

Modélisations de l'algorithme LLL et de ses entrées

Loïck LHOTE

GREYC, ENSICAEN, CNRS-UMR 6072

Journées DynA3S

Paris, 4 février 2014



- 1 Algorithme LLL
 - Réduction des réseaux
 - Exemple d'application
 - Algorithme LLL
- 2 Modélisations de l'algorithme LLL et de ses entrées
 - Modèles d'exécution (part 1)
 - Modèles d'entrée
- 3 Résultats obtenus dans les différents modèles
 - Modèle M1 : cfg/tas de sable
 - Modèle M1 et modèles d'entrée
 - Modèle M2 : système dynamique de $\mathbb{R}^d$
 - Modèle M3 : systèmes dynamiques probabilistes de $\mathbb{R}^d$
- 4 Conclusion

Généralisations de l'algorithme d'Euclide



Généralisations de l'algorithme d'Euclide

PGCD

Approximation Rationnelle Simultanée

Problème :

Etant donné $\vec{y} \in \mathbb{R}^n$, trouver $q \in \mathbb{Z}$ avec $q \leq M$ et $\vec{p} \in \mathbb{Z}^n$ tels que $\|q \cdot \vec{y} - \vec{p}\|$ est petit.

Développement en fractions continues

$$\frac{u}{v} = \frac{1}{m_1 + \frac{1}{m_2 + \frac{1}{\ddots \frac{1}{m_{p-1} + \frac{1}{m_p + 0}}}}}$$

Généralisations de l'algorithme d'Euclide

Réduction des réseaux

- Algorithmes LLL, HKZ, BKZ, ...
- Modélisation des algorithmes (tas de sable, CFG, etc.)
- Modélisation des entrées

La modélisation des algorithmes conduit à de nouveaux systèmes dynamiques

PGCD

Approximation Rationnelle Simultanée

Problème :

Etant donné $\vec{y} \in \mathbb{R}^n$, trouver $q \in \mathbb{Z}$ avec $q \leq M$ et $\vec{p} \in \mathbb{Z}^n$ tels que $\|q \cdot \vec{y} - \vec{p}\|$ est petit.

Développement en fractions continues

$$\frac{u}{v} = \frac{1}{m_1 + \frac{1}{m_2 + \frac{1}{\ddots \frac{1}{m_{p-1} + \frac{1}{m_p + 0}}}}}$$

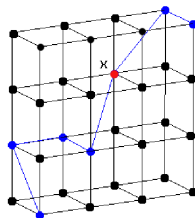
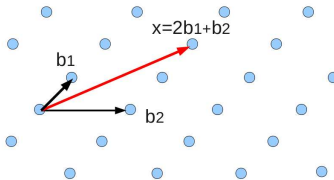
Réseaux euclidiens

Un **réseau** euclidien de $\mathbb{R}^n$ est un sous-groupe discret de $\mathbb{R}^n$.

Si $B := (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_d)$ est un système de d vecteurs linéairement indépendants de $\mathbb{R}^n$, le réseau engendré par B est

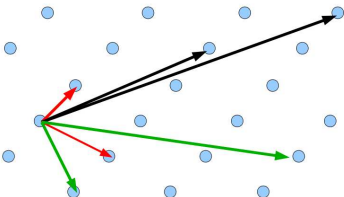
$$\mathcal{L} := \{\mathbf{x} \in \mathbb{R}^n; \quad \mathbf{x} = \sum_{i=1}^d x_i \mathbf{b}_i, \quad x_i \in \mathbb{Z}\}$$

Le système $B := (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_d)$ est une **base** du réseau $\mathcal{L}$
 d : **dimension** du réseau.



Réseaux euclidiens

Un réseau possède
une infinité de bases



Problème essentiel :

Trouver une “bonne base” du réseau $\mathcal{L}$
avec des vecteurs **assez courts** et **assez orthogonaux**
à partir d’une base quelconque de $\mathcal{L}$.

[Problème SVP]

Peut-on trouver une base contenant un plus court
vecteur non nul du réseau ?

But de la réduction : trouver en un temps “raisonnable” une “assez bonne” base.

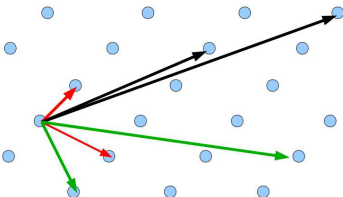
L’algorithme LLL conçu en 1982 par A. Lenstra, H. Lenstra et L. Lovász
trouve en **temps polynomial** une **assez bonne** base d’un réseau.

pour $d = 2$: l’algorithme de Gauss trouve la meilleure base

pour $d \geq 3$: l’algorithme LLL généralise l’algorithme de Gauss

Réseaux euclidiens

Un réseau possède
une infinité de bases



Problème essentiel :

Trouver une “bonne base” du réseau $\mathcal{L}$
avec des vecteurs **assez courts** et **assez orthogonaux**
à partir d’une base quelconque de $\mathcal{L}$.

[Problème SVP]

Peut-on trouver une base contenant un plus court
vecteur non nul du réseau ?

But de la réduction : trouver en un temps “raisonnable” une “assez bonne” base.

L’algorithme LLL conçu en 1982 par A. Lenstra, H. Lenstra et L. Lovász
trouve en **temps polynomial** une **assez bonne** base d’un réseau.

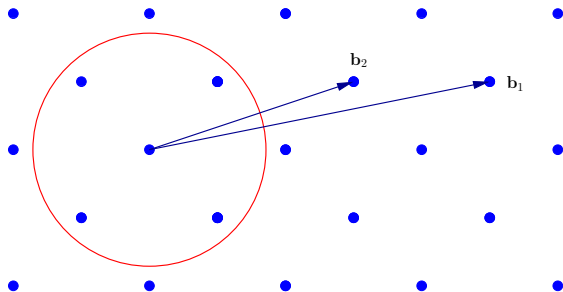
pour $d = 2$: l’algorithme de Gauss trouve la meilleure base

pour $d \geq 3$: l’algorithme LLL généralise l’algorithme de Gauss

Programmation Linéaire Entière (H. Lenstra)

Une application importante pour utiliser une base réduite

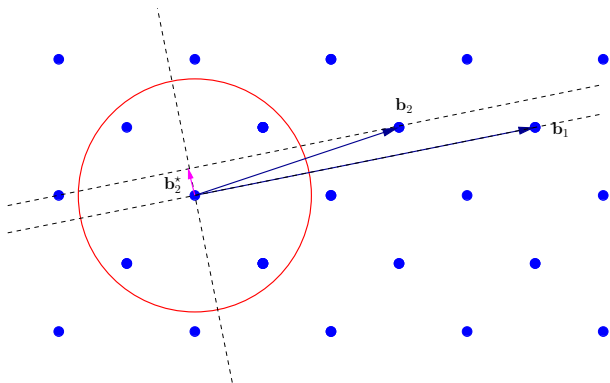
Problème : Énumérer tous les vecteurs d'un réseau $\mathcal{L}$ à l'intérieur d'une boule.



Programmation Linéaire Entière (H. Lenstra)

Une application importante pour utiliser une base réduite

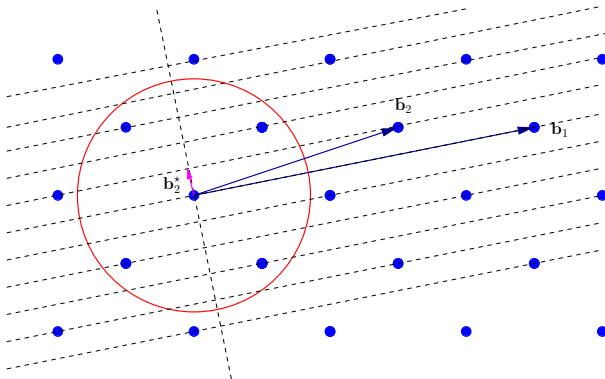
Problème : Énumérer tous les vecteurs d'un réseau $\mathcal{L}$ à l'intérieur d'une boule.



Programmation Linéaire Entière (H. Lenstra)

Une application importante pour utiliser une base réduite

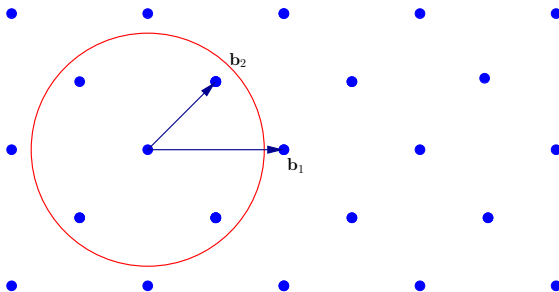
Problème : Énumérer tous les vecteurs d'un réseau $\mathcal{L}$ à l'intérieur d'une boule.



Programmation Linéaire Entière (H. Lenstra)

Une application importante pour utiliser une base réduite

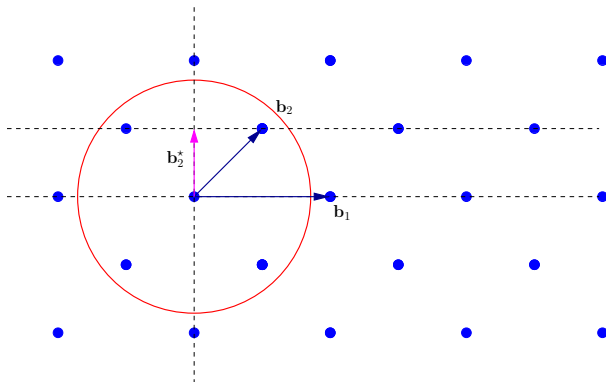
Problème : Énumérer tous les vecteurs d'un réseau $\mathcal{L}$ à l'intérieur d'une boule.



Programmation Linéaire Entière (H. Lenstra)

Une application importante pour utiliser une base réduite

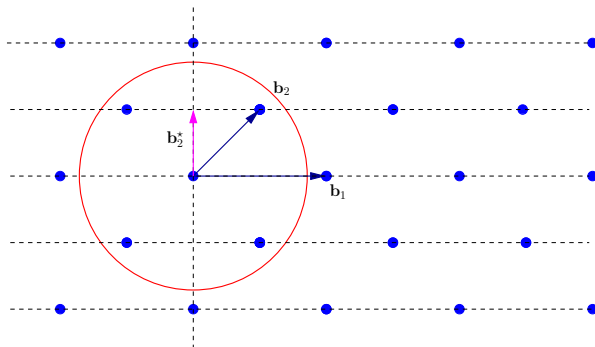
Problème : Énumérer tous les vecteurs d'un réseau $\mathcal{L}$ à l'intérieur d'une boule.



Programmation Linéaire Entière (H. Lenstra)

Une application importante pour utiliser une base réduite

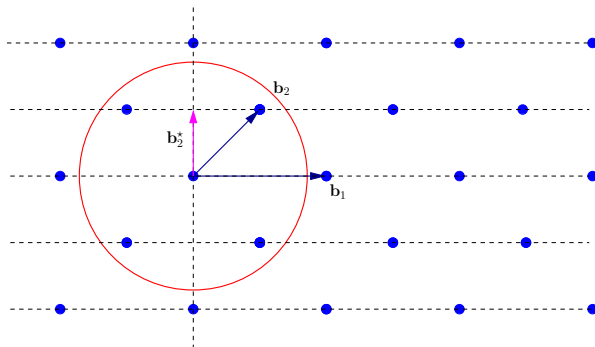
Problème : Énumérer tous les vecteurs d'un réseau $\mathcal{L}$ à l'intérieur d'une boule.



Programmation Linéaire Entière (H. Lenstra)

Une application importante pour utiliser une base réduite

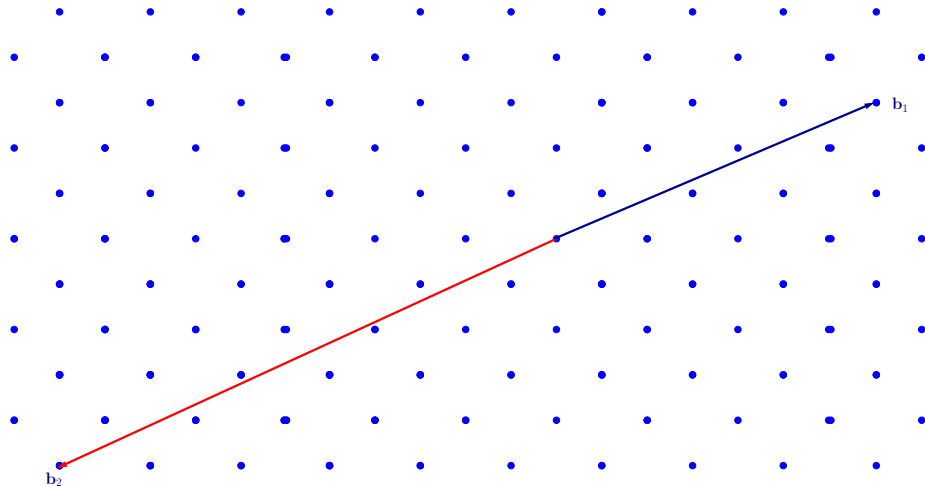
Problème : Énumérer tous les vecteurs d'un réseau $\mathcal{L}$ à l'intérieur d'une boule.



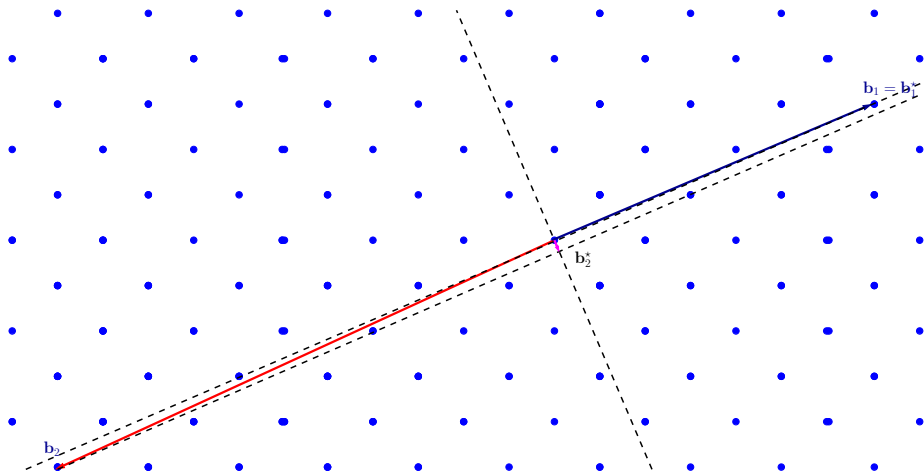
Conclusion : utiliser une base réduite conduit à des calculs plus efficaces.

Remarque : b_2^* doit être le plus long possible.

