- On crible $\mathbf{t}'_i$ à l'aide des $\mathbf{t}_j$ précédents.
- Après que $\mathbf{t}'_i$ a passé le crible : $\mathbf{t}_i := \mathbf{t}'_i - \mathbf{e}_i \in L$.

Ça fonctionne : Soit $\mathbf{s} \in L$ t.q. $\|\mathbf{s}\| = \lambda_1(L)$.

- Dans certains cas : $\mathbf{e}_i = (\mathbf{e}_i - \mathbf{s}) + \mathbf{s}$, avec $\mathbf{e}_i - \mathbf{s} \in \mathcal{B}(\lambda_1(L))$.
- Alors $\mathbf{t}'_i$ est le même, mais pas le $\mathbf{t}_i$ final.
- N fixé assez grand pour que l'on ait le même $\mathbf{t}^{\text{end}} \in L$ au moins 2 fois.

$\Rightarrow$ on trouve $\mathbf{t}^{\text{end}}$ et $\mathbf{t}^{\text{end}} + \mathbf{s}$, avec grande probabilité.

Correction : Comment s'assurer que l'on résout SVP ?

Intuition : Si l'algo ne résout pas SVP, c'est qu'il "voit" le réseau.

Principe : Cacher le réseau à l'algo en ajoutant du bruit : $\mathbf{t}_i \to \mathbf{t}_i + \mathbf{e}_i$.

Modifications :

- On tire $\mathbf{e}_i$ dans une boule de rayon $\approx \lambda_1(L)$.
- On définit $\mathbf{t}'_i = \mathbf{e}_i \bmod L \quad (\mathbf{t}_i = \mathbf{t}'_i - \mathbf{e}_i)$.
- On crible $\mathbf{t}'_i$ à l'aide des $\mathbf{t}_j$ précédents.
- Après que $\mathbf{t}'_i$ a passé le crible : $\mathbf{t}_i := \mathbf{t}'_i - \mathbf{e}_i \in L$.

Ça fonctionne : Soit $\mathbf{s} \in L$ t.q. $\|\mathbf{s}\| = \lambda_1(L)$.

- Dans certains cas : $\mathbf{e}_i = (\mathbf{e}_i - \mathbf{s}) + \mathbf{s}$, avec $\mathbf{e}_i - \mathbf{s} \in \mathcal{B}(\lambda_1(L))$.
- Alors $\mathbf{t}'_i$ est le même, mais pas le $\mathbf{t}_i$ final.
- N fixé assez grand pour que l'on ait le même $\mathbf{t}^{end} \in L$ au moins 2 fois.

$\Rightarrow$ on trouve $\mathbf{t}^{end}$ et $\mathbf{t}^{end} + \mathbf{s}$, avec grande probabilité.

Correction : Comment s'assurer que l'on résout SVP ?

Intuition : Si l'algo ne résout pas SVP, c'est qu'il "voit" le réseau.

Principe : Cacher le réseau à l'algo en ajoutant du bruit : $\mathbf{t}_i \to \mathbf{t}_i + \mathbf{e}_i$.

Modifications :

- On tire $\mathbf{e}_i$ dans une boule de rayon $\approx \lambda_1(L)$.
- On définit $\mathbf{t}'_i = \mathbf{e}_i \bmod L \quad (\mathbf{t}_i = \mathbf{t}'_i - \mathbf{e}_i)$.
- On crible $\mathbf{t}'_i$ à l'aide des $\mathbf{t}_j$ précédents.
- Après que $\mathbf{t}'_i$ a passé le crible : $\mathbf{t}_i := \mathbf{t}'_i - \mathbf{e}_i \in L$.

Ça fonctionne : Soit $\mathbf{s} \in L$ t.q. $\|\mathbf{s}\| = \lambda_1(L)$.

- Dans certains cas : $\mathbf{e}_i = (\mathbf{e}_i - \mathbf{s}) + \mathbf{s}$, avec $\mathbf{e}_i - \mathbf{s} \in \mathcal{B}(\lambda_1(L))$.
- Alors $\mathbf{t}'_i$ est le même, mais pas le $\mathbf{t}_i$ final.
- N fixé assez grand pour que l'on ait le même $\mathbf{t}^{end} \in L$ au moins 2 fois.

$\Rightarrow$ on trouve $\mathbf{t}^{end}$ et $\mathbf{t}^{end} + \mathbf{s}$, avec grande probabilité.

Plan du cours

Motivation

- On souhaite résoudre SVP_γ ou HSVP_γ, efficacement.
- Mais pour un petit γ, c'est (essentiellement) NP-difficile.
- Et les meilleurs algos connus sont exponentiels.

Principe :

- effectuer des opérations simples sur une base donnée
- en maintenant la propriété d'être une base
- en progressant petit à petit

Motivation

- On souhaite résoudre SVP_γ ou HSVP_γ, efficacement.
- Mais pour un petit γ, c'est (essentiellement) NP-difficile.
- Et les meilleurs algos connus sont exponentiels.

Principe :

- effectuer des opérations simples sur une base donnée
- en maintenant la propriété d'être une base
- en progressant petit à petit

Une approche infructueuse : tenter de minimiser les normes

Idéalement, on pourrait souhaiter une base atteignant les minimas.

- Les minimas sont atteints par des vecteurs lin. ind. du réseau.
- Mais il existe des réseaux dont aucune base n'atteint les minimas.

$$\begin{bmatrix} 2 & 0 & \dots & 0 & 1 \\ 0 & 2 & \dots & 0 & 1 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & 2 & 1 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

Réduction au sens de Minkowski : Pour tout i, prendre $\mathbf{b}_i$ de longueur minimale tel que $(\mathbf{b}_j)_{j \leq i}$ puisse être complété en une base.

- Définition très peu maniable
- Meilleurs résultats de qualités connus peu encourageants

  Van der Waerden (1956) : $\|\mathbf{b}_i\| \leq 2^{O(n)} \cdot \lambda_i(L)$, pour tout i.

Une approche infructueuse : tenter de minimiser les normes

Idéalement, on pourrait souhaiter une base atteignant les minimas.

- Les minimas sont atteints par des vecteurs lin. ind. du réseau.
- Mais il existe des réseaux dont aucune base n'atteint les minimas.

$$\begin{bmatrix} 2 & 0 & \dots & 0 & 1 \\ 0 & 2 & \dots & 0 & 1 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & 2 & 1 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

Réduction au sens de Minkowski : Pour tout i, prendre $\mathbf{b}_i$ de longueur minimale tel que $(\mathbf{b}_j)_{j \leq i}$ puisse être complété en une base.

- Définition très peu maniable
- Meilleurs résultats de qualités connus peu encourageants

  Van der Waerden (1956) : $\|\mathbf{b}_i\| \leq 2^{O(n)} \cdot \lambda_i(L)$, pour tout i.

Une approche infructueuse : tenter de minimiser les normes

Idéalement, on pourrait souhaiter une base atteignant les minimas.

- Les minimas sont atteints par des vecteurs lin. ind. du réseau.
- Mais il existe des réseaux dont aucune base n'atteint les minimas.

$$\begin{bmatrix} 2 & 0 & \dots & 0 & 1 \\ 0 & 2 & \dots & 0 & 1 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & 2 & 1 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

Réduction au sens de Minkowski : Pour tout i, prendre $\mathbf{b}_i$ de longueur minimale tel que $(\mathbf{b}_j)_{j \leq i}$ puisse être complété en une base.

- Définition très peu maniable
- Meilleurs résultats de qualités connus peu encourageants

  Van der Waerden (1956) : $\|\mathbf{b}_i\| \leq 2^{O(n)} \cdot \lambda_i(L)$, pour tout i.

Une approche infructueuse : tenter de minimiser les normes

Idéalement, on pourrait souhaiter une base atteignant les minimas.

- Les minimas sont atteints par des vecteurs lin. ind. du réseau.
- Mais il existe des réseaux dont aucune base n'atteint les minimas.

$$\begin{bmatrix} 2 & 0 & \dots & 0 & 1 \\ 0 & 2 & \dots & 0 & 1 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & 2 & 1 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

Réduction au sens de Minkowski : Pour tout i, prendre $\mathbf{b}_i$ de longueur minimale tel que $(\mathbf{b}_j)_{j \leq i}$ puisse être complété en une base.

- Définition très peu maniable
- Meilleurs résultats de qualités connus peu encourageants
  Van der Waerden (1956) : $\|\mathbf{b}_i\| \leq 2^{O(n)} \cdot \lambda_i(L)$, pour tout i.

Une approche infructueuse : tenter de minimiser les normes

Idéalement, on pourrait souhaiter une base atteignant les minimas.

- Les minimas sont atteints par des vecteurs lin. ind. du réseau.
- Mais il existe des réseaux dont aucune base n'atteint les minimas.

$$\begin{bmatrix} 2 & 0 & \dots & 0 & 1 \\ 0 & 2 & \dots & 0 & 1 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & 2 & 1 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

Réduction au sens de Minkowski : Pour tout i, prendre $\mathbf{b}_i$ de longueur minimale tel que $(\mathbf{b}_j)_{j \leq i}$ puisse être complété en une base.

- Définition très peu maniable
- Meilleurs résultats de qualités connus peu encourageants

  Van der Waerden (1956) : $\|\mathbf{b}_i\| \leq 2^{O(n)} \cdot \lambda_i(L)$, pour tout i.