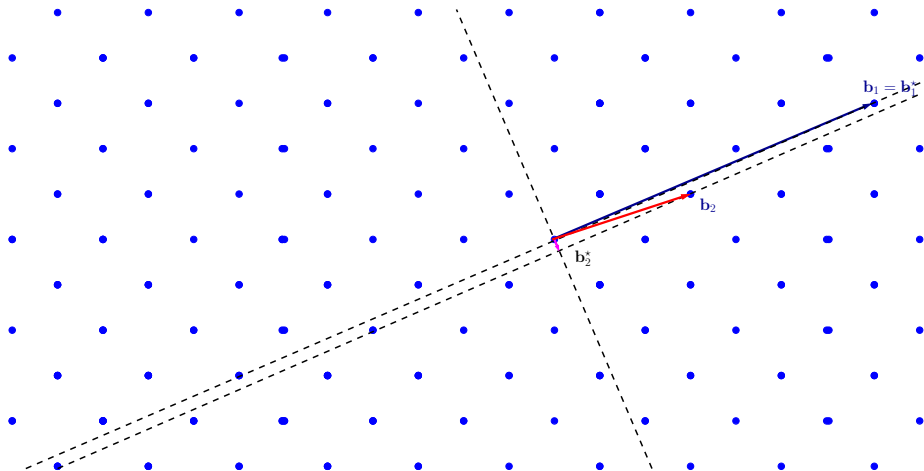
Principes de l'algorithme



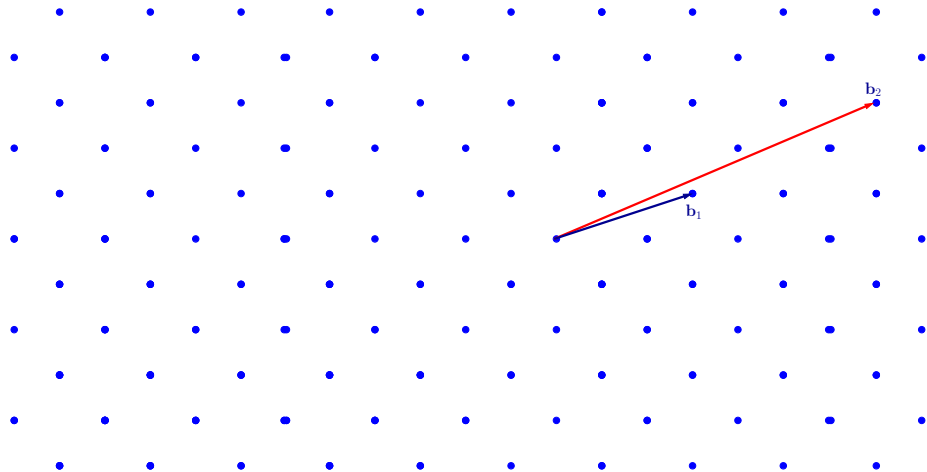
Principes de l'algorithme



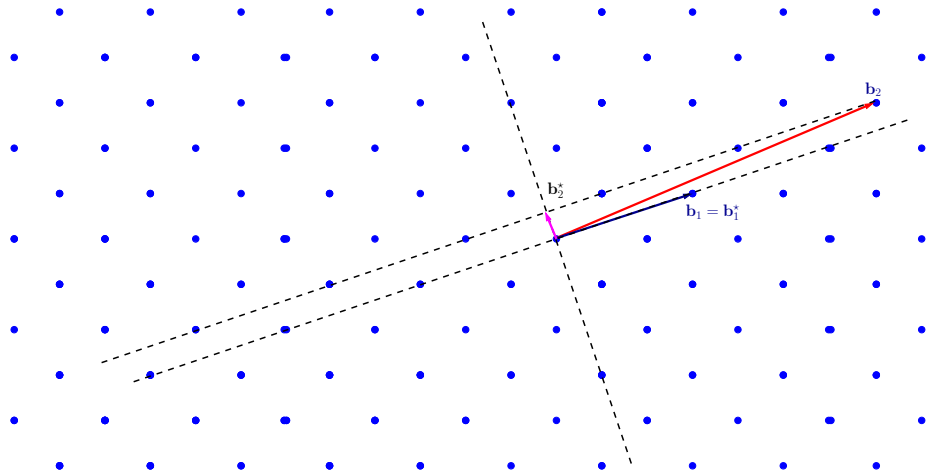
Principes de l'algorithme



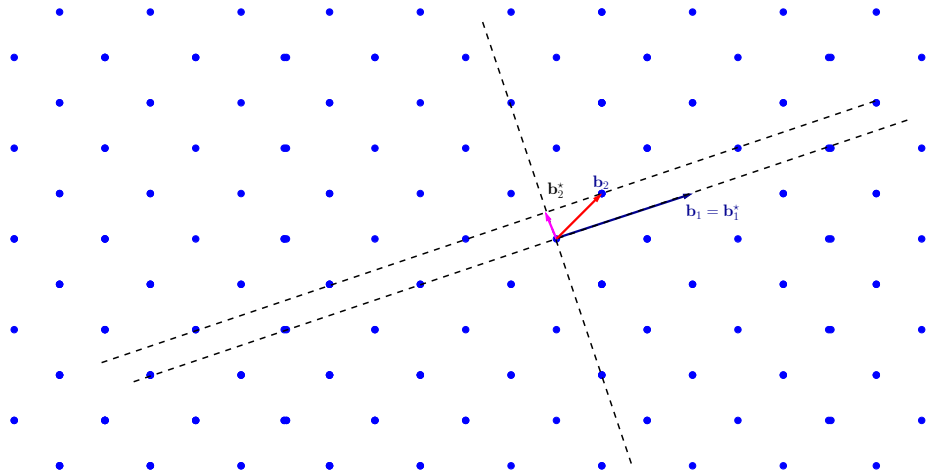
Principes de l'algorithme



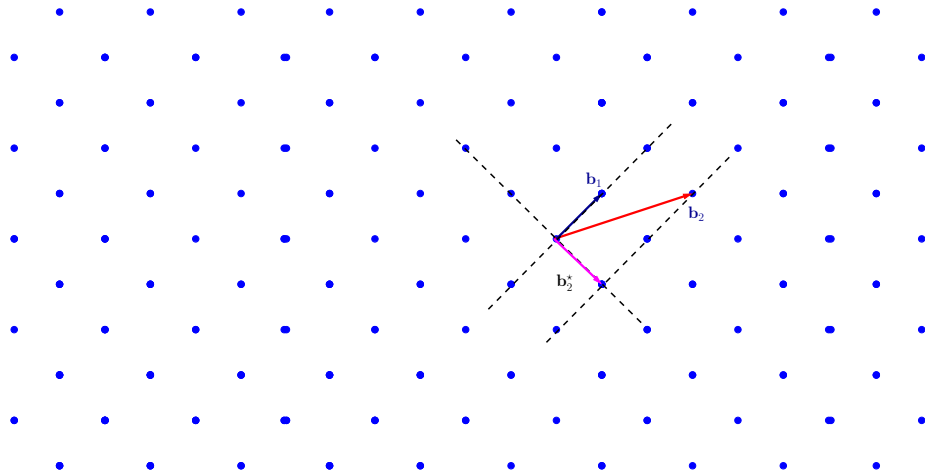
Principes de l'algorithme



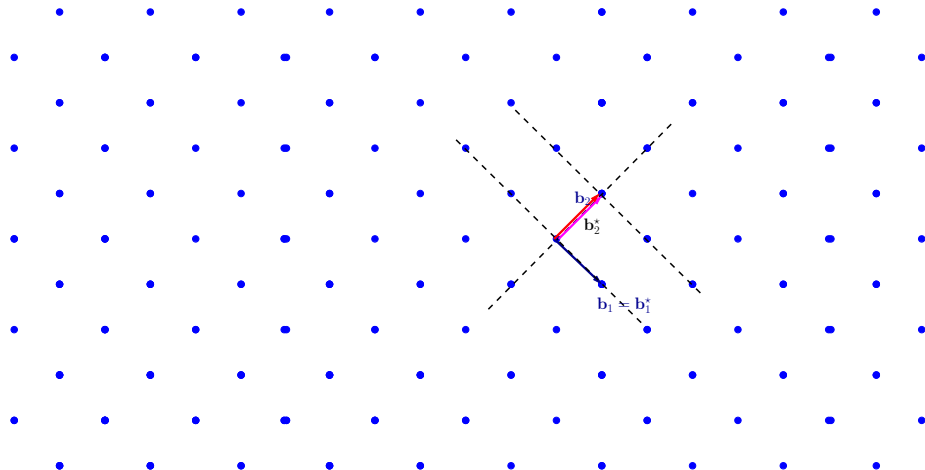
Principes de l'algorithme



Principes de l'algorithme



Principes de l'algorithme



$B^* = (\mathbf{b}_1^*, \dots, \mathbf{b}_d^*)$: la base des orthogonalisés

de Gram Schmidt

$\mathcal{P} : B \rightarrow B^*$ (la matrice de passage)

$$\begin{array}{c} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_i \\ \mathbf{b}_{i+1} \\ \vdots \\ \mathbf{b}_d \end{array} \begin{pmatrix} \mathbf{b}_1^* & \mathbf{b}_2^* & \dots & \mathbf{b}_i^* & \mathbf{b}_{i+1}^* & \dots & \mathbf{b}_d^* \\ 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ m_{2,1} & 1 & \ddots & & & & \vdots \\ \dots & \ddots & \ddots & & & & \vdots \\ \dots & \dots & \dots & 1 & 0 & & \vdots \\ \dots & \dots & \dots & m_{i+1,i} & 1 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & & \ddots & 0 \\ m_{d,1} & m_{d,2} & \dots & \dots & \dots & \dots & 1 \end{pmatrix}$$

$$B_i := \begin{pmatrix} \mathbf{u}_i \\ \mathbf{v}_i \end{pmatrix} \begin{pmatrix} \mathbf{b}_i^* & \mathbf{b}_{i+1}^* \\ 1 & 0 \\ m_{i+1,i} & 1 \end{pmatrix}.$$

Les **translations** :

- diminuent la norme des vecteurs
- rendent **la base propre**

Une base B est dite **propre** si la matrice de passage $\mathcal{P}$ vérifie :

$$|m_{i,j}| \leq 1/2 \text{ pour } 1 \leq j < i \leq n$$

Les **échanges** :

(t paramètre de l'algorithme, $t > 1$)

- effectués si $\|\mathbf{v}_i\| < (1/t)\|\mathbf{u}_i\|$
- rendent la base plus orthogonale

L'algorithme LLL = L'algorithme de t -Gauss pour les bases locales B_i

$B^* = (\mathbf{b}_1^*, \dots, \mathbf{b}_d^*)$: la base des orthogonalisés

de Gram Schmidt

$\mathcal{P} : B \rightarrow B^*$ (la matrice de passage)

$$\begin{matrix} & \mathbf{b}_1^* & \mathbf{b}_2^* & \dots & \mathbf{b}_i^* & \mathbf{b}_{i+1}^* & \dots & \mathbf{b}_d^* \\ \mathbf{b}_1 & \left(\begin{array}{ccccccc} 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ m_{2,1} & 1 & \ddots & & & & \vdots \\ \vdots & & \ddots & \ddots & & & \vdots \\ \dots & & & & & & \vdots \\ \dots & & & & 1 & 0 & \vdots \\ \dots & & & & m_{i+1,i} & 1 & \vdots \\ \vdots & & & & \vdots & & \ddots & 0 \\ m_{d,1} & m_{d,2} & \dots & \dots & \dots & \dots & \dots & 1 \end{array} \right) \end{matrix}$$

$$B_i := \begin{pmatrix} \mathbf{u}_i \\ \mathbf{v}_i \end{pmatrix} = \begin{pmatrix} \mathbf{b}_i^* & \mathbf{b}_{i+1}^* \\ 1 & 0 \\ m_{i+1,i} & 1 \end{pmatrix}.$$

Les **translations** :

- diminuent la norme des vecteurs
- rendent **la base propre**

Une base B est dite **propre** si la matrice de passage $\mathcal{P}$ vérifie :

$$|m_{i,j}| \leq 1/2 \text{ pour } 1 \leq j < i \leq n$$

Les **échanges** :

(t paramètre de l'algorithme, $t > 1$)

- effectués si $\|\mathbf{v}_i\| < (1/t)\|\mathbf{u}_i\|$
- rendent la base plus orthogonale

L'algorithme LLL = L'algorithme de t -Gauss pour les bases locales B_i

$B^* = (\mathbf{b}_1^*, \dots, \mathbf{b}_d^*)$: la base des orthogonalisés

de Gram Schmidt

$\mathcal{P} : B \rightarrow B^*$ (la matrice de passage)

$$\begin{matrix} & \mathbf{b}_1^* & \mathbf{b}_2^* & \dots & \mathbf{b}_i^* & \mathbf{b}_{i+1}^* & \dots & \mathbf{b}_d^* \\ \mathbf{b}_1 & \left(\begin{array}{ccccccc} 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ m_{2,1} & 1 & \ddots & & & & \vdots \\ \vdots & & \ddots & \ddots & & & \vdots \\ \dots & & & & & & \vdots \\ \dots & & & & 1 & 0 & \vdots \\ \dots & & & & m_{i+1,i} & 1 & \vdots \\ \vdots & & & & \vdots & & \ddots \\ m_{d,1} & m_{d,2} & \dots & \dots & \dots & \dots & 1 \end{array} \right) \end{matrix}$$

$$B_i := \begin{matrix} \mathbf{u}_i \\ \mathbf{v}_i \end{matrix} \begin{pmatrix} \mathbf{b}_i^* & \mathbf{b}_{i+1}^* \\ 1 & 0 \\ m_{i+1,i} & 1 \end{pmatrix}.$$

Les **translations** :

- diminuent la norme des vecteurs
- rendent **la base propre**

Une base B est dite **propre** si la matrice de passage $\mathcal{P}$ vérifie :

$$|m_{i,j}| \leq 1/2 \text{ pour } 1 \leq j < i \leq n$$

Les **échanges** :

(t paramètre de l'algorithme, $t > 1$)

- effectués si $\|\mathbf{v}_i\| < (1/t)\|\mathbf{u}_i\|$
- rendent la base plus orthogonale

L'algorithme LLL = L'algorithme de t -Gauss pour les bases locales B_i

$B^* = (\mathbf{b}_1^*, \dots, \mathbf{b}_d^*)$: la base des orthogonalisés

de Gram Schmidt

$\mathcal{P} : B \rightarrow B^*$ (la matrice de passage)

$$\begin{matrix} & \mathbf{b}_1^* & \mathbf{b}_2^* & \dots & \mathbf{b}_i^* & \mathbf{b}_{i+1}^* & \dots & \mathbf{b}_d^* \\ \mathbf{b}_1 & \left(\begin{array}{ccccccc} 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ m_{2,1} & 1 & \ddots & & & & \vdots \\ \vdots & & \ddots & \ddots & & & \vdots \\ \dots & & & & & & \vdots \\ \dots & & & & 1 & 0 & \vdots \\ \dots & & & & m_{i+1,i} & 1 & \vdots \\ \vdots & & & & \vdots & & \ddots \\ m_{d,1} & m_{d,2} & \dots & \dots & \dots & \dots & 1 \end{array} \right) \end{matrix}$$

$$B_i := \begin{matrix} \mathbf{u}_i \\ \mathbf{v}_i \end{matrix} \begin{pmatrix} \mathbf{b}_i^* & \mathbf{b}_{i+1}^* \\ 1 & 0 \\ m_{i+1,i} & 1 \end{pmatrix}.$$

Les **translations** :

- diminuent la norme des vecteurs
- rendent **la base propre**

Une base B est dite **propre** si la matrice de passage $\mathcal{P}$ vérifie :

$$|m_{i,j}| \leq 1/2 \text{ pour } 1 \leq j < i \leq n$$

Les **échanges** :

(t paramètre de l'algorithme, $t > 1$)

- effectués si $\|\mathbf{v}_i\| < (1/t)\|\mathbf{u}_i\|$
- rendent la base plus orthogonale

L'algorithme LLL = L'algorithme de t -Gauss pour les bases locales B_i

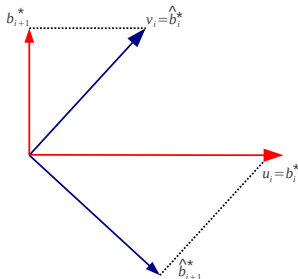
Les **échanges** rendent la base plus orthogonale

$$\ell_i := \|\mathbf{b}_i^*\|, \quad r_i := \frac{\ell_{i+1}}{\ell_i} \quad (\text{rapports de Siegel}) \quad \text{et} \quad \nu_i := \{m_{i+1,i}\} \quad (\text{partie fractionnelle centrée})$$

$$B_i = \begin{matrix} \mathbf{u}_i \\ \mathbf{v}_i \end{matrix} \begin{pmatrix} \mathbf{b}_i^* & \mathbf{b}_{i+1}^* \\ 1 & 0 \\ m_{i+1,i} & 1 \end{pmatrix}$$

Si la base B_i n'est pas réduite,
un échange modifie la base B_i .

Les nouveaux rapports de Siegel :



$$\tilde{r}_{i-1} = \rho \cdot r_{i-1}$$

$$\tilde{r}_i = \frac{1}{\rho^2} \cdot r_i$$

$$\tilde{r}_{i+1} = \rho \cdot r_{i+1}.$$

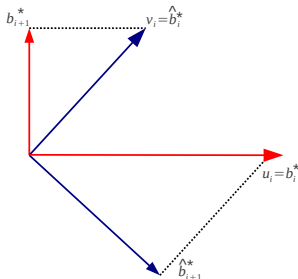
avec un **facteur de décroissance** ρ

$$\rho^2 = r_i^2 + \nu_i^2 \leq 1$$

Les **échanges** rendent la base plus orthogonale

$$\ell_i := \|\mathbf{b}_i^*\|, \quad r_i := \frac{\ell_{i+1}}{\ell_i} \quad (\text{rapports de Siegel}) \quad \text{et} \quad \nu_i := \{m_{i+1,i}\} \quad (\text{partie fractionnelle centrée})$$

$$B_i = \begin{matrix} \mathbf{u}_i \\ \mathbf{v}_i \end{matrix} \begin{pmatrix} \mathbf{b}_i^* & \mathbf{b}_{i+1}^* \\ 1 & 0 \\ m_{i+1,i} & 1 \end{pmatrix}$$



Si la base B_i n'est pas réduite,
un échange modifie la base B_i .

Les nouveaux rapports de Siegel :

$$\check{r}_{i-1} = \rho \cdot r_{i-1}$$

$$\check{r}_i = \frac{1}{\rho^2} \cdot r_i$$

$$\check{r}_{i+1} = \rho \cdot r_{i+1}.$$

avec un **facteur de décroissance** ρ

$$\rho^2 = r_i^2 + \nu_i^2 \leq 1$$

Algorithme LLL(t), $t > 1$ **Entrées:** $B = (\mathbf{b}_1, \dots, \mathbf{b}_d)$ **Résultat:** $\hat{B} = (\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_d)$ t -réduite.Calculer $(B^*, \mathcal{P})$; $i \leftarrow 1$;**tant que** $i < d$ **faire** **Translator** $\mathbf{b}_{i+1} // \mathbf{b}_i$ (t.q. $|m_{i+1,i}| \leq 1/2$); Tester $\|\mathbf{v}_i\| > (1/t) \|\mathbf{u}_i\|$? **Si oui** **Translator** $\mathbf{b}_{i+1} // \mathbf{b}_k$ (t.q. $|m_{k,i}| \leq 1/2$); $i \leftarrow i + 1$; **Sinon** **Échanger** $\mathbf{b}_i$ et $\mathbf{b}_{i+1}$; Recalculer $(B^*, \mathcal{P})$; Si $i \neq 1$ alors $i \leftarrow i - 1$;**fin**

Condition de Lovász

$$\mathcal{L}(t) : \quad r_i^2 \geq \frac{1}{t^2} - \nu_i^2$$

$$\|\mathbf{v}_i\| \geq (1/t) \|\mathbf{u}_i\|$$

Condition de Siegel

$$\mathcal{S}(s) : \quad r_i \geq \frac{1}{s}$$

Une base B d'un réseau $\mathcal{L}$ qui est propre est dite **réduite** au sens de t -Lovász (resp. s -Siegel) si vérifie la condition $\mathcal{L}(t)$ (resp. $\mathcal{S}(s)$)

si s et t vérifient

$$\frac{1}{s^2} = \frac{1}{t^2} - \frac{1}{4}, \text{ avec } s \geq \frac{2}{\sqrt{3}} \text{ et } t \geq 1$$

alors $\mathcal{L}(t) \Rightarrow \mathcal{S}(s)$

Algorithmme LLL(t), $t > 1$

Entrées: $B = (\mathbf{b}_1, \dots, \mathbf{b}_d)$

Résultat: $\hat{B} = (\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_d)$ t -réduite.

Calculer $(B^*, \mathcal{P})$;

$i \leftarrow 1$;

tant que $i < d$ **faire**

Translator $\mathbf{b}_{i+1} // \mathbf{b}_i$

(t.q. $|m_{i+1,i}| \leq 1/2$);

Tester $\|\mathbf{v}_i\| > (1/t) \|\mathbf{u}_i\|$?

Si oui

Translator $\mathbf{b}_{i+1} // \mathbf{b}_k$

(t.q. $|m_{k,i}| \leq 1/2$);

$i \leftarrow i + 1$;

Sinon

Échanger $\mathbf{b}_i$ et $\mathbf{b}_{i+1}$;

Recalculer $(B^*, \mathcal{P})$;

Si $i \neq 1$ alors $i \leftarrow i - 1$;

fin

Condition de Lovász

$$\mathcal{L}(t) : \quad r_i^2 \geq \frac{1}{t^2} - \nu_i^2$$

$$\|\mathbf{v}_i\| \geq (1/t) \|\mathbf{u}_i\|$$

Condition de Siegel

$$\mathcal{S}(s) : \quad r_i \geq \frac{1}{s}$$

Une base B d'un réseau $\mathcal{L}$ qui est propre est dite **réduite** au sens de t -Lovász (resp. s -Siegel) si vérifie la condition $\mathcal{L}(t)$ (resp. $\mathcal{S}(s)$)

si s et t vérifient

$$\frac{1}{s^2} = \frac{1}{t^2} - \frac{1}{4}, \text{ avec } s \geq \frac{2}{\sqrt{3}} \text{ et } t \geq 1$$

alors $\mathcal{L}(t) \Rightarrow \mathcal{S}(s)$

Potentiel d'une base : $\Delta(B) = \prod_{i=1}^d \ell_i^{(d-i)}$.

À chaque échange $\Delta(B)$ décroît d'un facteur ρ ($\rho^{(j)}$: ρ à l'étape j).

$$\rho \leq \frac{1}{t} = \left(\frac{1}{s^2} + \frac{1}{4} \right)^{1/2} \leq 1$$

Le nombre d'échanges :

$$K(B) = \frac{1}{|\log \bar{\rho}|} \log \frac{\Delta(B)}{\Delta(\hat{B})}, \quad \text{avec} \quad \log \bar{\rho} = \frac{1}{K} \sum_{j=1}^K \log \rho^{(j)}$$

Pire des cas (uniquement $t > 1$) :

$$K(B) \leq \frac{1}{\log t} \log \frac{\Delta(B)}{\Delta(\hat{B})},$$

Remarque :

- Le pire des cas peut être rarement atteint
- L'analyse probabiliste rend mieux compte du comportement "générique"

Potentiel d'une base : $\Delta(B) = \prod_{i=1}^d \ell_i^{(d-i)}$.

À chaque échange $\Delta(B)$ décroît d'un facteur ρ ($\rho^{(j)}$: ρ à l'étape j).

$$\rho \leq \frac{1}{t} = \left(\frac{1}{s^2} + \frac{1}{4} \right)^{1/2} \leq 1$$

Le nombre d'échanges :

$$K(B) = \frac{1}{|\log \bar{\rho}|} \log \frac{\Delta(B)}{\Delta(\widehat{B})}, \quad \text{avec} \quad \log \bar{\rho} = \frac{1}{K} \sum_{j=1}^K \log \rho^{(j)}$$

Pire des cas (uniquement $t > 1$) :

$$K(B) \leq \frac{1}{\log t} \log \frac{\Delta(B)}{\Delta(\widehat{B})},$$

Remarque :

- Le pire des cas peut être rarement atteint
- L'analyse probabiliste rend mieux compte du comportement "générique"

Potentiel d'une base : $\Delta(B) = \prod_{i=1}^d \ell_i^{(d-i)}$.

À chaque échange $\Delta(B)$ décroît d'un facteur ρ ($\rho^{(j)}$: ρ à l'étape j).

$$\rho \leq \frac{1}{t} = \left(\frac{1}{s^2} + \frac{1}{4} \right)^{1/2} \leq 1$$

Le nombre d'échanges :

$$K(B) = \frac{1}{|\log \bar{\rho}|} \log \frac{\Delta(B)}{\Delta(\widehat{B})}, \quad \text{avec} \quad \log \bar{\rho} = \frac{1}{K} \sum_{j=1}^K \log \rho^{(j)}$$

Pire des cas (uniquement $t > 1$) :

$$K(B) \leq \frac{1}{\log t} \log \frac{\Delta(B)}{\Delta(\widehat{B})},$$

Remarque :

- Le pire des cas peut être rarement atteint
- L'analyse probabiliste rend mieux compte du comportement "générique"

Potentiel d'une base : $\Delta(B) = \prod_{i=1}^d \ell_i^{(d-i)}$.

À chaque échange $\Delta(B)$ décroît d'un facteur ρ ($\rho^{(j)}$: ρ à l'étape j).

$$\rho \leq \frac{1}{t} = \left(\frac{1}{s^2} + \frac{1}{4} \right)^{1/2} \leq 1$$

Le nombre d'échanges :

$$K(B) = \frac{1}{|\log \bar{\rho}|} \log \frac{\Delta(B)}{\Delta(\widehat{B})}, \quad \text{avec} \quad \log \bar{\rho} = \frac{1}{K} \sum_{j=1}^K \log \rho^{(j)}$$

Pire des cas (uniquement $t > 1$) :

$$K(B) \leq \frac{1}{\log t} \log \frac{\Delta(B)}{\Delta(\widehat{B})},$$

Remarque :

- Le pire des cas peut être rarement atteint
- L'analyse probabiliste rend mieux compte du comportement "générique"

Plan

- 1 Algorithme LLL
 - Réduction des réseaux
 - Exemple d'application
 - Algorithme LLL
- 2 Modélisations de l'algorithme LLL et de ses entrées
 - Modèles d'exécution (part 1)
 - Modèles d'entrée
- 3 Résultats obtenus dans les différents modèles
 - Modèle M1 : cfg/tas de sable
 - Modèle M1 et modèles d'entrée
 - Modèle M2 : système dynamique de $\mathbb{R}^d$
 - Modèle M3 : systèmes dynamiques probabilistes de $\mathbb{R}^d$
- 4 Conclusion

Approches simplifiées

Modélisation simplifiée de l'exécution :

- Une vue simplifiée de la réduction inspirée des tas de sable
- Différents degrés de simplification pour la modélisation allant du tas de sable au véritable algorithme LLL

Modélisation des principales entrées cryptographiques :

- "tas plein" : Ajtai
- "uni-tas" : NTRU, Sac-à-dos, Schnorr...
- "tas à trous" : Coppersmith

Approches simplifiées

Modélisation simplifiée de l'exécution :

- Une vue simplifiée de la réduction inspirée des tas de sable
- Différents degrés de simplification pour la modélisation allant du tas de sable au véritable algorithme LLL

Modélisation des principales entrées cryptographiques :

- "tas plein" : Ajtai
- "uni-tas" : NTRU, Sac-à-dos, Schnorr...
- "tas à trous" : Coppersmith

Approches simplifiées

Modélisation simplifiée de l'exécution :

- Une vue simplifiée de la réduction inspirée des tas de sable
- Différents degrés de simplification pour la modélisation allant du tas de sable au véritable algorithme LLL

Modélisation des principales entrées cryptographiques :

- "tas plein" : Ajtai
- "uni-tas" : NTRU, Sac-à-dos, Schnorr...
- "tas à trous" : Coppersmith

1 Algorithme LLL

- Réduction des réseaux
- Exemple d'application
- Algorithme LLL

2 Modélisations de l'algorithme LLL et de ses entrées

- Modèles d'exécution (part 1)
- Modèles d'entrée

3 Résultats obtenus dans les différents modèles

- Modèle M1 : cfg/tas de sable
- Modèle M1 et modèles d'entrée
- Modèle M2 : système dynamique de $\mathbb{R}^d$
- Modèle M3 : systèmes dynamiques probabilistes de $\mathbb{R}^d$

4 Conclusion

Notre approche :

le rôle essentiel est joué par les rapports

$$r_i = \frac{\ell_{i+1}}{\ell_i}$$

les coefficients ν_i ont un rôle secondaire

si $r_i < 1/s$ alors

$$\tilde{T}_\rho^{(i)}(\mathbf{r}) = \begin{cases} \check{r}_{i-1} := \rho \cdot r_{i-1}; \\ \check{r}_i := \frac{1}{\rho^2} \cdot r_i; \\ \check{r}_{i+1} := \rho \cdot r_{i+1}; \end{cases}$$

Point de vue additif :

$$\mathbf{c} := (c_1, \dots, c_{d-1})$$

avec $c_i = -\log_s r_i$ et $\alpha = -\log_s \rho$

si $c_i > 1$ alors

$$T_\alpha^{(i)}(\mathbf{c}) = \begin{cases} \check{c}_{i-1} := c_{i-1} + \alpha \\ \check{c}_i := c_i - 2\alpha \\ \check{c}_{i+1} := c_{i+1} + \alpha \end{cases}$$

Notre approche :

le rôle essentiel est joué par les rapports

$$r_i = \frac{\ell_{i+1}}{\ell_i}$$

les coefficients ν_i ont un rôle secondaire

Point de vue additif :

$$\mathbf{c} := (c_1, \dots, c_{d-1})$$

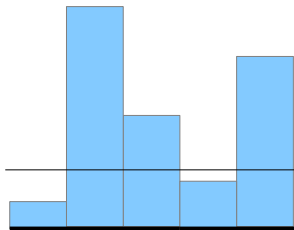
avec $c_i = -\log_s r_i$ et $\alpha = -\log_s \rho$

si $r_i < 1/s$ alors

$$\tilde{T}_\rho^{(i)}(\mathbf{r}) = \begin{cases} \check{r}_{i-1} := \rho \cdot r_{i-1}; \\ \check{r}_i := \frac{1}{\rho^2} \cdot r_i; \\ \check{r}_{i+1} := \rho \cdot r_{i+1}; \end{cases}$$

si $c_i > 1$ alors

$$T_\alpha^{(i)}(\mathbf{c}) = \begin{cases} \check{c}_{i-1} := c_{i-1} + \alpha \\ \check{c}_i := c_i - 2\alpha \\ \check{c}_{i+1} := c_{i+1} + \alpha \end{cases}$$