De n^2 à $n(n+1)/2$ variables : factorisation QR

Objectif de la réduction de réseau

Étant donnée $B \in \mathbb{R}^{n \times n}$ de rang plein, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q.
les colonnes de $B \cdot U$ aient de petites normes

$\|\cdot\|$ stable par rotations $\Rightarrow$ on travaille sur des matrices triangulaires :

$$B = \underbrace{Q}_{\text{orthogonale}} \cdot \underbrace{R}_{\text{triangulaire sup}} \qquad \text{(factorisation QR)}$$

- r_{11} est la norme de $\mathbf{b}_1$
- r_{ii} est la norme de la projection de $\mathbf{b}_i$ orthogonalement à $(\mathbf{b}_j)_{j<i}$

Taille binaire de R : $O(n^2 \cdot n \log \|B\|)$ bits, avec $\|B\| = \max \|\mathbf{b}_i\|$.
Coût de calculer R : $O(n^3 \cdot \mathcal{M}(n \log \|B\|))$ opérations binaires.

De n^2 à $n(n+1)/2$ variables : factorisation QR

Objectif de la réduction de réseau

Étant donnée $B \in \mathbb{R}^{n \times n}$ de rang plein, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q.
les colonnes de $B \cdot U$ aient de petites normes

$\|\cdot\|$ stable par rotations $\Rightarrow$ on travaille sur des matrices triangulaires :

$$B = \underbrace{Q}_{\text{orthogonale}} \cdot \underbrace{R}_{\text{triangulaire sup}} \qquad \textbf{(factorisation QR)}$$

- r_{11} est la norme de $\mathbf{b}_1$
- r_{ii} est la norme de la projection de $\mathbf{b}_i$ orthogonalement à $(\mathbf{b}_j)_{j<i}$

Taille binaire de R : $O(n^2 \cdot n \log \|B\|)$ bits, avec $\|B\| = \max \|\mathbf{b}_i\|$.
Coût de calculer R : $O(n^3 \cdot \mathcal{M}(n \log \|B\|))$ opérations binaires.

De n^2 à $n(n+1)/2$ variables : factorisation QR

Objectif de la réduction de réseau

Étant donnée $B \in \mathbb{R}^{n \times n}$ de rang plein, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q.
les colonnes de $B \cdot U$ aient de petites normes

$\|\cdot\|$ stable par rotations $\Rightarrow$ on travaille sur des matrices triangulaires :

$$B = \underbrace{Q}_{\text{orthogonale}} \cdot \underbrace{R}_{\text{triangulaire sup}} \qquad \textbf{(factorisation QR)}$$

- r_{11} est la norme de $\mathbf{b}_1$
- r_{ii} est la norme de la projection de $\mathbf{b}_i$ orthogonalement à $(\mathbf{b}_j)_{j<i}$

Taille binaire de R : $O(n^2 \cdot n \log \|B\|)$ bits, avec $\|B\| = \max \|\mathbf{b}_i\|$.
Coût de calculer R : $O(n^3 \cdot \mathcal{M}(n \log \|B\|))$ opérations binaires.

De n^2 à $n(n+1)/2$ variables : factorisation QR

Objectif de la réduction de réseau

Étant donnée $B \in \mathbb{R}^{n \times n}$ de rang plein, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q.
les colonnes de $B \cdot U$ aient de petites normes

$\|\cdot\|$ stable par rotations $\Rightarrow$ on travaille sur des matrices triangulaires :

$$B = \underbrace{Q}_{\text{orthogonale}} \cdot \underbrace{R}_{\text{triangulaire sup}} \qquad \textbf{(factorisation QR)}$$

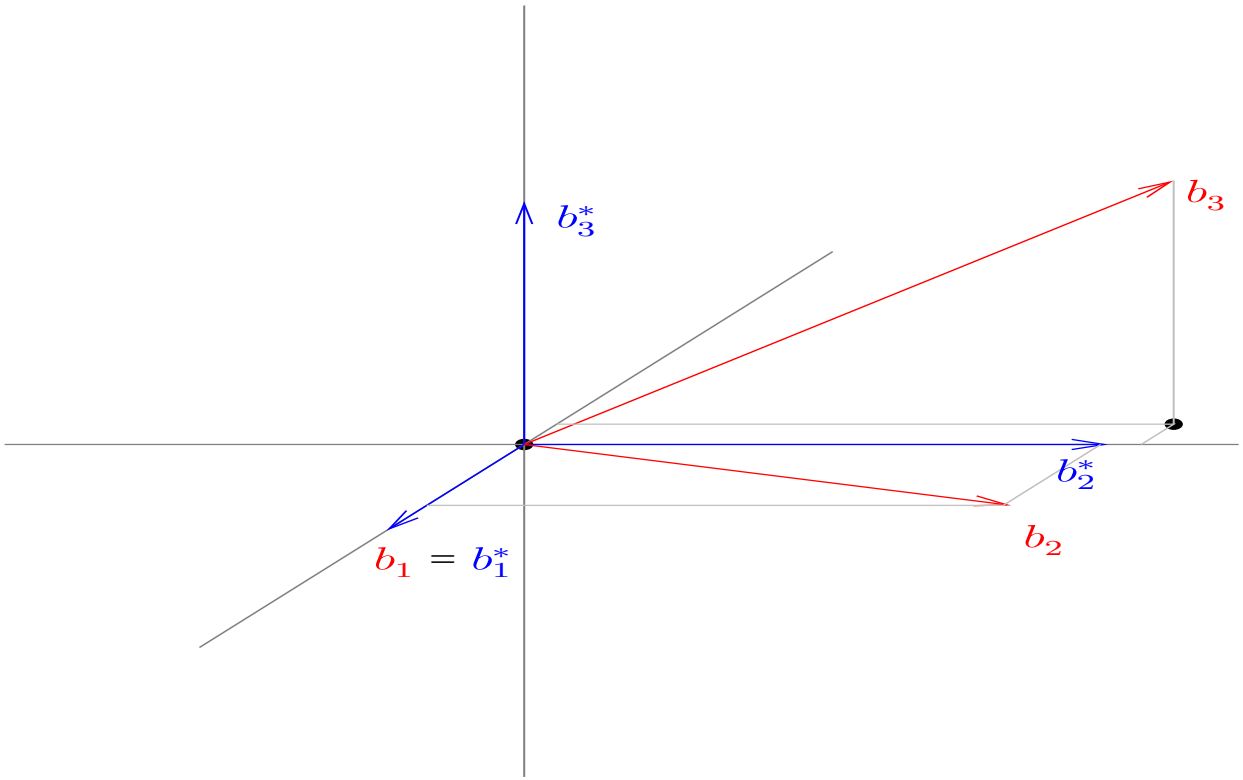
- r_{11} est la norme de $\mathbf{b}_1$
- r_{ii} est la norme de la projection de $\mathbf{b}_i$ orthogonalement à $(\mathbf{b}_j)_{j<i}$

Taille binaire de R : $O(n^2 \cdot n \log \|B\|)$ bits, avec $\|B\| = \max \|\mathbf{b}_i\|$.
Coût de calculer R : $O(n^3 \cdot \mathcal{M}(n \log \|B\|))$ opérations binaires.

La factorisation QR, ce n'est pas compliqué

QR est équivalent à l'orthogonalisation de Gram-Schmidt :

- Pour tout i, $\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j<i} \mu_{ij} \mathbf{b}_j^*$ est la projection de $\mathbf{b}_i$ orthogonalement à $\mathrm{Vec}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})$.
- On a $\|\mathbf{b}_i^*\| = r_{ii}$ et $\mu_{ij} = \frac{r_{ji}}{r_{jj}}$ pour $i > j$.



Factorisation QR et réseaux

Minimum et QR

Soit L un réseau, et $B = (\mathbf{b}_i)_i = QR$ une base de L. Alors :

$$\lambda_1(L) \geq \min_j r_{jj}.$$

Déterminant et QR

Soit L un réseau, et $B = (\mathbf{b}_i)_i = QR$ une base de L. Alors :

$$\det(L) = \prod_j r_{jj}.$$

Factorisation QR et réseaux

Minimum et QR

Soit L un réseau, et $B = (\mathbf{b}_i)_i = QR$ une base de L. Alors :

$$\lambda_1(L) \geq \min_j r_{jj}.$$

Déterminant et QR

Soit L un réseau, et $B = (\mathbf{b}_i)_i = QR$ une base de L. Alors :

$$\det(L) = \prod_j r_{jj}.$$

De $n(n+1)/2$ à n variables : la "proprification"

Objectif de la réduction de réseaux

Étant donnée $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure,
trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q. $R \cdot U$ ait un petit facteur R.

$$\begin{bmatrix} \ddots & \vdots & \dots & \vdots & \dots \\ & r_{ii} & \dots & r_{ij} & \dots \\ & & \ddots & \vdots & \dots \\ & & & r_{jj} & \dots \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & & & & \\ & 1 & & -\lfloor \frac{r_{ij}}{r_{ii}} \rceil & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \ddots \end{bmatrix}$$

$$\Rightarrow \left| r_{ij}^{(new)} \right| \leq \frac{|r_{ii}|}{2}$$

Mais les autres coefficients de R peuvent avoir changés :
s'ils étaient petits, ils ne le sont peut-être plus...

De $n(n+1)/2$ à n variables : la "proprification"

Objectif de la réduction de réseaux

Étant donnée $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure,
trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q. $R \cdot U$ ait un petit facteur R.

$$\begin{bmatrix} \ddots & \vdots & \dots & \vdots & \dots \\ & r_{ii} & \dots & r_{ij} & \dots \\ & & \ddots & \vdots & \dots \\ & & & r_{jj} & \dots \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & & & & \\ & 1 & & -\lfloor \frac{r_{ij}}{r_{ii}} \rceil & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \ddots \end{bmatrix}$$

$$\Rightarrow \left| r_{ij}^{(new)} \right| \leq \frac{|r_{ii}|}{2}$$

Mais les autres coefficients de R peuvent avoir changés :
s'ils étaient petits, ils ne le sont peut-être plus...