Notre approche :

le rôle essentiel est joué par les rapports

$$r_i = \frac{\ell_{i+1}}{\ell_i}$$

les coefficients ν_i ont un rôle secondaire

Point de vue additif :

$$\mathbf{c} := (c_1, \dots, c_{d-1})$$

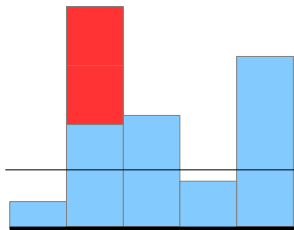
avec $c_i = -\log_s r_i$ et $\alpha = -\log_s \rho$

si $r_i < 1/s$ alors

$$\tilde{T}_\rho^{(i)}(\mathbf{r}) = \begin{cases} \check{r}_{i-1} := \rho \cdot r_{i-1}; \\ \check{r}_i := \frac{1}{\rho^2} \cdot r_i; \\ \check{r}_{i+1} := \rho \cdot r_{i+1}; \end{cases}$$

si $c_i > 1$ alors

$$T_\alpha^{(i)}(\mathbf{c}) = \begin{cases} \check{c}_{i-1} := c_{i-1} + \alpha \\ \check{c}_i := c_i - 2\alpha \\ \check{c}_{i+1} := c_{i+1} + \alpha \end{cases}$$



Notre approche :

le rôle essentiel est joué par les rapports

$$r_i = \frac{\ell_{i+1}}{\ell_i}$$

les coefficients ν_i ont un rôle secondaire

Point de vue additif :

$$\mathbf{c} := (c_1, \dots, c_{d-1})$$

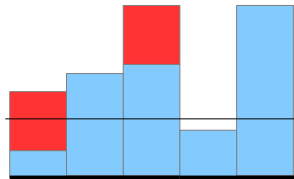
avec $c_i = -\log_s r_i$ et $\alpha = -\log_s \rho$

si $r_i < 1/s$ alors

$$\tilde{T}_\rho^{(i)}(\mathbf{r}) = \begin{cases} \check{r}_{i-1} := \rho \cdot r_{i-1}; \\ \check{r}_i := \frac{1}{\rho^2} \cdot r_i; \\ \check{r}_{i+1} := \rho \cdot r_{i+1}; \end{cases}$$

si $c_i > 1$ alors

$$T_\alpha^{(i)}(\mathbf{c}) = \begin{cases} \check{c}_{i-1} := c_{i-1} + \alpha \\ \check{c}_i := c_i - 2\alpha \\ \check{c}_{i+1} := c_{i+1} + \alpha \end{cases}$$



Notre approche :

le rôle essentiel est joué par les rapports

$$r_i = \frac{\ell_{i+1}}{\ell_i}$$

les coefficients ν_i ont un rôle secondaire

Point de vue additif :

$$\mathbf{c} := (c_1, \dots, c_{d-1})$$

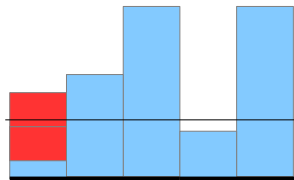
avec $c_i = -\log_s r_i$ et $\alpha = -\log_s \rho$

si $r_i < 1/s$ alors

$$\tilde{T}_\rho^{(i)}(\mathbf{r}) = \begin{cases} \check{r}_{i-1} := \rho \cdot r_{i-1}; \\ \check{r}_i := \frac{1}{\rho^2} \cdot r_i; \\ \check{r}_{i+1} := \rho \cdot r_{i+1}; \end{cases}$$

si $c_i > 1$ alors

$$T_\alpha^{(i)}(\mathbf{c}) = \begin{cases} \check{c}_{i-1} := c_{i-1} + \alpha \\ \check{c}_i := c_i - 2\alpha \\ \check{c}_{i+1} := c_{i+1} + \alpha \end{cases}$$



Notre approche :

le rôle essentiel est joué par les rapports

$$r_i = \frac{\ell_{i+1}}{\ell_i}$$

les coefficients ν_i ont un rôle secondaire

Point de vue additif :

$$\mathbf{c} := (c_1, \dots, c_{d-1})$$

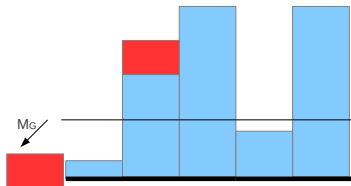
avec $c_i = -\log_s r_i$ et $\alpha = -\log_s \rho$

si $r_i < 1/s$ alors

$$\tilde{T}_\rho^{(i)}(\mathbf{r}) = \begin{cases} \tilde{r}_{i-1} := \rho \cdot r_{i-1}; \\ \tilde{r}_i := \frac{1}{\rho^2} \cdot r_i; \\ \tilde{r}_{i+1} := \rho \cdot r_{i+1}; \end{cases}$$

si $c_i > 1$ alors

$$T_\alpha^{(i)}(\mathbf{c}) = \begin{cases} \check{c}_{i-1} := c_{i-1} + \alpha \\ \check{c}_i := c_i - 2\alpha \\ \check{c}_{i+1} := c_{i+1} + \alpha \end{cases}$$



Notre approche :

le rôle essentiel est joué par les rapports

$$r_i = \frac{\ell_{i+1}}{\ell_i}$$

les coefficients ν_i ont un rôle secondaire

Point de vue additif :

$$\mathbf{c} := (c_1, \dots, c_{d-1})$$

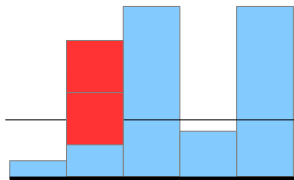
avec $c_i = -\log_s r_i$ et $\alpha = -\log_s \rho$

si $r_i < 1/s$ alors

$$\tilde{T}_\rho^{(i)}(\mathbf{r}) = \begin{cases} \check{r}_{i-1} := \rho \cdot r_{i-1}; \\ \check{r}_i := \frac{1}{\rho^2} \cdot r_i; \\ \check{r}_{i+1} := \rho \cdot r_{i+1}; \end{cases}$$

si $c_i > 1$ alors

$$T_\alpha^{(i)}(\mathbf{c}) = \begin{cases} \check{c}_{i-1} := c_{i-1} + \alpha \\ \check{c}_i := c_i - 2\alpha \\ \check{c}_{i+1} := c_{i+1} + \alpha \end{cases}$$



Notre approche :

le rôle essentiel est joué par les rapports

$$r_i = \frac{\ell_{i+1}}{\ell_i}$$

les coefficients ν_i ont un rôle secondaire

Point de vue additif :

$$\mathbf{c} := (c_1, \dots, c_{d-1})$$

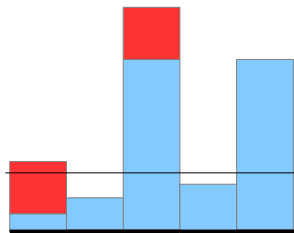
avec $c_i = -\log_s r_i$ et $\alpha = -\log_s \rho$

si $r_i < 1/s$ alors

$$\tilde{T}_\rho^{(i)}(\mathbf{r}) = \begin{cases} \check{r}_{i-1} := \rho \cdot r_{i-1}; \\ \check{r}_i := \frac{1}{\rho^2} \cdot r_i; \\ \check{r}_{i+1} := \rho \cdot r_{i+1}; \end{cases}$$

si $c_i > 1$ alors

$$T_\alpha^{(i)}(\mathbf{c}) = \begin{cases} \check{c}_{i-1} := c_{i-1} + \alpha \\ \check{c}_i := c_i - 2\alpha \\ \check{c}_{i+1} := c_{i+1} + \alpha \end{cases}$$



Notre approche :

le rôle essentiel est joué par les rapports

$$r_i = \frac{\ell_{i+1}}{\ell_i}$$

les coefficients ν_i ont un rôle secondaire

Point de vue additif :

$$\mathbf{c} := (c_1, \dots, c_{d-1})$$

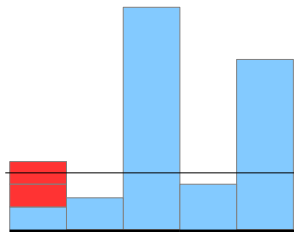
avec $c_i = -\log_s r_i$ et $\alpha = -\log_s \rho$

si $r_i < 1/s$ alors

$$\tilde{T}_\rho^{(i)}(\mathbf{r}) = \begin{cases} \check{r}_{i-1} := \rho \cdot r_{i-1}; \\ \check{r}_i := \frac{1}{\rho^2} \cdot r_i; \\ \check{r}_{i+1} := \rho \cdot r_{i+1}; \end{cases}$$

si $c_i > 1$ alors

$$T_\alpha^{(i)}(\mathbf{c}) = \begin{cases} \check{c}_{i-1} := c_{i-1} + \alpha \\ \check{c}_i := c_i - 2\alpha \\ \check{c}_{i+1} := c_{i+1} + \alpha \end{cases}$$



Notre approche :

le rôle essentiel est joué par les rapports

$$r_i = \frac{\ell_{i+1}}{\ell_i}$$

les coefficients ν_i ont un rôle secondaire

Point de vue additif :

$$\mathbf{c} := (c_1, \dots, c_{d-1})$$

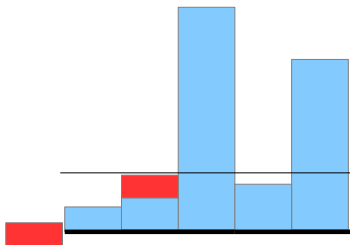
avec $c_i = -\log_s r_i$ et $\alpha = -\log_s \rho$

si $r_i < 1/s$ alors

$$\tilde{T}_\rho^{(i)}(\mathbf{r}) = \begin{cases} \check{r}_{i-1} := \rho \cdot r_{i-1}; \\ \check{r}_i := \frac{1}{\rho^2} \cdot r_i; \\ \check{r}_{i+1} := \rho \cdot r_{i+1}; \end{cases}$$

si $c_i > 1$ alors

$$T_\alpha^{(i)}(\mathbf{c}) = \begin{cases} \check{c}_{i-1} := c_{i-1} + \alpha \\ \check{c}_i := c_i - 2\alpha \\ \check{c}_{i+1} := c_{i+1} + \alpha \end{cases}$$



Notre approche :

le rôle essentiel est joué par les rapports

$$r_i = \frac{\ell_{i+1}}{\ell_i}$$

les coefficients ν_i ont un rôle secondaire

Point de vue additif :

$$\mathbf{c} := (c_1, \dots, c_{d-1})$$

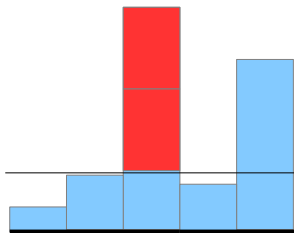
avec $c_i = -\log_s r_i$ et $\alpha = -\log_s \rho$

si $r_i < 1/s$ alors

$$\tilde{T}_\rho^{(i)}(\mathbf{r}) = \begin{cases} \check{r}_{i-1} := \rho \cdot r_{i-1}; \\ \check{r}_i := \frac{1}{\rho^2} \cdot r_i; \\ \check{r}_{i+1} := \rho \cdot r_{i+1}; \end{cases}$$

si $c_i > 1$ alors

$$T_\alpha^{(i)}(\mathbf{c}) = \begin{cases} \check{c}_{i-1} := c_{i-1} + \alpha \\ \check{c}_i := c_i - 2\alpha \\ \check{c}_{i+1} := c_{i+1} + \alpha \end{cases}$$



Notre approche :

le rôle essentiel est joué par les rapports

$$r_i = \frac{\ell_{i+1}}{\ell_i}$$

les coefficients ν_i ont un rôle secondaire

Point de vue additif :

$$\mathbf{c} := (c_1, \dots, c_{d-1})$$

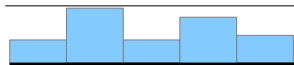
avec $c_i = -\log_s r_i$ et $\alpha = -\log_s \rho$

si $r_i < 1/s$ alors

$$\tilde{T}_\rho^{(i)}(\mathbf{r}) = \begin{cases} \check{r}_{i-1} := \rho \cdot r_{i-1}; \\ \check{r}_i := \frac{1}{\rho^2} \cdot r_i; \\ \check{r}_{i+1} := \rho \cdot r_{i+1}; \end{cases}$$

si $c_i > 1$ alors

$$T_\alpha^{(i)}(\mathbf{c}) = \begin{cases} \check{c}_{i-1} := c_{i-1} + \alpha \\ \check{c}_i := c_i - 2\alpha \\ \check{c}_{i+1} := c_{i+1} + \alpha \end{cases}$$



Entrées: $\mathbf{c} \in \mathcal{J}$

Sorties: $\mathbf{c} \in \mathcal{O} = \prod \mathcal{O}_i$

tant que $\mathbf{c} \notin \mathcal{O}$ **faire**

 Choisir i tel que $c_i \notin \mathcal{O}_i$;

 Choisir $\alpha \in \mathbb{R}^+$;

$\mathbf{c} \leftarrow T_\alpha^{(i)}(\mathbf{c})$;

fin

Renvoyer $\mathbf{c}$

Deux principaux choix :

Choisir i : la stratégie

classique, gloutonne, aléatoire...

Choisir α : calculé, fixé...

$$\alpha_i = -\frac{1}{2} \log_s(s^{-2c_i} + \nu_i^2)$$

M1 : $\alpha_i = \alpha$ est fixé au cours de l'algorithme

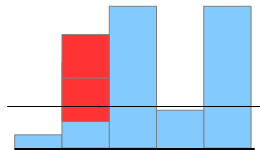
M2 : α_i est fonction de c_i , ν_i fixé dans $[-\frac{1}{2}, \frac{1}{2}]$

M3 : α_i est fonction de c_i , $\nu_i \in \mathcal{R} [-\frac{1}{2}, \frac{1}{2}]$

M4 : LLL avec la condition de Siegel ($c_i \leq 1$)

M5 : vrai LLL avec la condition de Lovász

$$c_i \leq \frac{1}{2} \log_s \left(\frac{1}{i^2} + \nu_i^2 \right)$$



Entrées: $\mathbf{c} \in \mathcal{J}$

Sorties: $\mathbf{c} \in \mathcal{O} = \prod \mathcal{O}_i$

tant que $\mathbf{c} \notin \mathcal{O}$ **faire**

 Choisir i tel que $c_i \notin \mathcal{O}_i$;

 Choisir $\alpha \in \mathbb{R}^+$;

$\mathbf{c} \leftarrow T_{\alpha}^{(i)}(\mathbf{c})$;

fin

Renvoyer $\mathbf{c}$

Deux principaux choix :

Choisir i : la stratégie

classique, gloutonne, aléatoire...