De $n(n+1)/2$ à n variables : la "proprification"

Objectif de la réduction de réseaux

Étant donnée $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q. $R \cdot U$ ait un petit facteur R.

$$\begin{bmatrix} \ddots & \vdots & \dots & \vdots & \dots \\ & r_{ii} & \dots & r_{ij} & \dots \\ & & \ddots & \vdots & \dots \\ & & & r_{jj} & \dots \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & & & & \\ & 1 & & -\lfloor \frac{r_{ij}}{r_{ii}} \rceil & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \ddots \end{bmatrix}$$

$$\Rightarrow \left| r_{ij}^{(new)} \right| \leq \frac{|r_{ii}|}{2}$$

Mais les autres coefficients de R peuvent avoir changés : s'ils étaient petits, ils ne le sont peut-être plus...

De $n(n+1)/2$ à n variables : la "proprification"

Objectif de la réduction de réseaux

Étant donnée $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q. $R \cdot U$ ait un petit facteur R.

$$\begin{bmatrix} \ddots & \vdots & \dots & \vdots & \dots \\ & r_{ii} & \dots & r_{ij} & \dots \\ & & \ddots & \vdots & \dots \\ & & & r_{jj} & \dots \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & & & & \\ & 1 & & -\lfloor \frac{r_{ij}}{r_{ii}} \rceil & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \ddots \end{bmatrix}$$

$$\Rightarrow \left| r_{ij}^{(new)} \right| \leq \frac{|r_{ii}|}{2}$$

Mais les autres coefficients de R peuvent avoir changés : s'ils étaient petits, ils ne le sont peut-être plus...

De $n(n+1)/2$ à n variables : la "proprification"

$$\begin{bmatrix} \ddots & \vdots & \dots & \vdots & \dots \\ & r_{ii} & \dots & r_{ij} & \dots \\ & & \ddots & \vdots & \dots \\ & & & r_{jj} & \dots \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & & & & \\ & 1 & & -\lfloor \frac{r_{ij}}{r_{ii}} \rceil & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \ddots \end{bmatrix}$$

Ça modifie le haut de la j-ème colonne $\Rightarrow$ Aller du bas vers le haut.

Base propre

Une base est propre si $|r_{ij}| \leq r_{ii}/2$ pour tous $i < j$

- La proprification permet de contrôler les tailles des coefficients extra-diagonaux à l'aide des coefficients diagonaux
- Coût : $O(n^2 \mathcal{M}(n \log \|B\|))$ opérations binaires par colonne.

De $n(n+1)/2$ à n variables : la "proprification"

$$\begin{bmatrix} \ddots & \vdots & \dots & \vdots & \dots \\ & r_{ii} & \dots & r_{ij} & \dots \\ & & \ddots & \vdots & \dots \\ & & & r_{jj} & \dots \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & & & & \\ & 1 & & -\lfloor \frac{r_{ij}}{r_{ii}} \rceil & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \ddots \end{bmatrix}$$

Ça modifie le haut de la j-ème colonne $\Rightarrow$ Aller du bas vers le haut.

Base propre

Une base est propre si $|r_{ij}| \leq r_{ii}/2$ pour tous $i < j$

- La proprification permet de contrôler les tailles des coefficients extra-diagonaux à l'aide des coefficients diagonaux
- Coût : $O(n^2 \mathcal{M}(n \log \|B\|))$ opérations binaires par colonne.

De $n(n+1)/2$ à n variables : la "proprification"

$$\begin{bmatrix} \ddots & \vdots & \dots & \vdots & \dots \\ & r_{ii} & \dots & r_{ij} & \dots \\ & & \ddots & \vdots & \dots \\ & & & r_{jj} & \dots \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & & & & \\ & 1 & & -\lfloor \frac{r_{ij}}{r_{ii}} \rceil & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \ddots \end{bmatrix}$$

Ça modifie le haut de la j-ème colonne $\Rightarrow$ Aller du bas vers le haut.

Base propre

Une base est propre si $|r_{ij}| \leq r_{ii}/2$ pour tous $i < j$

- La proprification permet de contrôler les tailles des coefficients extra-diagonaux à l'aide des coefficients diagonaux
- Coût : $O(n^2 \mathcal{M}(n \log \|B\|))$ opérations binaires par colonne.

Où en sommes-nous?

Objectif de la réduction de réseaux

Étant donnée $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q. le facteur R de $R \cdot U$ ait de petits coefficients diagonaux

Le produit des r_{ii} est constant : c'est le déterminant !

$\Rightarrow$ Les rendre petits, c'est la même chose que

- les rendre équilibrés
- les empêcher de décroître trop vite

Où en sommes-nous?

Objectif de la réduction de réseaux

Étant donnée $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q. le facteur R de $R \cdot U$ ait de petits coefficients diagonaux

Le produit des r_{ii} est constant : c'est le déterminant !

$\Rightarrow$ Les rendre petits, c'est la même chose que

- les rendre équilibrés
- les empêcher de décroître trop vite

Où en sommes-nous?

Objectif de la réduction de réseaux

Étant donnée $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q. le facteur R de $R \cdot U$ ait de petits coefficients diagonaux

Le produit des r_{ii} est constant : c'est le déterminant !

$\Rightarrow$ Les rendre petits, c'est la même chose que

- les rendre équilibrés
- les empêcher de décroître trop vite

Un objectif séduisant

Réduction HKZ

Une matrice R triangulaire supérieure est HKZ-réduite si

- elle est propre
- $r_{11} = \lambda_1(L)$ avec $L = \sum_i \mathbb{Z}\mathbf{r}_i$
- et $(r_{ij})_{i,j>1}$ est HKZ-réduite

Le théorème de Minkowski nous donne, pour tout $i \leq n$:

$$r_{ii} \leq \sqrt{n-i+1} \cdot \left(\prod_{j=i}^{n} r_{jj} \right)^{\frac{1}{n-i+1}}$$

Fixer r_{nn} permet de borner supérieurement les autres r_{ii}.
Comme c'est multiplicatif, on utilise plutôt $x_i = \log r_{ii}$.

Un objectif séduisant

Réduction HKZ

Une matrice R triangulaire supérieure est HKZ-réduite si

- elle est propre
- $r_{11} = \lambda_1(L)$ avec $L = \sum_i \mathbb{Z}\mathbf{r}_i$
- et $(r_{ij})_{i,j>1}$ est HKZ-réduite

Le théorème de Minkowski nous donne, pour tout $i \leq n$:

$$r_{ii} \leq \sqrt{n-i+1} \cdot \left(\prod_{j=i}^{n} r_{jj} \right)^{\frac{1}{n-i+1}}$$

Fixer r_{nn} permet de borner supérieurement les autres r_{ii}.
Comme c'est multiplicatif, on utilise plutôt $x_i = \log r_{ii}$.

Un objectif séduisant

Réduction HKZ

Une matrice R triangulaire supérieure est HKZ-réduite si

- elle est propre
- $r_{11} = \lambda_1(L)$ avec $L = \sum_i \mathbb{Z}\mathbf{r}_i$
- et $(r_{ij})_{i,j>1}$ est HKZ-réduite

Le théorème de Minkowski nous donne, pour tout $i \leq n$:

$$r_{ii} \leq \sqrt{n-i+1} \cdot \left(\prod_{j=i}^{n} r_{jj} \right)^{\frac{1}{n-i+1}}$$

Fixer r_{nn} permet de borner supérieurement les autres r_{ii}.
Comme c'est multiplicatif, on utilise plutôt $x_i = \log r_{ii}$.

Un objectif séduisant

Réduction HKZ

Une matrice R triangulaire supérieure est HKZ-réduite si

- elle est propre
- $r_{11} = \lambda(L)$ avec $L = \sum_i \mathbb{Z}\mathbf{r}_i$
- et $(r_{ij})_{i,j>1}$ est HKZ-réduite

Un objectif séduisant

Réduction HKZ

Une matrice R triangulaire supérieure est HKZ-réduite si

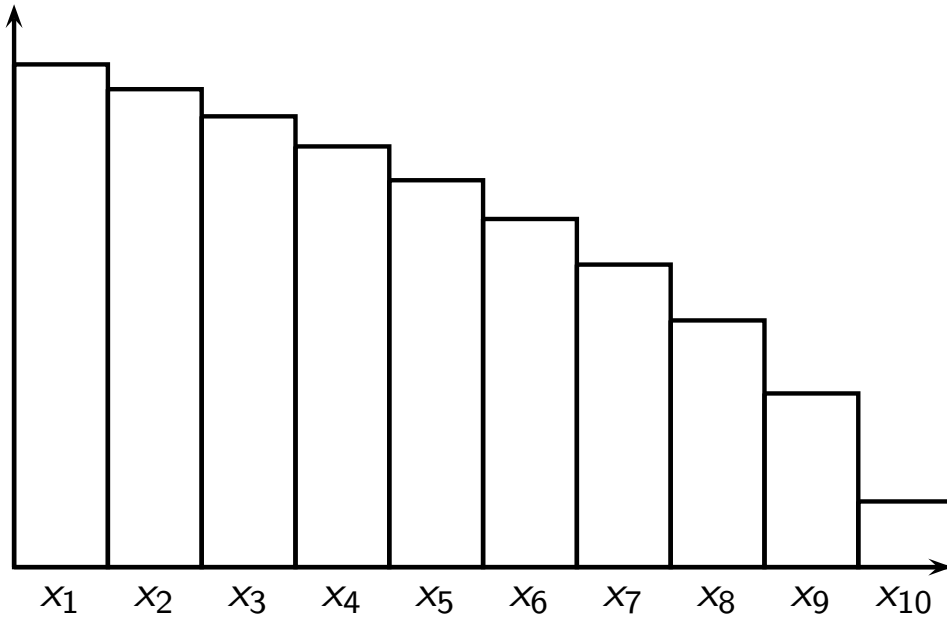
- elle est propre
- $r_{11} = \lambda(L)$ avec $L = \sum_i \mathbb{Z}\mathbf{r}_i$
- et $(r_{ij})_{i,j>1}$ est HKZ-réduite

Profil d'une base HKZ pire cas :

$$\begin{aligned} x_i &= \log r_{ii} \\ &= O(\log^2(n-i+1)) \end{aligned}$$

Coût d'une réduction HKZ :

- n résolutions de SVP.



Un objectif séduisant

Réduction HKZ

Une matrice R triangulaire supérieure est HKZ-réduite si

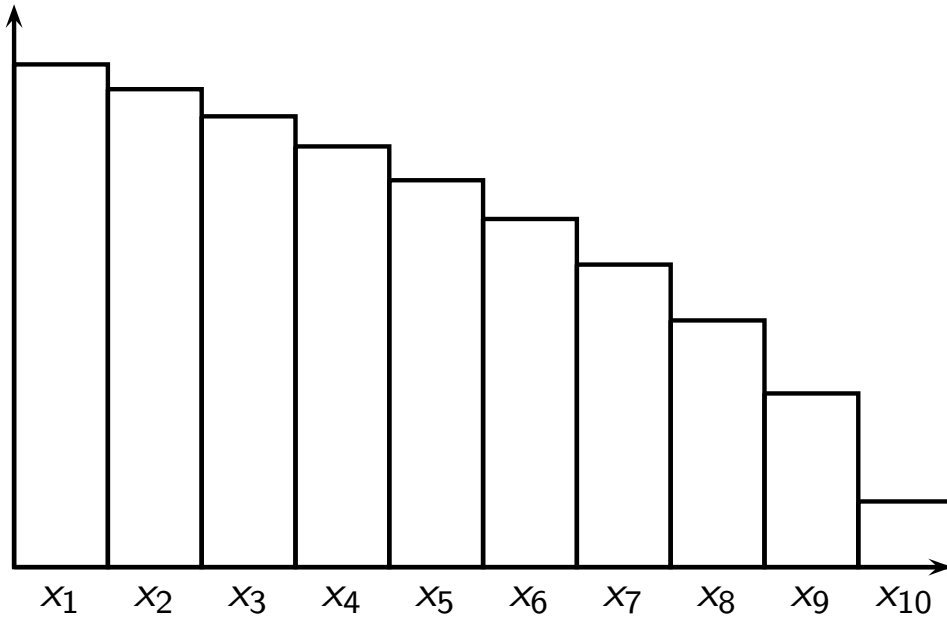
- elle est propre
- $r_{11} = \lambda(L)$ avec $L = \sum_i \mathbb{Z}\mathbf{r}_i$
- et $(r_{ij})_{i,j>1}$ est HKZ-réduite

Profil d'une base HKZ pire cas :

$$\begin{aligned} x_i &= \log r_{ii} \\ &= O(\log^2(n-i+1)) \end{aligned}$$

Coût d'une réduction HKZ :

- n résolutions de SVP.



Réduction de réseaux : les règles du jeu

Objectif de la réduction de réseaux

Étant donnée $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q. le facteur R de $R \cdot U$ ait de petits coefficients diagonaux

HKZ est trop cher... Que peut-on faire ?

1. Échanger deux vecteurs consécutifs t.q. $r_{i+1,i+1} \ll r_{i,i}$ [LLL82]
2. Équilibrer localement les coefficients diagonaux en appliquant HKZ à une sous-matrice sur la diagonale de R [Sch87]

$$\begin{bmatrix} \ddots & \cdots & \vdots \\ & \begin{bmatrix} r_{ii} & \cdots & \cdots \\ & \ddots & \vdots \\ & & r_{jj} \end{bmatrix} & \vdots \\ & & \ddots \end{bmatrix} \cdot \begin{bmatrix} Id & \cdots & \vdots \\ & \begin{bmatrix} U_{HKZ} \end{bmatrix} & \vdots \\ & & Id \end{bmatrix}$$

Multiples stratégies : adaptative, non-adaptative, gloutonne, etc.

Réduction de réseaux : les règles du jeu

Objectif de la réduction de réseaux

Étant donnée $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q. le facteur R de $R \cdot U$ ait de petits coefficients diagonaux

HKZ est trop cher... Que peut-on faire ?

1. Échanger deux vecteurs consécutifs t.q. $r_{i+1,i+1} \ll r_{i,i}$ [LLL82]
2. Équilibrer localement les coefficients diagonaux en appliquant HKZ à une sous-matrice sur la diagonale de R [Sch87]

Multiples stratégies : adaptative, non-adaptative, gloutonne, etc

Réduction de réseaux : les règles du jeu

Objectif de la réduction de réseaux

Étant donnée $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q. le facteur R de $R \cdot U$ ait de petits coefficients diagonaux

HKZ est trop cher... Que peut-on faire ?

1. Échanger deux vecteurs consécutifs t.q. $r_{i+1,i+1} \ll r_{i,i}$ [LLL82]
2. Équilibrer localement les coefficients diagonaux en appliquant HKZ à une sous-matrice sur la diagonale de R [Sch87]

$$\begin{bmatrix} \ddots & & \vdots & \\ & \begin{bmatrix} r_{ii} & \cdots & \cdots \\ & \ddots & \vdots \\ & & r_{jj} \end{bmatrix} & & \\ & & \ddots & \end{bmatrix} \cdot \begin{bmatrix} Id & \cdots & \vdots & \\ & \begin{bmatrix} & U_{HKZ} & \end{bmatrix} & \vdots & \\ & & & Id \end{bmatrix}$$

Multiples stratégies : adaptative, non-adaptative, gloutonne, etc

Réduction de réseaux : les règles du jeu

Objectif de la réduction de réseaux

Étant donnée $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q. le facteur R de $R \cdot U$ ait de petits coefficients diagonaux

HKZ est trop cher... Que peut-on faire ?

1. Échanger deux vecteurs consécutifs t.q. $r_{i+1,i+1} \ll r_{i,i}$ [LLL82]
2. Équilibrer localement les coefficients diagonaux en appliquant HKZ à une sous-matrice sur la diagonale de R [Sch87]

$$\begin{bmatrix} \ddots & & \vdots & \\ & \begin{bmatrix} r_{ii} & \cdots & \cdots \\ & \ddots & \vdots \\ & & r_{jj} \end{bmatrix} & \vdots & \\ & & \ddots & \end{bmatrix} \cdot \begin{bmatrix} Id & \cdots & \vdots & \\ & \begin{bmatrix} & U_{HKZ} & \end{bmatrix} & \vdots & \\ & & & Id \end{bmatrix}$$

Multiples stratégies : adaptative, non-adaptative, gloutonne, etc

Plan

L'algorithme LLL

Entrée : une base B d'un réseau L.

1. Calculer le facteur R.
2. Proprifier R, mettre B à jour.
3. S'il existe i t.q. $(r_{i,i+1}^2 + r_{i+1,i+1}^2)^{1/2} \leq 0.99 \cdot r_{i,i}$: échanger $\mathbf{b}_i$ et $\mathbf{b}_{i+1}$; goto 1.
4. Sinon, renvoyer B.

Pourquoi cette condition, appelée condition de Lovász ?

- Si elle est satisfaite, alors $r_{i+1,i+1} \ll r_{i,i}$.
- Avec l'échange, le nouveau $r_{i,i}$ est $(r_{i,i+1}^2 + r_{i+1,i+1}^2)^{1/2}$.

$\Rightarrow$ $r_{i,i}$ a baissé, et $r_{i,i} \cdot r_{i+1,i+1}$ est constant.

L'algorithme LLL

Entrée : une base B d'un réseau L.

1. Calculer le facteur R.
2. Proprifier R, mettre B à jour.
3. S'il existe i t.q. $(r_{i,i+1}^2 + r_{i+1,i+1}^2)^{1/2} \leq 0.99 \cdot r_{i,i}$: échanger $\mathbf{b}_i$ et $\mathbf{b}_{i+1}$; goto 1.
4. Sinon, renvoyer B.

Pourquoi cette condition, appelée condition de Lovász ?

- Si elle est satisfaite, alors $r_{i+1,i+1} \ll r_{i,i}$.
- Avec l'échange, le nouveau $r_{i,i}$ est $(r_{i,i+1}^2 + r_{i+1,i+1}^2)^{1/2}$.

$\Rightarrow$ $r_{i,i}$ a baissé, et $r_{i,i} \cdot r_{i+1,i+1}$ est constant.