Choisir α : calculé, fixé...

$$\alpha_i = -\frac{1}{2} \log_s (s^{-2c_i} + \nu_i^2)$$

M1 : $\alpha_i = \alpha$ est fixé au cours de l'algorithme

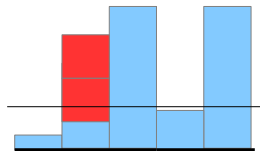
M2 : α_i est fonction de c_i , ν_i fixé dans $[-\frac{1}{2}, \frac{1}{2}]$

M3 : α_i est fonction de c_i , $\nu_i \in \mathcal{R}$ $[-\frac{1}{2}, \frac{1}{2}]$

M4 : LLL avec la condition de Siegel ($c_i \leq 1$)

M5 : vrai LLL avec la condition de Lovász

$$c_i \leq \frac{1}{2} \log_s \left(\frac{1}{t^2} + \nu_i^2 \right)$$



Entrées: $\mathbf{c} \in \mathcal{J}$

Sorties: $\mathbf{c} \in \mathcal{O} = \prod \mathcal{O}_i$

tant que $\mathbf{c} \notin \mathcal{O}$ **faire**

 Choisir i tel que $c_i \notin \mathcal{O}_i$;

 Choisir $\alpha \in \mathbb{R}^+$;

$\mathbf{c} \leftarrow T_{\alpha}^{(i)}(\mathbf{c})$;

fin

Renvoyer $\mathbf{c}$

Deux principaux choix :

Choisir i : la stratégie

classique, gloutonne, aléatoire...

Choisir α : calculé, fixé...

$$\alpha_i = -\frac{1}{2} \log_s(s^{-2c_i} + \nu_i^2)$$

M1 : $\alpha_i = \alpha$ est fixé au cours de l'algorithme

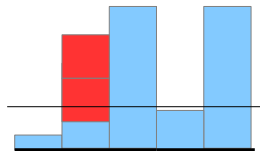
M2 : α_i est fonction de c_i , ν_i fixé dans $[-\frac{1}{2}, \frac{1}{2}]$

M3 : α_i est fonction de c_i , $\nu_i \in \mathcal{R}$ $[-\frac{1}{2}, \frac{1}{2}]$

M4 : LLL avec la condition de Siegel ($c_i \leq 1$)

M5 : vrai LLL avec la condition de Lovász

$$c_i \leq \frac{1}{2} \log_s \left(\frac{1}{t^2} + \nu_i^2 \right)$$



Entrées: $\mathbf{c} \in \mathcal{J}$

Sorties: $\mathbf{c} \in \mathcal{O} = \prod \mathcal{O}_i$

tant que $\mathbf{c} \notin \mathcal{O}$ **faire**

 Choisir i tel que $c_i \notin \mathcal{O}_i$;

 Choisir $\alpha \in \mathbb{R}^+$;

$\mathbf{c} \leftarrow T_{\alpha}^{(i)}(\mathbf{c})$;

fin

Renvoyer $\mathbf{c}$

Deux principaux choix :

Choisir i : la stratégie

classique, gloutonne, aléatoire...

Choisir α : calculé, fixé...

$$\alpha_i = -\frac{1}{2} \log_s(s^{-2c_i} + \nu_i^2)$$

M1 : $\alpha_i = \alpha$ est fixé au cours de l'algorithme

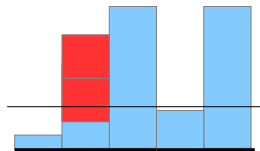
M2 : α_i est fonction de c_i , ν_i fixé dans $[-\frac{1}{2}, \frac{1}{2}]$

M3 : α_i est fonction de c_i , $\nu_i \in \mathcal{R}$ $[-\frac{1}{2}, \frac{1}{2}]$

M4 : LLL avec la condition de Siegel ($c_i \leq 1$)

M5 : vrai LLL avec la condition de Lovász

$$c_i \leq \frac{1}{2} \log_s \left(\frac{1}{t^2} + \nu_i^2 \right)$$



Entrées: $\mathbf{c} \in \mathcal{J}$

Sorties: $\mathbf{c} \in \mathcal{O} = \prod \mathcal{O}_i$

tant que $\mathbf{c} \notin \mathcal{O}$ **faire**

 Choisir i tel que $c_i \notin \mathcal{O}_i$;

 Choisir $\alpha \in \mathbb{R}^+$;

$\mathbf{c} \leftarrow T_{\alpha}^{(i)}(\mathbf{c})$;

fin

Renvoyer $\mathbf{c}$

Deux principaux choix :

Choisir i : la stratégie

classique, gloutonne, aléatoire...

Choisir α : calculé, fixé...

$$\alpha_i = -\frac{1}{2} \log_s(s^{-2c_i} + \nu_i^2)$$

M1 : $\alpha_i = \alpha$ est fixé au cours de l'algorithme

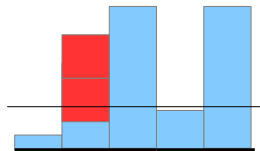
M2 : α_i est fonction de c_i , ν_i fixé dans $[-\frac{1}{2}, \frac{1}{2}]$

M3 : α_i est fonction de c_i , $\nu_i \in \mathcal{R}$ $[-\frac{1}{2}, \frac{1}{2}]$

M4 : LLL avec la condition de Siegel ($c_i \leq 1$)

M5 : vrai LLL avec la condition de Lovász

$$c_i \leq \frac{1}{2} \log_s \left(\frac{1}{t^2} + \nu_i^2 \right)$$



Entrées: $\mathbf{c} \in \mathcal{J}$

Sorties: $\mathbf{c} \in \mathcal{O} = \prod \mathcal{O}_i$

tant que $\mathbf{c} \notin \mathcal{O}$ **faire**

 Choisir i tel que $c_i \notin \mathcal{O}_i$;

 Choisir $\alpha \in \mathbb{R}^+$;

$\mathbf{c} \leftarrow T_{\alpha}^{(i)}(\mathbf{c})$;

fin

Renvoyer $\mathbf{c}$

Deux principaux choix :

Choisir i : la stratégie

classique, gloutonne, aléatoire...

Choisir α : calculé, fixé...

$$\alpha_i = -\frac{1}{2} \log_s(s^{-2c_i} + \nu_i^2)$$

M1 : $\alpha_i = \alpha$ est fixé au cours de l'algorithme

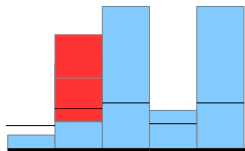
M2 : α_i est fonction de c_i , ν_i fixé dans $[-\frac{1}{2}, \frac{1}{2}]$

M3 : α_i est fonction de c_i , $\nu_i \in \mathcal{R}$ $[-\frac{1}{2}, \frac{1}{2}]$

M4 : LLL avec la condition de Siegel ($c_i \leq 1$)

M5 : vrai LLL avec la condition de Lovász

$$c_i \leq \frac{1}{2} \log_s \left(\frac{1}{t^2} + \nu_i^2 \right)$$



1 Algorithme LLL

- Réduction des réseaux
- Exemple d'application
- Algorithme LLL

2 Modélisations de l'algorithme LLL et de ses entrées

- Modèles d'exécution (part 1)
- Modèles d'entrée

3 Résultats obtenus dans les différents modèles

- Modèle M1 : cfg/tas de sable
- Modèle M1 et modèles d'entrée
- Modèle M2 : système dynamique de $\mathbb{R}^d$
- Modèle M3 : systèmes dynamiques probabilistes de $\mathbb{R}^d$

4 Conclusion

Approche générale

$$B = \begin{pmatrix} \ell_1 & 0 & \cdots & 0 \\ * & \ell_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ * & \cdots & * & \ell_d \end{pmatrix}.$$

La longueur des orthogonalisés se lit sur la diagonale

Trois paramètres : (Υ, d, g)

- d : dimension du réseau
- densité g strictement positive définie sur $[0, 1]$ qui vérifie donc $\int_0^1 g(t)dt = 1$
- Υ : masse moyenne totale du cfg

$$\mathbb{E}[c_i] = \frac{1}{d-1} \Upsilon g\left(\frac{i}{d}\right) \quad \text{et} \quad \mathbb{E}[\mathcal{M}] \approx \Upsilon$$

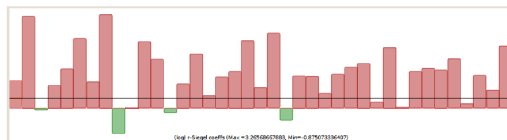
L'énergie moyenne vérifie

$$\mathbb{E}[\mathcal{E}] \sim d^2 \mathbb{E}[\mathcal{M}] \int_0^1 x(1-x)g(x)dx.$$

Les entrées "grand tas" (Ajtai)

Applications

- Preuve de la connection entre le pire des cas/le cas moyen (Ajtai '96)
- Les bases d'Ajtai modélisent des instances difficiles en moyenne, vis-à-vis du problème SVP



$\mathcal{A}(\Upsilon, d, g)$: des *cfg* défini par une densité g strictement positive définie sur $[0, 1]$ et de classe $\mathcal{C}^1$ qui vérifie donc $\int_0^1 g(t)dt = 1$.

$$\mathbb{E}[c_i] = \frac{1}{d-1} \Upsilon g\left(\frac{i}{d}\right) \quad \text{et} \quad \mathbb{E}[\mathcal{M}] \approx \Upsilon$$

Réseaux uni-tas

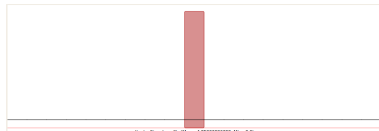
Applications

- Cryptanalyse des protocoles de type **sac à dos**
- Factorisation des entiers par la méthode de **Schnorr**
- Cryptosystème **NTRU**

$$\mathcal{B} = \left(\begin{array}{c|c} X \cdot I_i & 0 \\ \hline A & Y \cdot I_{d-i} \end{array} \right).$$

(i) $d \in \mathbb{N}$, un réel $\beta \in [0, 1]$ et l'entier $i \in [1 \dots d-1]$ est défini par $i := [\beta d]$,

(iii) X et Y str. positives, avec $X \gg Y$,



$$\mathcal{U}(\Upsilon, d, \beta)$$

$$\mathbb{E}[c_i] = \Upsilon \quad \text{et} \quad \mathbb{E}[c_j] = 0 \quad j \neq i$$

L'énergie vérifie, pour $d \rightarrow \infty$,

$$\mathbb{E}[\mathcal{E}] = \sum_{i=1}^{d-1} i(d-i) \mathbb{E}[c_i] \sim \begin{cases} d^2 \beta(1-\beta) \Upsilon & \text{si } \beta \in]0, 1[\\ d \Upsilon & \text{si } \beta \in \{0, 1\} \end{cases}.$$

- 1 Algorithme LLL
 - Réduction des réseaux
 - Exemple d'application
 - Algorithme LLL
- 2 Modélisations de l'algorithme LLL et de ses entrées
 - Modèles d'exécution (part 1)
 - Modèles d'entrée
- 3 Résultats obtenus dans les différents modèles
 - Modèle M1 : cfg/tas de sable
 - Modèle M1 et modèles d'entrée
 - Modèle M2 : système dynamique de $\mathbb{R}^d$
 - Modèle M3 : systèmes dynamiques probabilistes de $\mathbb{R}^d$
- 4 Conclusion

Chip Firing Game (M1) : α fixé

Chip Firing Game : le modèle $\mathcal{C}_d(\mathbf{c}, H, \alpha)$

si $c_i > H$ alors $\check{c}_{i-1} := c_{i-1} + \alpha$ $\check{c}_i := c_i - 2\alpha$ $\check{c}_{i+1} := c_{i+1} + \alpha$



$K(\mathbf{c} \mapsto \hat{\mathbf{c}})$ est indépendante du chemin

Énergie d'un cfg : $\mathcal{E}(\mathbf{c}) := \sum_{i=1}^{d-1} i(d-i)c_i$
 $\mathcal{E}(\check{\mathbf{c}}) = \mathcal{E}(\mathbf{c}) - 2\alpha$

La configuration finale est unique
 et ne dépend pas de la stratégie.

$$K(\mathbf{c}) = \frac{1}{2\alpha} [\mathcal{E}(\mathbf{c}) - \mathcal{E}(\hat{\mathbf{c}})] = \frac{1}{2\alpha} \sum_{i=1}^{d-1} i(d-i)(c_i - \hat{c}_i).$$

Déterminer

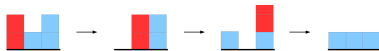
- le nombre d'itérations
- la configuration finale et son énergie

deux problèmes très liés

Chip Firing Game (M1) : α fixé

Chip Firing Game : le modèle $\mathcal{C}_d(\mathbf{c}, H, \alpha)$

si $c_i > H$ alors $\check{c}_{i-1} := c_{i-1} + \alpha$ $\check{c}_i := c_i - 2\alpha$ $\check{c}_{i+1} := c_{i+1} + \alpha$



La configuration finale est **unique**
et **ne dépend** pas de la stratégie.

$K(\mathbf{c} \mapsto \hat{\mathbf{c}})$ est **indépendante** du chemin

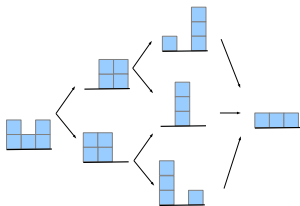
Énergie d'un cfg : $\mathcal{E}(\mathbf{c}) := \sum_{i=1}^{d-1} i(d-i)c_i$
 $\mathcal{E}(\check{\mathbf{c}}) = \mathcal{E}(\mathbf{c}) - 2\alpha$

$$K(\mathbf{c}) = \frac{1}{2\alpha} [\mathcal{E}(\mathbf{c}) - \mathcal{E}(\hat{\mathbf{c}})] = \frac{1}{2\alpha} \sum_{i=1}^{d-1} i(d-i)(c_i - \hat{c}_i).$$

Déterminer

- le nombre d'itérations
- la configuration finale et son énergie

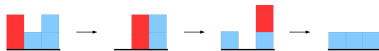
deux problèmes très liés



Chip Firing Game (M1) : α fixé

Chip Firing Game : le modèle $\mathcal{C}_d(\mathbf{c}, H, \alpha)$

si $c_i > H$ alors $\check{c}_{i-1} := c_{i-1} + \alpha$ $\check{c}_i := c_i - 2\alpha$ $\check{c}_{i+1} := c_{i+1} + \alpha$

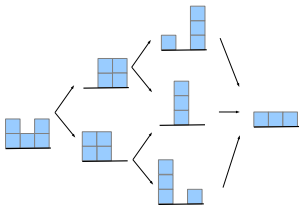


La configuration finale est **unique**
et **ne dépend** pas de la stratégie.

$K(\mathbf{c} \mapsto \hat{\mathbf{c}})$ est **indépendante** du chemin

Énergie d'un *cfg* : $\mathcal{E}(\mathbf{c}) := \sum_{i=1}^{d-1} i(d-i)c_i$
 $\mathcal{E}(\check{\mathbf{c}}) = \mathcal{E}(\mathbf{c}) - 2\alpha$

$$K(\mathbf{c}) = \frac{1}{2\alpha} [\mathcal{E}(\mathbf{c}) - \mathcal{E}(\hat{\mathbf{c}})] = \frac{1}{2\alpha} \sum_{i=1}^{d-1} i(d-i)(c_i - \hat{c}_i).$$



Déterminer

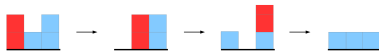
- le nombre d'itérations
- la configuration finale et son énergie

deux problèmes très liés

Chip Firing Game (M1) : α fixé

Chip Firing Game : le modèle $\mathcal{C}_d(\mathbf{c}, H, \alpha)$

si $c_i > H$ alors $\check{c}_{i-1} := c_{i-1} + \alpha$ $\check{c}_i := c_i - 2\alpha$ $\check{c}_{i+1} := c_{i+1} + \alpha$

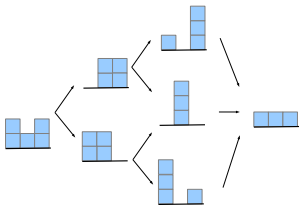


La configuration finale est **unique**
et **ne dépend** pas de la stratégie.