La réduction de Lenstra, Lenstra et Lovász

Une base $B = (\mathbf{b}_i)_{i \leq n} = QR$ est LLL-réduite si

- $\forall i, j : \ |r_{ij}| \leq r_{i,i}/2$
- $\forall i : \ (r_{i,i+1}^2 + r_{i+1,i+1}^2)^{1/2} > 0.99 \cdot r_{i,i}$

La réduction de Lenstra, Lenstra et Lovász

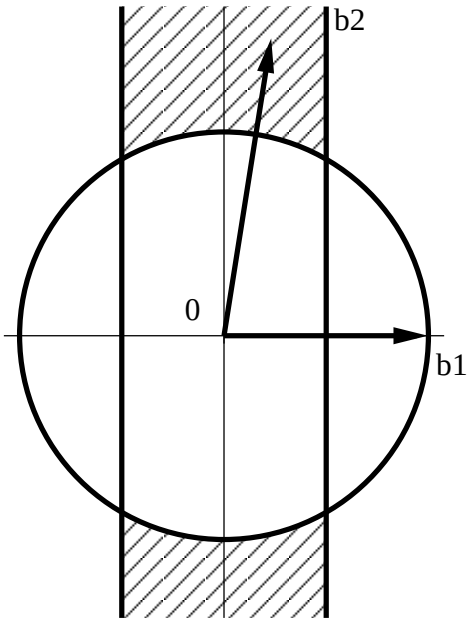
Une base $B = (\mathbf{b}_i)_{i \leq n} = QR$ est LLL-réduite si

- $\forall i, j : \ |r_{ij}| \leq r_{i,i}/2$
- $\forall i : \ (r_{i,i+1}^2 + r_{i+1,i+1}^2)^{1/2} > 0.99 \cdot r_{i,i}$

Les $r_{i,i}$ ne peuvent pas décroître trop vite.

$\Rightarrow \ \lambda_1(L) \leq \|\mathbf{b}_1\| \leq 2^{n/2} \cdot \lambda_1(L)$

$\Rightarrow \ \|\mathbf{b}_1\| \leq 2^{n/4} \cdot (\det L)^{1/n}$



La réduction de Lenstra, Lenstra et Lovász

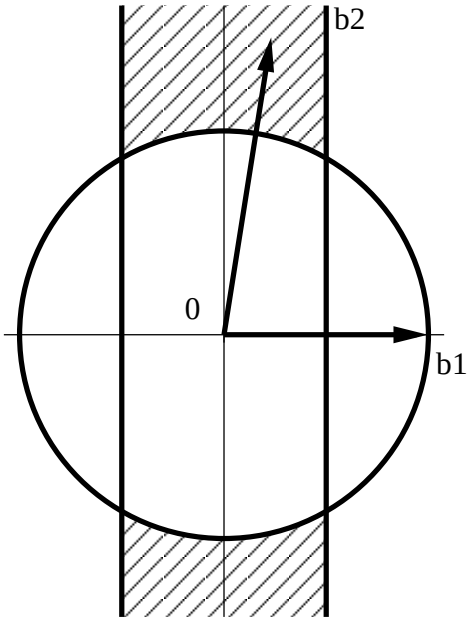
Une base $B = (\mathbf{b}_i)_{i \leq n} = QR$ est LLL-réduite si

- $\forall i, j : \ |r_{ij}| \leq r_{i,i}/2$
- $\forall i : \ (r_{i,i+1}^2 + r_{i+1,i+1}^2)^{1/2} > 0.99 \cdot r_{i,i}$

Les $r_{i,i}$ ne peuvent pas décroître trop vite.

$\Rightarrow \ \lambda_1(L) \leq \|\mathbf{b}_1\| \leq 2^{n/2} \cdot \lambda_1(L)$

$\Rightarrow \ \|\mathbf{b}_1\| \leq 2^{n/4} \cdot (\det L)^{1/n}$



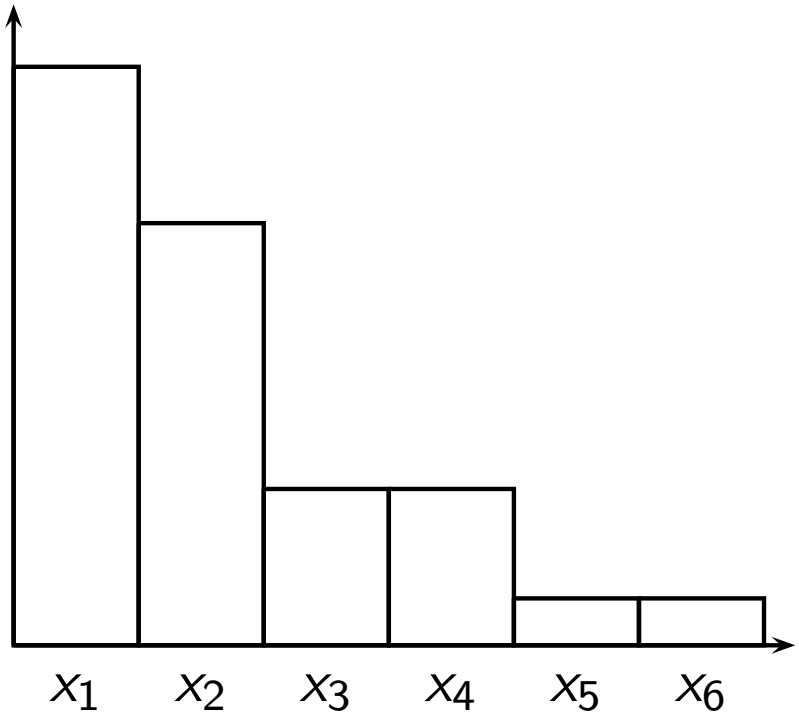
Modèle tas-de-sable" de LLL [MadVal10]

LLL : si $r_{i,i} \gg r_{i+1,i+1}$, faire $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

$\Rightarrow r_{i,i}$ décroît d'au moins un facteur constant.

$\Rightarrow x_{i,i} = \log r_{i,i}$ décroît d'au moins une constante.

Terminaison ?



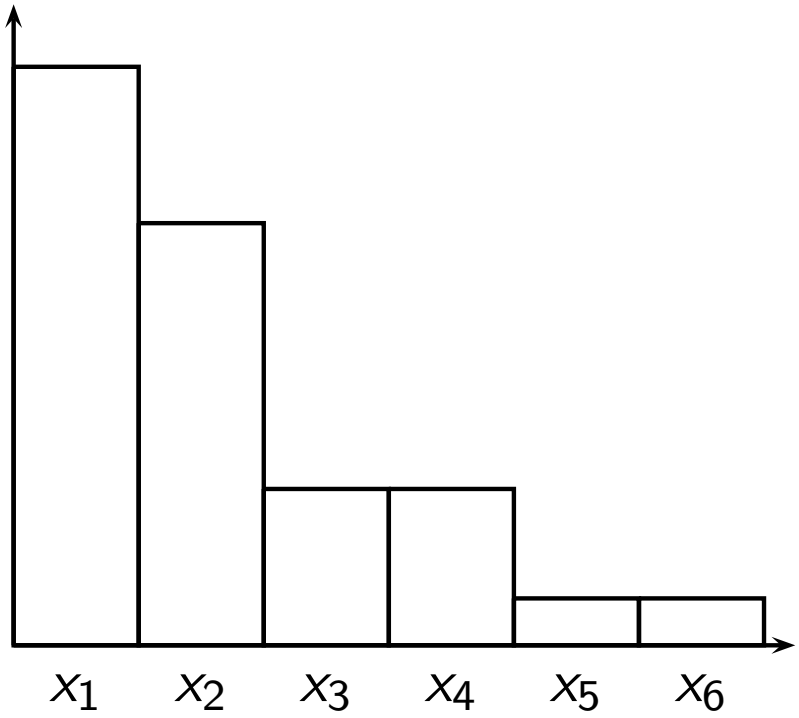
Modèle tas-de-sable" de LLL [MadVal10]

LLL : si $r_{i,i} \gg r_{i+1,i+1}$, faire $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

$\Rightarrow r_{i,i}$ décroît d'au moins un facteur constant.

$\Rightarrow x_{i,i} = \log r_{i,i}$ décroît d'au moins une constante.

Terminaison ?



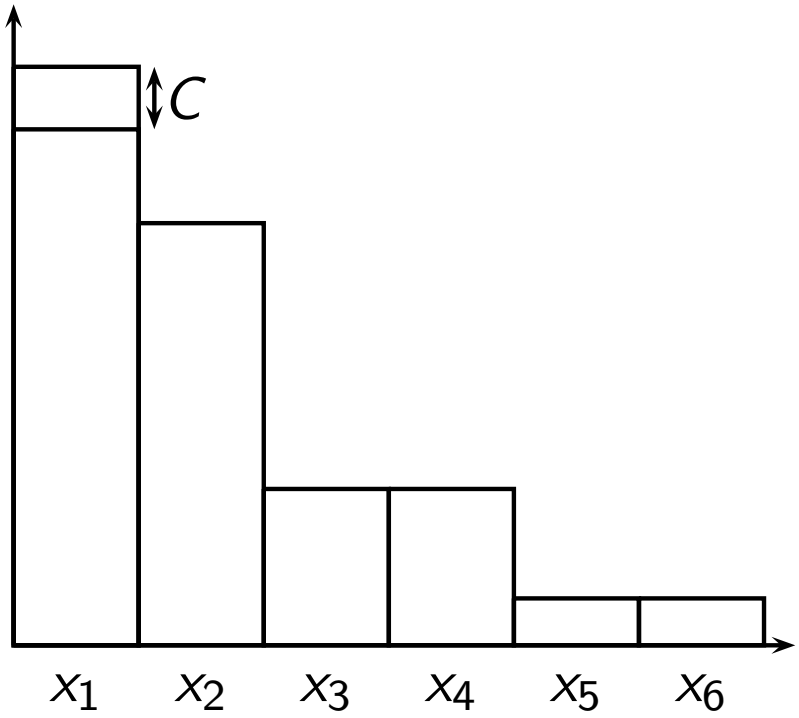
Modèle tas-de-sable" de LLL [MadVal10]

LLL : si $r_{i,i} \gg r_{i+1,i+1}$, faire $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

$\Rightarrow r_{i,i}$ décroît d'au moins un facteur constant.

$\Rightarrow x_{i,i} = \log r_{i,i}$ décroît d'au moins une constante.

Terminaison ?



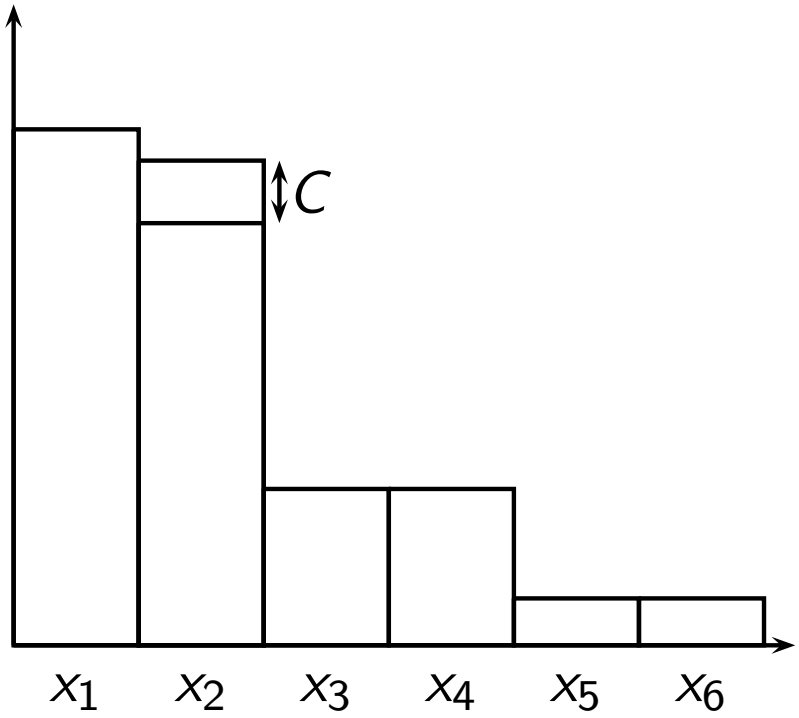
Modèle tas-de-sable" de LLL [MadVal10]

LLL : si $r_{i,i} \gg r_{i+1,i+1}$, faire $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

$\Rightarrow r_{i,i}$ décroît d'au moins un facteur constant.

$\Rightarrow x_{i,i} = \log r_{i,i}$ décroît d'au moins une constante.

Terminaison ?



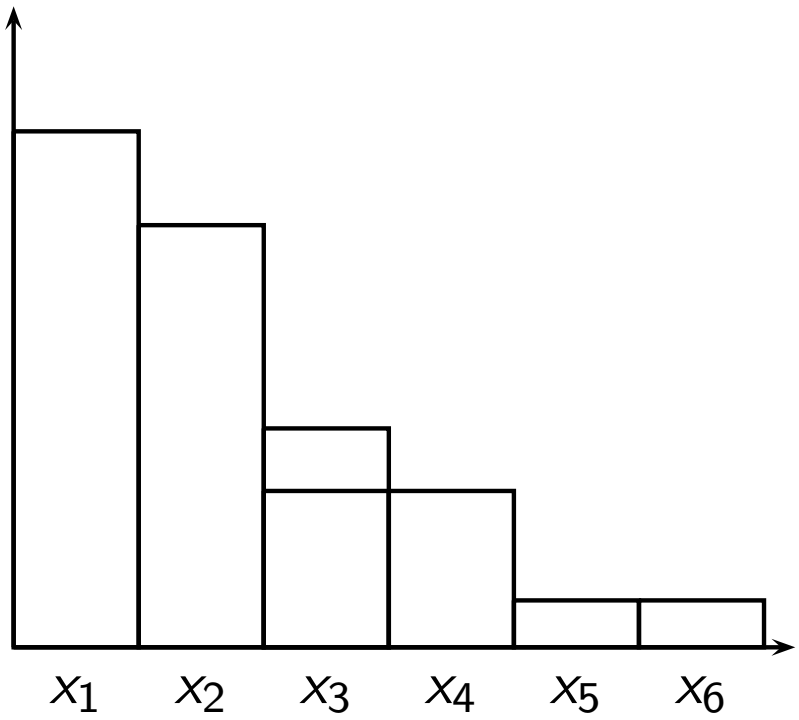
Modèle tas-de-sable" de LLL [MadVal10]

LLL : si $r_{i,i} \gg r_{i+1,i+1}$, faire $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

$\Rightarrow r_{i,i}$ décroît d'au moins un facteur constant.

$\Rightarrow x_{i,i} = \log r_{i,i}$ décroît d'au moins une constante.

Terminaison ?



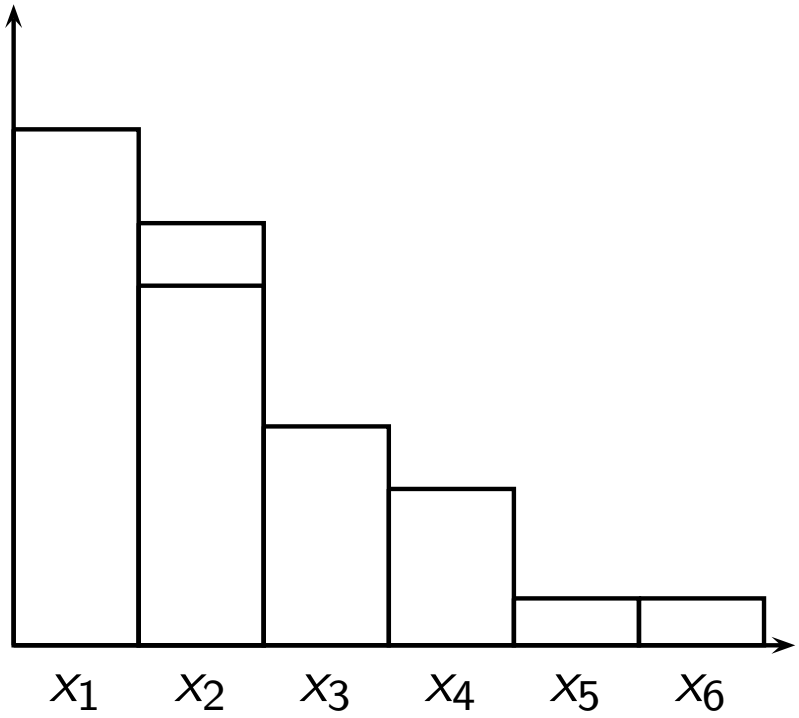
Modèle tas-de-sable" de LLL [MadVal10]

LLL : si $r_{i,i} \gg r_{i+1,i+1}$, faire $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

$\Rightarrow r_{i,i}$ décroît d'au moins un facteur constant.

$\Rightarrow x_{i,i} = \log r_{i,i}$ décroît d'au moins une constante.

Terminaison ?



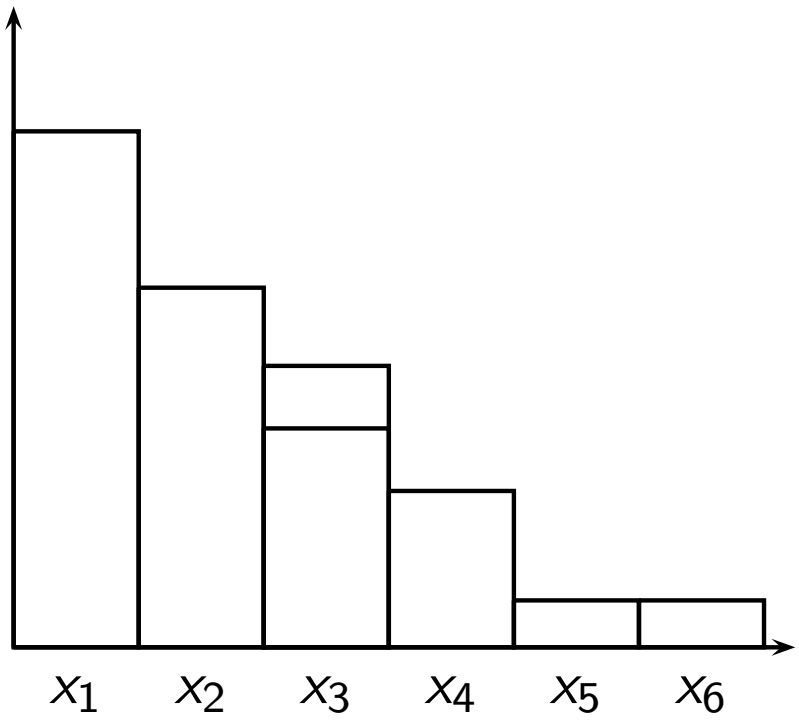
Modèle tas-de-sable" de LLL [MadVal10]

LLL : si $r_{i,i} \gg r_{i+1,i+1}$, faire $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

$\Rightarrow r_{i,i}$ décroît d'au moins un facteur constant.

$\Rightarrow x_{i,i} = \log r_{i,i}$ décroît d'au moins une constante.

Terminaison ?