$K(\mathbf{c} \mapsto \hat{\mathbf{c}})$ est **indépendante** du chemin

Énergie d'un *cfg* : $\mathcal{E}(\mathbf{c}) := \sum_{i=1}^{d-1} i(d-i)c_i$
 $\mathcal{E}(\check{\mathbf{c}}) = \mathcal{E}(\mathbf{c}) - 2\alpha$

$$K(\mathbf{c}) = \frac{1}{2\alpha} [\mathcal{E}(\mathbf{c}) - \mathcal{E}(\hat{\mathbf{c}})] = \frac{1}{2\alpha} \sum_{i=1}^{d-1} i(d-i)(c_i - \hat{c}_i).$$



Déterminer

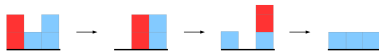
- le nombre d'itérations
- la configuration finale et son énergie

deux problèmes très liés

Chip Firing Game (M1) : α fixé

Chip Firing Game : le modèle $\mathcal{C}_d(\mathbf{c}, H, \alpha)$

si $c_i > H$ alors $\check{c}_{i-1} := c_{i-1} + \alpha$ $\check{c}_i := c_i - 2\alpha$ $\check{c}_{i+1} := c_{i+1} + \alpha$

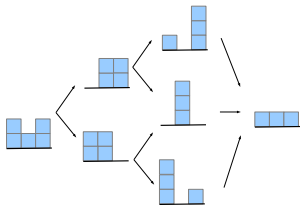


La configuration finale est **unique**
et **ne dépend** pas de la stratégie.

$K(\mathbf{c} \mapsto \hat{\mathbf{c}})$ est **indépendante** du chemin

Énergie d'un *cfg* : $\mathcal{E}(\mathbf{c}) := \sum_{i=1}^{d-1} i(d-i)c_i$
 $\mathcal{E}(\check{\mathbf{c}}) = \mathcal{E}(\mathbf{c}) - 2\alpha$

$$K(\mathbf{c}) = \frac{1}{2\alpha} [\mathcal{E}(\mathbf{c}) - \mathcal{E}(\hat{\mathbf{c}})] = \frac{1}{2\alpha} \sum_{i=1}^{d-1} i(d-i)(c_i - \hat{c}_i).$$



Déterminer

- le nombre d'itérations
- la configuration finale et son énergie

deux problèmes très liés

Réduction des réseaux dans le modèle CFG

CFG	Réseaux
$\mathbf{c} = (c_1, \dots, c_{d-1})$	$\mathbf{c} = (\log_s r_1, \dots, \log_s r_{d-1})$
Masse $\mathcal{M}(\mathbf{c}) = \sum_{i=1}^{d-1} c_i$	$\log_s \prod_{i=1}^{d-1} r_i$
Énergie $\mathcal{E}(\mathbf{c})$	$\log_s \Delta(B)$
$K(B) = \frac{1}{2\alpha} (\mathcal{E}(\mathbf{c}) - \mathcal{E}(\hat{\mathbf{c}}))$	$K(B) = \frac{1}{2\alpha} \log_s \frac{\Delta(B)}{\Delta(\hat{B})}$

Pour un réseau **totalement non réduit**, la masse $\mathcal{M}(\mathbf{c})$ quantifie la **difficulté de la réduction**

$$\Upsilon := \mathcal{M}(\mathbf{c}) = \log_s \ell_1 - \log_s \ell_d = \log_s \ell_{\max} - \log_s \ell_{\min}$$

Réduction des réseaux dans le modèle CFG

CFG	Réseaux
$\mathbf{c} = (c_1, \dots, c_{d-1})$	$\mathbf{c} = (\log_s r_1, \dots, \log_s r_{d-1})$
Masse $\mathcal{M}(\mathbf{c}) = \sum_{i=1}^{d-1} c_i$	$\log_s \prod_{i=1}^{d-1} r_i$
Énergie $\mathcal{E}(\mathbf{c})$	$\log_s \Delta(B)$
$K(B) = \frac{1}{2\alpha} (\mathcal{E}(\mathbf{c}) - \mathcal{E}(\hat{\mathbf{c}}))$	$K(B) = \frac{1}{2\alpha} \log_s \frac{\Delta(B)}{\Delta(\hat{B})}$

Pour un réseau **totalement non réduit**, la masse $\mathcal{M}(\mathbf{c})$ quantifie la **difficulté de la réduction**

$$\Upsilon := \mathcal{M}(\mathbf{c}) = \log_s \ell_1 - \log_s \ell_d = \log_s \ell_{\max} - \log_s \ell_{\min}$$

Réduction des réseaux dans le modèle CFG

CFG	Réseaux
$\mathbf{c} = (c_1, \dots, c_{d-1})$	$\mathbf{c} = (\log_s r_1, \dots, \log_s r_{d-1})$
Masse $\mathcal{M}(\mathbf{c}) = \sum_{i=1}^{d-1} c_i$	$\log_s \prod_{i=1}^{d-1} r_i$
Énergie $\mathcal{E}(\mathbf{c})$	$\log_s \Delta(B)$
$K(B) = \frac{1}{2\alpha} (\mathcal{E}(\mathbf{c}) - \mathcal{E}(\hat{\mathbf{c}}))$	$K(B) = \frac{1}{2\alpha} \log_s \frac{\Delta(B)}{\Delta(\hat{B})}$

Pour un réseau **totalement non réduit**, la masse $\mathcal{M}(\mathbf{c})$ quantifie la **difficulté de la réduction**

$$\Upsilon := \mathcal{M}(\mathbf{c}) = \log_s \ell_1 - \log_s \ell_d = \log_s \ell_{\max} - \log_s \ell_{\min}$$

1 Algorithme LLL

- Réduction des réseaux
- Exemple d'application
- Algorithme LLL

2 Modélisations de l'algorithme LLL et de ses entrées

- Modèles d'exécution (part 1)
- Modèles d'entrée

3 Résultats obtenus dans les différents modèles

- Modèle M1 : cfg/tas de sable
- **Modèle M1 et modèles d'entrée**
- Modèle M2 : système dynamique de $\mathbb{R}^d$
- Modèle M3 : systèmes dynamiques probabilistes de $\mathbb{R}^d$

4 Conclusion

Réseaux uni-tas

Pour un cfg $\mathcal{C}_d(\mathbf{c}, h, H)$ avec une seule pile en position i de hauteur $\Upsilon = \Theta(d^a)$.

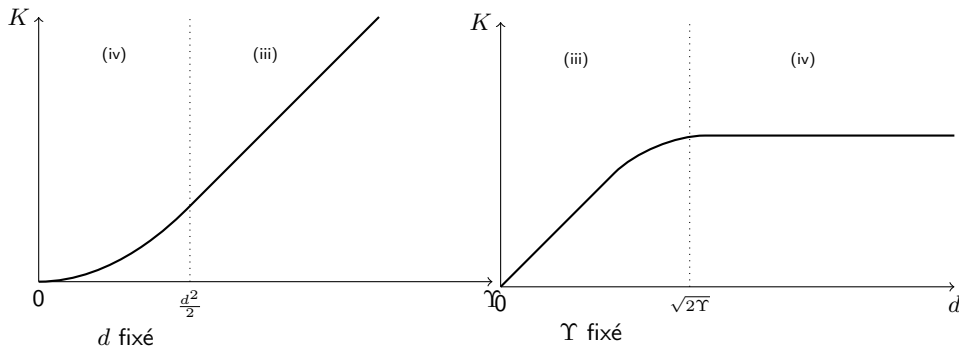
Si **la pile n'est pas sur les bords**, i.e., $i = \beta d$ avec $\beta \in]0, 1[$

- Si $0 < a \leq 1$ (ne s'étale pas jusqu'au bord), alors $K = O(d^3)$
- Si $a > 1$ (s'étale jusqu'au bord), alors $K = \Theta(d^{2+a})$

Si **la pile est sur les bords**, en position $i \in \{1, d-1\}$:

- Si $a \leq 2$ (ne s'étale pas jusqu'à l'autre bord), alors on a $K = O(d^3)$
- Si $a > 2$ (s'étale complètement), alors on a $K = \Theta(d^{a+1})$

"uni-tas" au début



Entrées "grand-tas" et cfg

Nombre d'itérations :

dans le modèle $\mathcal{A}(\Upsilon, d, g)$, la masse moyenne Υ est une fonction au moins linéaire de d , et l'énergie initiale $\mathcal{E} = \mathcal{E}(\mathbf{c})$ moyenne vérifie

$$\mathbb{E}[\mathcal{E}] \sim d^2 \Upsilon I(g), \quad \text{avec} \quad I(g) = \int_0^1 x(1-x)g(x)dx.$$

Il y a deux cas principaux :

Cas où la masse moyenne Υ vérifie $\Upsilon = \Theta(d^a)$ avec $a > 1$.

$$\mathbb{E}[K] = \Theta(d^{a+2})$$

Cas où la masse moyenne Υ est linéaire en d et asymptotique à $C \cdot d$, avec $C > 1$.

$$\mathbb{E}[K] = \Theta(d^3)$$

$$\tilde{\Upsilon} = \log \prod r_i = \frac{1}{\log s} \Upsilon \quad s = \text{paramètre condition de Siegel}$$

Modèles	K pire des cas	K pour M1(α)	K pour M2(μ)	K exp.
$\mathcal{A}(\tilde{\Upsilon}, d)$	$\frac{1}{12 \log t} d(d+1) \tilde{\Upsilon}$	$\frac{1}{12\alpha} d(d+1) \tilde{\Upsilon}$		$\Theta(d^2 \tilde{\Upsilon})$
$\mathcal{N}(\tilde{\Upsilon}, d)$	$\frac{1}{8 \log t} d^2 \tilde{\Upsilon}$	$\frac{1}{8\alpha} d^2 \tilde{\Upsilon}$		$\Theta(d^2 \tilde{\Upsilon})$
$\mathcal{K}(\tilde{\Upsilon}, d)$	$\frac{1}{2 \log t} d \tilde{\Upsilon}$	$\frac{1}{2\alpha} d \tilde{\Upsilon}$		$\Theta(d \tilde{\Upsilon})$

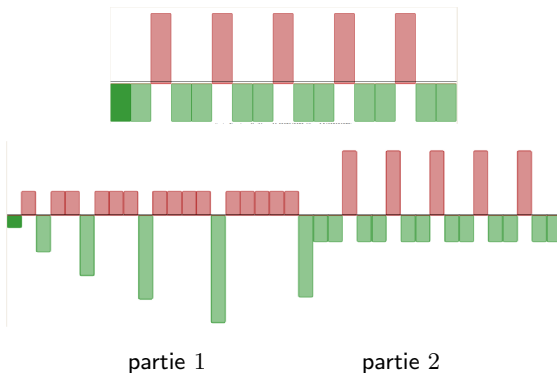
TABLE: La réduction est supposée difficile : $\tilde{\Upsilon}/d^a \rightarrow \infty$ avec $a = 1$ pour les modèles $\mathcal{A}, \mathcal{N}$ et $a = 2$ pour le modèle $\mathcal{K}$.

$\mathcal{A}$ =Ajtai (grands-tas), $\mathcal{N}$ =NTRU (uni-tas au milieu),
 $\mathcal{K}$ =Sac-à-dos (uni-tas au début)

CFG à trous : réseaux de Coppersmith

Interviennent dans la méthode de Coppersmith

- qui permet de calculer les **petites racines** de polynômes multivariés modulo un entier
- utiliser dans les attaques de RSA



CFG à trous : indépendance

$\mathbf{c}_g$ et $\mathbf{c}_d$: deux *cfg* avec des piles strictement positives à gauche et à droite,
avec des configurations finales $\hat{\mathbf{c}}_g$ et $\hat{\mathbf{c}}_d$

$\mathbf{c}_m$: un *cfg* avec des piles négatives ou nulles au milieu

Indépendance

Les $\mathbf{c}_g$ et $\mathbf{c}_d$ sont indépendants dans le *cfg* composé des trois blocs s'il existe une configuration $\hat{\mathbf{c}}_m$ réduite telle que la configuration finale du *cfg* total est $\hat{\mathbf{c}}_g \cdot \hat{\mathbf{c}}_m \cdot \hat{\mathbf{c}}_d$

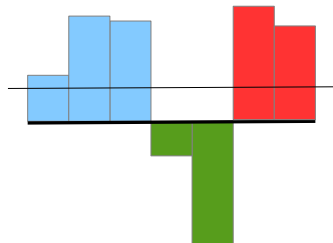
Des conditions suffisantes pour l'indépendance et la dépendance :

pour l'indépendance :

$$\mathcal{M}_D + \mathcal{M}_G \leq \mathcal{M}$$

pour la dépendance :

$$\mathcal{M}_D + \mathcal{M}_G > \mathcal{M}$$



CFG à trous : indépendance

$\mathbf{c}_g$ et $\mathbf{c}_d$: deux *cfg* avec des piles strictement positives à gauche et à droite, avec des configurations finales $\hat{\mathbf{c}}_g$ et $\hat{\mathbf{c}}_d$

$\mathbf{c}_m$: un *cfg* avec des piles négatives ou nulles au milieu

Indépendance

Les $\mathbf{c}_g$ et $\mathbf{c}_d$ sont indépendants dans le *cfg* composé des trois blocs s'il existe une configuration $\hat{\mathbf{c}}_m$ réduite telle que la configuration finale du *cfg* total est $\hat{\mathbf{c}}_g \cdot \hat{\mathbf{c}}_m \cdot \hat{\mathbf{c}}_d$

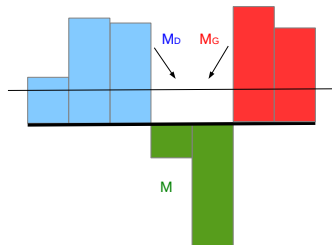
Des conditions suffisantes pour l'indépendance et la dépendance :

pour l'indépendance :

$$M_D + M_G \leq M$$

pour la dépendance :

$$M_D + M_G > M$$



CFG à trous : indépendance

$\mathbf{c}_g$ et $\mathbf{c}_d$: deux *cfg* avec des piles strictement positives à gauche et à droite,
avec des configurations finales $\hat{\mathbf{c}}_g$ et $\hat{\mathbf{c}}_d$

$\mathbf{c}_m$: un *cfg* avec des piles négatives ou nulles au milieu

Indépendance

Les $\mathbf{c}_g$ et $\mathbf{c}_d$ sont indépendants dans le *cfg* composé des trois blocs s'il existe une configuration $\hat{\mathbf{c}}_m$ réduite telle que la configuration finale du *cfg* total est $\hat{\mathbf{c}}_g \cdot \hat{\mathbf{c}}_m \cdot \hat{\mathbf{c}}_d$