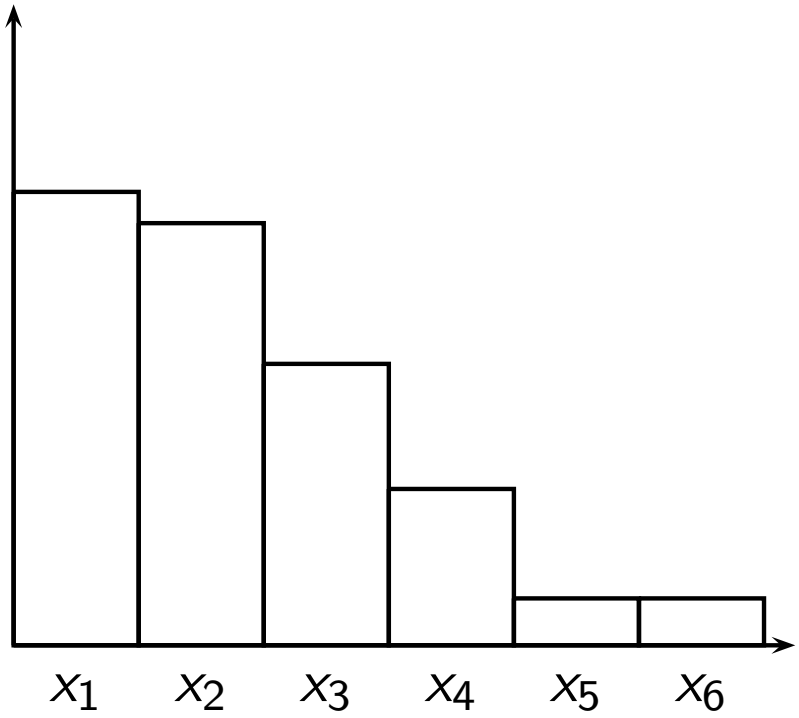
Modèle tas-de-sable" de LLL [MadVal10]

LLL : si $r_{i,i} \gg r_{i+1,i+1}$, faire $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

$\Rightarrow r_{i,i}$ décroît d'au moins un facteur constant.

$\Rightarrow x_{i,i} = \log r_{i,i}$ décroît d'au moins une constante.

Terminaison ?



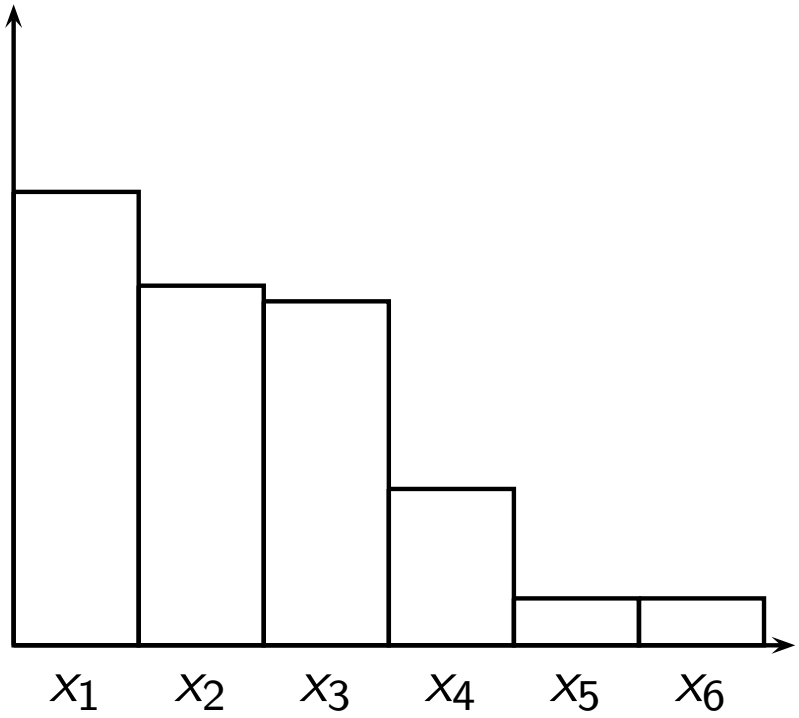
Modèle tas-de-sable" de LLL [MadVal10]

LLL : si $r_{i,i} \gg r_{i+1,i+1}$, faire $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

$\Rightarrow r_{i,i}$ décroît d'au moins un facteur constant.

$\Rightarrow x_{i,i} = \log r_{i,i}$ décroît d'au moins une constante.

Terminaison ?



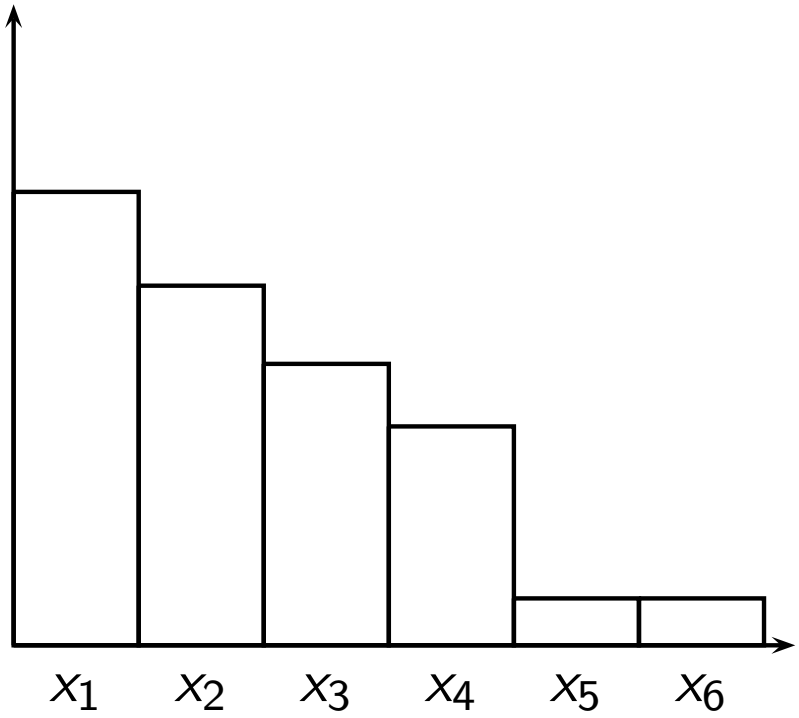
Modèle tas-de-sable" de LLL [MadVal10]

LLL : si $r_{i,i} \gg r_{i+1,i+1}$, faire $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

$\Rightarrow r_{i,i}$ décroît d'au moins un facteur constant.

$\Rightarrow x_{i,i} = \log r_{i,i}$ décroît d'au moins une constante.

Terminaison ?



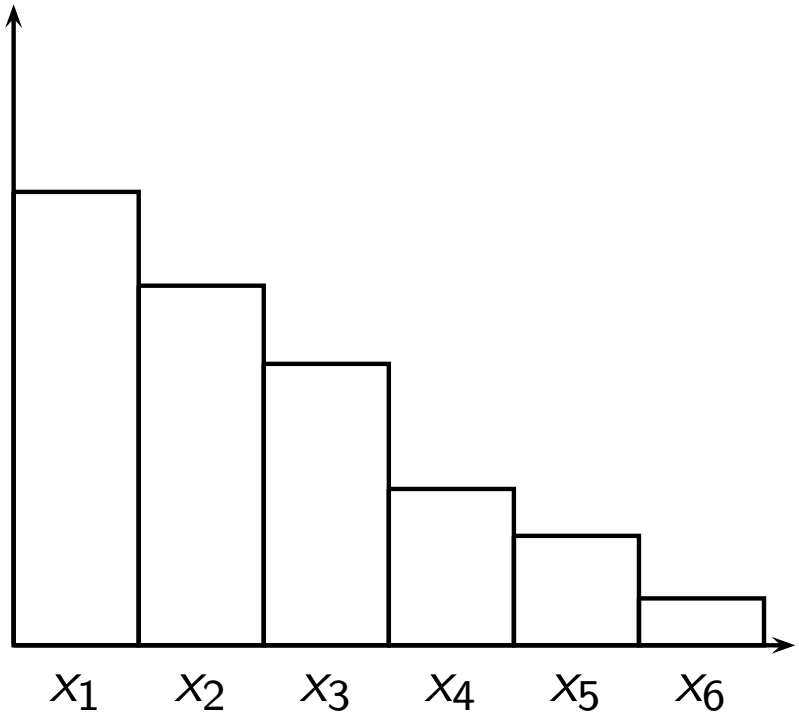
Modèle tas-de-sable" de LLL [MadVal10]

LLL : si $r_{i,i} \gg r_{i+1,i+1}$, faire $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

$\Rightarrow r_{i,i}$ décroît d'au moins un facteur constant.

$\Rightarrow x_{i,i} = \log r_{i,i}$ décroît d'au moins une constante.

Terminaison ?



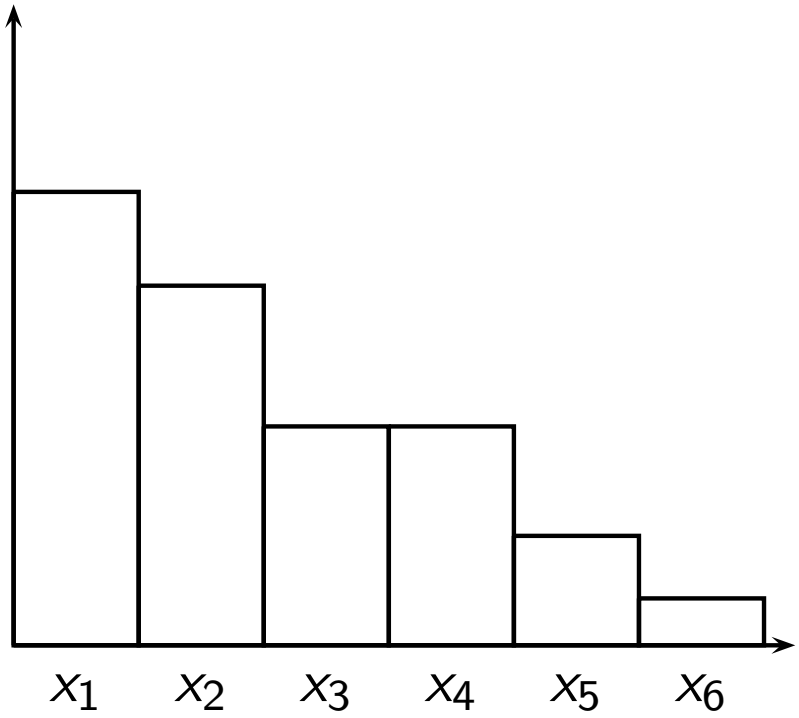
Modèle tas-de-sable" de LLL [MadVal10]

LLL : si $r_{i,i} \gg r_{i+1,i+1}$, faire $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

$\Rightarrow r_{i,i}$ décroît d'au moins un facteur constant.