Des conditions suffisantes pour l'indépendance et la dépendance :

pour l'indépendance :

$$\mathcal{M}_D + \mathcal{M}_G \leq \mathcal{M}$$

pour la dépendance :

$$\mathcal{M}_D + \mathcal{M}_G > \mathcal{M}$$



CFG à trous : indépendance

$\mathbf{c}_g$ et $\mathbf{c}_d$: deux *cfg* avec des piles strictement positives à gauche et à droite,
avec des configurations finales $\hat{\mathbf{c}}_g$ et $\hat{\mathbf{c}}_d$

$\mathbf{c}_m$: un *cfg* avec des piles négatives ou nulles au milieu

Indépendance

Les $\mathbf{c}_g$ et $\mathbf{c}_d$ sont indépendants dans le *cfg* composé des trois blocs s'il existe une configuration $\hat{\mathbf{c}}_m$ réduite telle que la configuration finale du *cfg* total est $\hat{\mathbf{c}}_g \cdot \hat{\mathbf{c}}_m \cdot \hat{\mathbf{c}}_d$

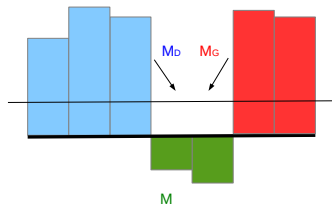
Des conditions suffisantes pour l'indépendance et la dépendance :

pour l'indépendance :

$$\mathcal{M}_D + \mathcal{M}_G \leq \mathcal{M}$$

pour la dépendance :

$$\mathcal{M}_D + \mathcal{M}_G > \mathcal{M}$$



CFG à trous : indépendance

$\mathbf{c}_g$ et $\mathbf{c}_d$: deux *cfg* avec des piles strictement positives à gauche et à droite, avec des configurations finales $\hat{\mathbf{c}}_g$ et $\hat{\mathbf{c}}_d$

$\mathbf{c}_m$: un *cfg* avec des piles négatives ou nulles au milieu

Indépendance

Les $\mathbf{c}_g$ et $\mathbf{c}_d$ sont indépendants dans le *cfg* composé des trois blocs s'il existe une configuration $\hat{\mathbf{c}}_m$ réduite telle que la configuration finale du *cfg* total est $\hat{\mathbf{c}}_g \cdot \hat{\mathbf{c}}_m \cdot \hat{\mathbf{c}}_d$

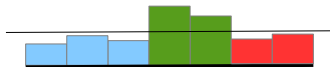
Des conditions suffisantes pour l'indépendance et la dépendance :

pour l'indépendance :

$$\mathcal{M}_D + \mathcal{M}_G \leq \mathcal{M}$$

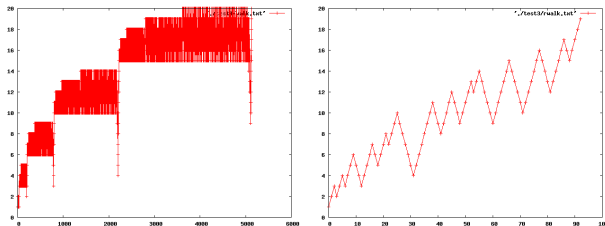
pour la dépendance :

$$\mathcal{M}_D + \mathcal{M}_G > \mathcal{M}$$

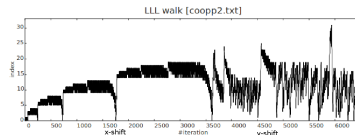


Les entrées "à trous"

Marche aléatoire de l'indice sur le réseau de Coppersmith



Une grande partie de la réduction peut être effectuée indépendamment sur les blocs



1 Algorithme LLL

- Réduction des réseaux
- Exemple d'application
- Algorithme LLL

2 Modélisations de l'algorithme LLL et de ses entrées

- Modèles d'exécution (part 1)
- Modèles d'entrée

3 Résultats obtenus dans les différents modèles

- Modèle M1 : cfg/tas de sable
- Modèle M1 et modèles d'entrée
- **Modèle M2 : système dynamique de $\mathbb{R}^d$**
- Modèle M3 : systèmes dynamiques probabilistes de $\mathbb{R}^d$

4 Conclusion

Modèle M2 : systèmes dynamiques de $\mathbb{R}^{d-1}$

ρ est fonction de r_i , ν_i fixé dans $[-\frac{1}{2}, \frac{1}{2}]$

On pose :

$$x_i := r_i^2 = \frac{\ell_{i+1}^2}{\ell_i^2} \leftarrow \text{variable}, \quad \check{x}_i := \check{r}_i^2 = \frac{\check{\ell}_{i+1}^2}{\check{\ell}_i^2}, \quad \mu := \nu_i^2 \leftarrow \text{constant}$$

Le facteur de décroissance : $\rho = x_i + \mu$

M2(t, μ)

$$\text{Si } r_i^2 < \frac{1}{t^2} - \mu$$

$$\check{r}_{i-1} = \rho r_{i-1}$$

$$\check{r}_i = \frac{1}{\rho^2} r_i$$

$$\check{r}_{i+1} = \rho r_{i+1}.$$

$$\text{Si } x_i < \frac{1}{t^2} - \mu$$

$$\check{x}_{i-1} = x_{i-1}(x_i + \mu)$$

$$\check{x}_i = \frac{x_i}{(x_i + \mu)^2}$$

$$\check{x}_{i+1} = x_{i+1}(x_i + \mu)$$

M2(t, μ) modélise et analyse l'algorithme LLL(t) (également pour $t = 1$)

Le trou du système M2(t, μ) (condition d'arrêt) est $(x_1, \dots, x_{d-1}) \in [\frac{1}{t^2} - \mu, +\infty[^{d-1}$.

Modèle M2 : systèmes dynamiques de $\mathbb{R}^{d-1}$

ρ est fonction de r_i , ν_i fixé dans $[-\frac{1}{2}, \frac{1}{2}]$

On pose :

$$x_i := r_i^2 = \frac{\ell_{i+1}^2}{\ell_i^2} \leftarrow \text{variable}, \quad \check{x}_i := \check{r}_i^2 = \frac{\check{\ell}_{i+1}^2}{\check{\ell}_i^2}, \quad \mu := \nu_i^2 \leftarrow \text{constant}$$

Le facteur de décroissance : $\rho = x_i + \mu$

M2(t, μ)

$$\text{Si } r_i^2 < \frac{1}{t^2} - \mu$$

$$\check{r}_{i-1} = \rho r_{i-1}$$

$$\check{r}_i = \frac{1}{\rho^2} r_i$$

$$\check{r}_{i+1} = \rho r_{i+1}.$$

$$\text{Si } x_i < \frac{1}{t^2} - \mu$$

$$\check{x}_{i-1} = x_{i-1}(x_i + \mu)$$

$$\check{x}_i = \frac{x_i}{(x_i + \mu)^2}$$

$$\check{x}_{i+1} = x_{i+1}(x_i + \mu)$$

M2(t, μ) modélise et analyse l'algorithme LLL(t) (également pour $t = 1$)

Le trou du système M2(t, μ) (condition d'arrêt) est $(x_1, \dots, x_{d-1}) \in [\frac{1}{t^2} - \mu, +\infty[^{d-1}$.

Modèle M2 : systèmes dynamiques de $\mathbb{R}^{d-1}$

ρ est fonction de r_i , ν_i fixé dans $[-\frac{1}{2}, \frac{1}{2}]$

On pose :

$$x_i := r_i^2 = \frac{\ell_{i+1}^2}{\ell_i^2} \leftarrow \text{variable}, \quad \check{x}_i := \check{r}_i^2 = \frac{\check{\ell}_{i+1}^2}{\check{\ell}_i^2}, \quad \mu := \nu_i^2 \leftarrow \text{constant}$$

Le facteur de décroissance : $\rho = x_i + \mu$

M2(t, μ)

$$\text{Si } r_i^2 < \frac{1}{t^2} - \mu$$

$$\check{r}_{i-1} = \rho r_{i-1}$$

$$\check{r}_i = \frac{1}{\rho^2} r_i$$

$$\check{r}_{i+1} = \rho r_{i+1}.$$

$$\text{Si } x_i < \frac{1}{t^2} - \mu$$

$$\check{x}_{i-1} = x_{i-1}(x_i + \mu)$$

$$\check{x}_i = \frac{x_i}{(x_i + \mu)^2}$$

$$\check{x}_{i+1} = x_{i+1}(x_i + \mu)$$

M2(t, μ) modélise et analyse l'algorithme LLL(t) (également pour $t = 1$)

Le trou du système M2(t, μ) (condition d'arrêt) est $(x_1, \dots, x_{d-1}) \in [\frac{1}{t^2} - \mu, +\infty[^{d-1}$.

Modèle M2 : systèmes dynamiques de $\mathbb{R}^{d-1}$

ρ est fonction de r_i , ν_i fixé dans $[-\frac{1}{2}, \frac{1}{2}]$

On pose :

$$x_i := r_i^2 = \frac{\ell_{i+1}^2}{\ell_i^2} \leftarrow \text{variable}, \quad \check{x}_i := \check{r}_i^2 = \frac{\check{\ell}_{i+1}^2}{\check{\ell}_i^2}, \quad \mu := \nu_i^2 \leftarrow \text{constant}$$

Le facteur de décroissance : $\rho = x_i + \mu$

M2(t, μ)

$$\text{Si } r_i^2 < \frac{1}{t^2} - \mu$$

$$\check{r}_{i-1} = \rho r_{i-1}$$

$$\check{r}_i = \frac{1}{\rho^2} r_i$$

$$\check{r}_{i+1} = \rho r_{i+1}.$$

$$\text{Si } x_i < \frac{1}{t^2} - \mu$$

$$\check{x}_{i-1} = x_{i-1}(x_i + \mu)$$

$$\check{x}_i = \frac{x_i}{(x_i + \mu)^2}$$

$$\check{x}_{i+1} = x_{i+1}(x_i + \mu)$$

M2(t, μ) modélise et analyse l'algorithme LLL(t) (également pour $t = 1$)

Le trou du système M2(t, μ) (condition d'arrêt) est $(x_1, \dots, x_{d-1}) \in [\frac{1}{t^2} - \mu, +\infty[^{d-1}$.

Étude du modèle M2

M2 n'est plus un tas de sable : c'est un système dynamique général, "à trou"

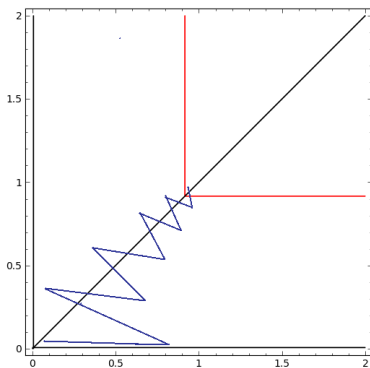
Nous analysons ce modèle avec $\mu \leq 1/4$:

- en dimension 2 : pour $t \geq 1$
 - des résultats similaires à ceux connus sur l'algorithme de Gauss
 - une distribution géométrique pour le nombre d'itérations
- en dimension 3 : (le premier cas où l'algorithme $LLL(t)$ n'est pas bien connu)
 - pour $t \geq 1$: le nombre d'étapes suit toujours une loi géométrique sauf pour des densités particulières
- en dimension d :
 - pour $t > 1$: l'asymptotique du nombre d'itérations du modèle $M2(t, \mu)$, ce résultat n'étant pas connu pour $LLL(t)$
 - le passage entre $LLL(t)$ et $LLL(1)$ est conjecturé.

Modèle M2 en dimension 3

$$T_1(x, y) = \left(\frac{x}{(x + \mu)^2}, y(x + \mu) \right), \quad T_2(x, y) = \left(x(y + \mu), \frac{y}{(y + \mu)^2} \right)$$

$$\text{si } x < \frac{1}{t^2} - \mu \quad \text{si } y < \frac{1}{t^2} - \mu$$



$$\mathcal{O}_{\mu,t}^2 = \{(x, y) : x \geq \frac{1}{t^2} - \mu; y \geq \frac{1}{t^2} - \mu\}.$$

La stratégie gloutonne :

(si $x \leq y$ on applique T_1 , sinon T_2)

$$\text{alors } \frac{\tilde{x}\tilde{y}}{xy} = \frac{1}{\min(x, y) + \mu} > 1$$

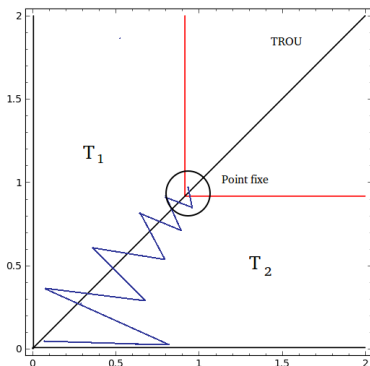
xy joue le même rôle que
 $\mathcal{E}$ (l'énergie d'un cfg) et
 Δ (le potentiel d'une base)

Difficulté : point fixe attractif $(1 - \mu, 1 - \mu)$

Modèle M2 en dimension 3

$$T_1(x, y) = \left(\frac{x}{(x + \mu)^2}, y(x + \mu) \right), \quad T_2(x, y) = \left(x(y + \mu), \frac{y}{(y + \mu)^2} \right)$$

si $x < \frac{1}{t^2} - \mu$
si $y < \frac{1}{t^2} - \mu$



$$\mathcal{O}_{\mu,t}^2 = \{(x, y) : x \geq \frac{1}{t^2} - \mu; y \geq \frac{1}{t^2} - \mu\}.$$

La stratégie gloutonne :

(si $x \leq y$ on applique T_1 , sinon T_2)

$$\text{alors} \quad \frac{\tilde{x}\tilde{y}}{xy} = \frac{1}{\min(x, y) + \mu} > 1$$

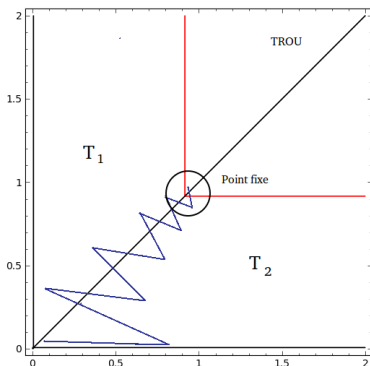
xy joue le même rôle que
 $\mathcal{E}$ (l'énergie d'un cfg) et
 Δ (le potentiel d'une base)

Difficulté : point fixe attractif $(1 - \mu, 1 - \mu)$

Modèle M2 en dimension 3

$$T_1(x, y) = \left(\frac{x}{(x + \mu)^2}, y(x + \mu) \right), \quad T_2(x, y) = \left(x(y + \mu), \frac{y}{(y + \mu)^2} \right)$$

$$\text{si } x < \frac{1}{t^2} - \mu \quad \text{si } y < \frac{1}{t^2} - \mu$$



$$\mathcal{O}_{\mu,t}^2 = \{(x, y) : x \geq \frac{1}{t^2} - \mu; y \geq \frac{1}{t^2} - \mu\}.$$

La stratégie gloutonne :

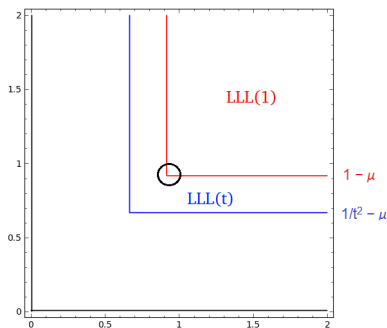
(si $x \leq y$ on applique T_1 , sinon T_2)

$$\text{alors } \frac{\tilde{x}\tilde{y}}{xy} = \frac{1}{\min(x, y) + \mu} > 1$$

xy joue le même rôle que
 $\mathcal{E}$ (l'énergie d'un cfg) et
 Δ (le potentiel d'une base)