$\Rightarrow x_{i,i} = \log r_{i,i}$ décroît d'au moins une constante.

Terminaison ?



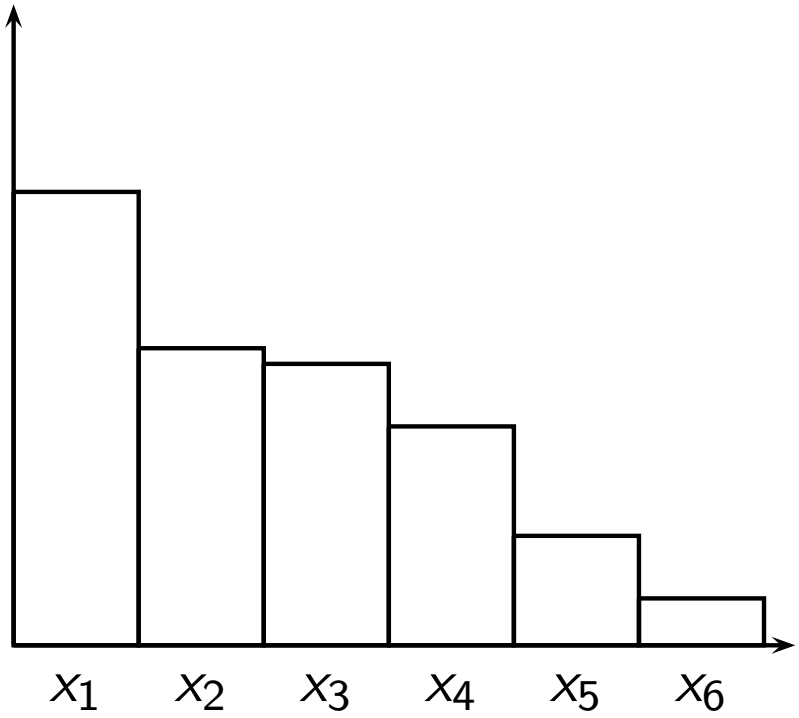
Modèle tas-de-sable" de LLL [MadVal10]

LLL : si $r_{i,i} \gg r_{i+1,i+1}$, faire $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

$\Rightarrow r_{i,i}$ décroît d'au moins un facteur constant.

$\Rightarrow x_{i,i} = \log r_{i,i}$ décroît d'au moins une constante.

Terminaison ?



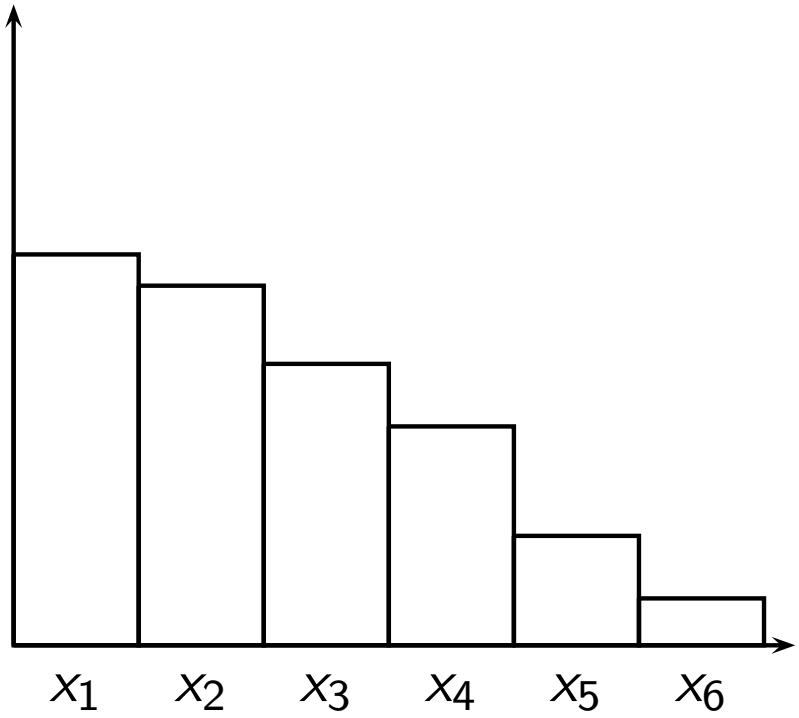
Modèle tas-de-sable" de LLL [MadVal10]

LLL : si $r_{i,i} \gg r_{i+1,i+1}$, faire $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

$\Rightarrow r_{i,i}$ décroît d'au moins un facteur constant.

$\Rightarrow x_{i,i} = \log r_{i,i}$ décroît d'au moins une constante.

Terminaison ?



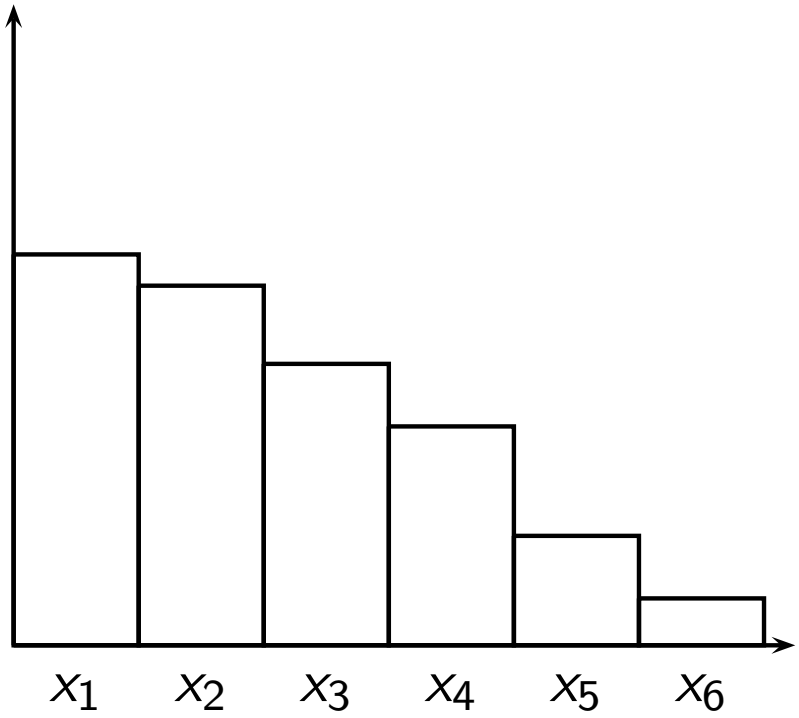
Modèle tas-de-sable" de LLL [MadVal10]

LLL : si $r_{i,i} \gg r_{i+1,i+1}$, faire $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

$\Rightarrow r_{i,i}$ décroît d'au moins un facteur constant.

$\Rightarrow x_{i,i} = \log r_{i,i}$ décroît d'au moins une constante.

Terminaison ? $\sum_{i \leq n}(n-i+1)x_i \searrow$.



Analyse de complexité de LLL

- Nombre d'itérations de boucles : $O(n^2 \log \|B\|)$.
- Coût d'une itération : $O(n^2 \cdot \mathcal{M}(n \log \|B\|))$.

$\Rightarrow$ $O(n^4 \log \|B\| \cdot \mathcal{M}(n \log \|B\|))$.

[Kaltofen83] : Le même algorithme coûte en fait:

$$O\left(n^4 \log \|B\| \cdot \frac{n \log \|B\|}{n + \log \|B\|} \mathcal{M}(n + \log \|B\|)\right) \text{ op. binaires.}$$

Analyse de complexité de LLL

- Nombre d'itérations de boucles : $O(n^2 \log \|B\|)$.
- Coût d'une itération : $O(n^2 \cdot \mathcal{M}(n \log \|B\|))$.

$\Rightarrow$ $O(n^4 \log \|B\| \cdot \mathcal{M}(n \log \|B\|))$.

[Kaltofen83] : Le même algorithme coûte en fait:

$$O\left(n^4 \log \|B\| \cdot \frac{n \log \|B\|}{n + \log \|B\|} \mathcal{M}(n + \log \|B\|)\right) \text{ op. binaires.}$$

Analyse de complexité de LLL

- Nombre d'itérations de boucles : $O(n^2 \log \|B\|)$.
- Coût d'une itération : $O(n^2 \cdot \mathcal{M}(n \log \|B\|))$.

$\Rightarrow$ $O(n^4 \log \|B\| \cdot \mathcal{M}(n \log \|B\|))$.

[Kaltofen83] : Le même algorithme coûte en fait:

$$O\left(n^4 \log \|B\| \cdot \frac{n \log \|B\|}{n + \log \|B\|} \mathcal{M}(n + \log \|B\|)\right) \text{ op. binaires.}$$