Difficulté : point fixe attractif $(1 - \mu, 1 - \mu)$

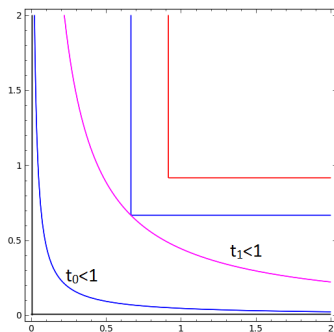
Hyperbole : $\mathcal{D}_{t,\mu} = \{(x, y) \in I_\mu \times I_\mu \mid xy \geq (\frac{1}{t^2} - \mu)^2\}$



idée de la preuve (trois phases)

- jusqu'au $\mathcal{D}_{t,\mu}$, $t > 1$ ($\text{LLL}(t)$)
- passage de $\mathcal{D}_{t,\mu}$ à $\mathcal{D}_{1,\mu}$
- entre $\mathcal{D}_{1,\mu}$ et le trou ($\text{LLL}(1)$)

Hyperbole : $\mathcal{D}_{t,\mu} = \{(x, y) \in I_\mu \times I_\mu \mid xy \geq (\frac{1}{t^2} - \mu)^2\}$



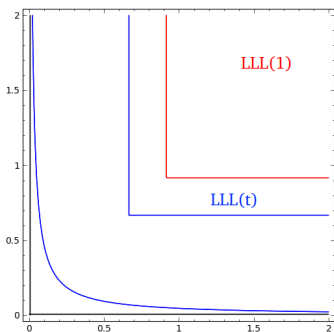
Pour $t > 1$ et $\mu \leq 1/4$, lorsque $xy \rightarrow 0$
le nombre d'itérations $K_{t,\mu}$ du
système dynamique M2(t, μ) satisfait :

$$K_{t,\mu}(x, y) = \log_\mu xy + O(1)$$

idée de la preuve (trois phases)

- jusqu'au $\mathcal{D}_{t,\mu}$, $t > 1$ (LLL(t))
- passage de $\mathcal{D}_{t,\mu}$ à $\mathcal{D}_{1,\mu}$
- entre $\mathcal{D}_{1,\mu}$ et le trou (LLL(1))

Hyperbole : $\mathcal{D}_{t,\mu} = \{(x, y) \in I_\mu \times I_\mu \mid xy \geq (\frac{1}{t^2} - \mu)^2\}$



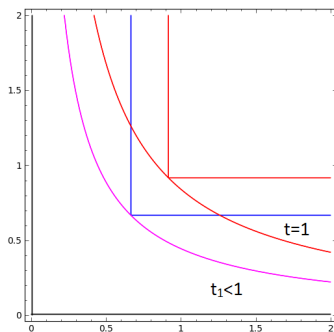
Pour $t > 1$ et $\mu \leq 1/4$, lorsque $xy \rightarrow 0$ le nombre d'itérations $K_{t,\mu}$ du système dynamique M2(t, μ) satisfait :

$$K_{t,\mu}(x, y) = \log_\mu xy + O(1)$$

idée de la preuve (trois phases)

- jusqu'au $\mathcal{D}_{t,\mu}$, $t > 1$ (LLL(t))
- passage de $\mathcal{D}_{t,\mu}$ à $\mathcal{D}_{1,\mu}$
- entre $\mathcal{D}_{1,\mu}$ et le trou (LLL(1))

Hyperbole : $\mathcal{D}_{t,\mu} = \{(x, y) \in I_\mu \times I_\mu \mid xy \geq (\frac{1}{t^2} - \mu)^2\}$



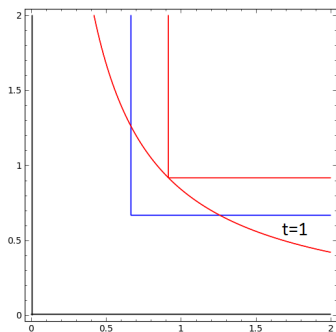
idée de la preuve (trois phases)

- jusqu'au $\mathcal{D}_{t,\mu}$, $t > 1$ (LLL(t))
- passage de $\mathcal{D}_{t,\mu}$ à $\mathcal{D}_{1,\mu}$
- entre $\mathcal{D}_{1,\mu}$ et le trou (LLL(1))

Pour $t \in]1, 2/\sqrt{3}[$ en partant d'un point $(x, y) \in \mathcal{D}_{t,\mu}$, le système applique alternativement les fonctions T_1 et T_2

Pour tout $\mu \in]0, 1/4]$, il existe $t_1 \leq 2/\sqrt{3}$ pour lequel, pour tout $t \leq t_1$, le système M2($1, \mu$) effectue **au plus 3 itérations** sur $\mathcal{D}_{t,\mu}$.

Hyperbole : $\mathcal{D}_{t,\mu} = \{(x, y) \in I_\mu \times I_\mu \mid xy \geq (\frac{1}{t^2} - \mu)^2\}$

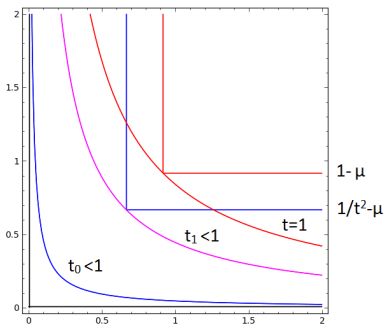


Le système dynamique $M2(1, \mu)$ effectue **au plus une itération** sur le domaine $\mathcal{D}_{1,\mu}$.

idée de la preuve (trois phases)

- jusqu'au $\mathcal{D}_{t,\mu}$, $t > 1$ (LLL(t))
- passage de $\mathcal{D}_{t,\mu}$ à $\mathcal{D}_{1,\mu}$
- entre $\mathcal{D}_{1,\mu}$ et le trou (LLL(1))

Hyperbole : $\mathcal{D}_{t,\mu} = \{(x, y) \in I_\mu \times I_\mu \mid xy \geq (\frac{1}{t^2} - \mu)^2\}$



idée de la preuve (trois phases)

- jusqu'au $\mathcal{D}_{t,\mu}$, $t > 1$ (LLL(t))
- passage de $\mathcal{D}_{t,\mu}$ à $\mathcal{D}_{1,\mu}$
- entre $\mathcal{D}_{1,\mu}$ et le trou (LLL(1))

Pour $t > 1$ et $\mu \leq 1/4$, lorsque $xy \rightarrow 0$
le nombre d'itérations $K_{t,\mu}$ du
système dynamique M2(t, μ) satisfait :

$$K_{t,\mu}(x, y) = \log_\mu xy + O(1)$$

Pour tout $\mu \in]0, 1/4]$, il existe $t_1 \leq 2/\sqrt{3}$ pour
lequel, pour tout $t \leq t_1$, le système M2($1, \mu$)
effectue **au plus 3 itérations** sur $\mathcal{D}_{t,\mu}$.

Le système dynamique M2($1, \mu$) effectue **au plus
une itération** sur le domaine $\mathcal{D}_{1,\mu}$.

Proposition

Pour $t \geq 1$ et $\mu \leq 1/4$, le nombre d'itérations $K_{t,\mu}$ du système dynamique $M2(t, \mu)$ satisfait : $K_{t,\mu}(x, y) = \log_{\mu} xy + O(1)$ lorsque $xy \rightarrow 0$ où la constante du O est uniforme pour $t \rightarrow 1$.

Le nombre d'itérations $K_{t,\mu}$ suit une loi asymptotiquement géométrique si (x, y) est distribué selon des lois puissances de paramètre r alors

- $K_{t,\mu}$ suit asymptotiquement une loi géométrique de raison μ^r
- la raison de la loi ne dépend que de μ et non de t

En dimension d

Une extension naturelle de la “quantité qui augmente” en dimension supérieure :

$$P(\mathbf{x}) := \prod_{i=1}^{d-1} x_i^{i(d-i)} = \prod_{i=1}^{d-1} r_i^{2i(d-i)},$$

Pour la stratégie gloutonne :

$$P(T(\mathbf{x})) = \frac{P(\mathbf{x})}{\min(x_i + \mu)^2} > P(\mathbf{x}),$$

Pour $t > 1$ et $\mu \leq 1/4$, pour tout $d \geq 3$ lorsque $P(\mathbf{x}) \rightarrow 0$ le nombre d'itérations $K_{t,\mu}$ du système dynamique M2(t, μ) satisfait :

$$K_{t,\mu}(\mathbf{x}) = \frac{1}{2} \log_{\mu} P(\mathbf{x}) + O(1)$$

K pire des cas LLL(t)	K pour M1(α)	K pour M2(μ)
$\frac{1}{\log t} \log \frac{\Delta(B)}{\Delta(\hat{B})}$	$\frac{1}{2\alpha} \log \frac{\Delta(B)}{\Delta(\hat{B})}$	$\frac{1}{ \log \mu } \log \frac{\Delta(B)}{\Delta(\hat{B})}$

$$t = 1?$$

En dimension d

Une extension naturelle de la “quantité qui augmente” en dimension supérieure :

$$P(\mathbf{x}) := \prod_{i=1}^{d-1} x_i^{i(d-i)} = \prod_{i=1}^{d-1} r_i^{2i(d-i)},$$

Pour la stratégie gloutonne :

$$P(T(\mathbf{x})) = \frac{P(\mathbf{x})}{\min(x_i + \mu)^2} > P(\mathbf{x}),$$

Pour $t > 1$ et $\mu \leq 1/4$, pour tout $d \geq 3$ lorsque $P(\mathbf{x}) \rightarrow 0$ le nombre d'itérations $K_{t,\mu}$ du système dynamique M2(t, μ) satisfait :

$$K_{t,\mu}(\mathbf{x}) = \frac{1}{2} \log_{\mu} P(\mathbf{x}) + O(1)$$

K pire des cas LLL(t)	K pour M1(α)	K pour M2(μ)
$\frac{1}{\log t} \log \frac{\Delta(B)}{\Delta(\hat{B})}$	$\frac{1}{2\alpha} \log \frac{\Delta(B)}{\Delta(\hat{B})}$	$\frac{1}{ \log \mu } \log \frac{\Delta(B)}{\Delta(\hat{B})}$

$t = 1?$

En dimension d

Une extension naturelle de la “quantité qui augmente” en dimension supérieure :

$$P(\mathbf{x}) := \prod_{i=1}^{d-1} x_i^{i(d-i)} = \prod_{i=1}^{d-1} r_i^{2i(d-i)},$$

Pour la stratégie gloutonne :

$$P(T(\mathbf{x})) = \frac{P(\mathbf{x})}{\min(x_i + \mu)^2} > P(\mathbf{x}),$$

Pour $t > 1$ et $\mu \leq 1/4$, pour tout $d \geq 3$ lorsque $P(\mathbf{x}) \rightarrow 0$ le nombre d'itérations $K_{t,\mu}$ du système dynamique M2(t, μ) satisfait :

$$K_{t,\mu}(\mathbf{x}) = \frac{1}{2} \log_{\mu} P(\mathbf{x}) + O(1)$$

K pire des cas LLL(t)	K pour M1(α)	K pour M2(μ)
$\frac{1}{\log t} \log \frac{\Delta(B)}{\Delta(\widehat{B})}$	$\frac{1}{2\alpha} \log \frac{\Delta(B)}{\Delta(\widehat{B})}$	$\frac{1}{ \log \mu } \log \frac{\Delta(B)}{\Delta(\widehat{B})}$

$$t = 1?$$

$$\tilde{\Upsilon} = \log \prod r_i = \frac{1}{\log s} \Upsilon \quad s = \text{paramètre condition de Siegel}$$

Modèles	K pire des cas	K pour M1(α)	K pour M2(μ)	K exp.
$\mathcal{A}(\tilde{\Upsilon}, d)$	$\frac{1}{12 \log t} d(d+1) \tilde{\Upsilon}$	$\frac{1}{12\alpha} d(d+1) \tilde{\Upsilon}$	$\frac{1}{6 \log \mu } d(d+1) \tilde{\Upsilon}$	$\Theta(d^2 \tilde{\Upsilon})$
$\mathcal{N}(\tilde{\Upsilon}, d)$	$\frac{1}{8 \log t} d^2 \tilde{\Upsilon}$	$\frac{1}{8\alpha} d^2 \tilde{\Upsilon}$	$\frac{1}{4 \log \mu } d^2 \tilde{\Upsilon}$	$\Theta(d^2 \tilde{\Upsilon})$
$\mathcal{K}(\tilde{\Upsilon}, d)$	$\frac{1}{2 \log t} d \tilde{\Upsilon}$	$\frac{1}{2\alpha} d \tilde{\Upsilon}$	$\frac{1}{ \log \mu } d \tilde{\Upsilon}$	$\Theta(d \tilde{\Upsilon})$

TABLE: La réduction est supposée difficile : $\tilde{\Upsilon}/d^a \rightarrow \infty$ avec $a = 1$ pour les modèles $\mathcal{A}, \mathcal{N}$ et $a = 2$ pour le modèle $\mathcal{K}$.

$\mathcal{A}$ =Ajtai (grands-tas), $\mathcal{N}$ =NTRU (uni-tas au milieu),
 $\mathcal{K}$ =Sac-à-dos (uni-tas au début)

1 Algorithme LLL

- Réduction des réseaux
- Exemple d'application
- Algorithme LLL

2 Modélisations de l'algorithme LLL et de ses entrées

- Modèles d'exécution (part 1)
- Modèles d'entrée

3 Résultats obtenus dans les différents modèles

- Modèle M1 : cfg/tas de sable
- Modèle M1 et modèles d'entrée
- Modèle M2 : système dynamique de $\mathbb{R}^d$
- Modèle M3 : systèmes dynamiques probabilistes de $\mathbb{R}^d$

4 Conclusion

Modèle M3 : systèmes dynamiques probabilistes

ρ est fonction de r_i , ν_i est uniforme dans $[-\frac{1}{2}, \frac{1}{2}]$

On pose :

$$x_i := r_i^2 = \frac{\ell_{i+1}^2}{\ell_i^2} \leftarrow \text{variable}, \quad \tilde{x}_i := \tilde{r}_i^2 = \frac{\tilde{\ell}_{i+1}^2}{\tilde{\ell}_i^2}, \quad \mu_i := \nu_i^2 \quad \text{avec} \quad \nu_i \in_R [-\frac{1}{2}, \frac{1}{2}]$$

M3(t)

$$\text{Si } x_i < \frac{1}{t^2} - \mu_i$$

$$\tilde{x}_{i-1} = x_{i-1}(x_i + \mu_i)$$

$$\tilde{x}_i = \frac{x_i}{(x_i + \mu_i)^2}$$

$$\tilde{x}_{i+1} = x_{i+1}(x_i + \mu_i)$$

Générer un nouveau μ_i

- Le facteur de décroissance : $\rho = x_i + \mu_i$
- M3(t) modélise et analyse l'algorithme LLL(t) (également pour $t = 1$)
- Le trou du système M2(t, μ) (condition d'arrêt) est

$$(x_1, \dots, x_{d-1}), (\mu_1, \dots, \mu_{d-1}) \quad \text{tels que}$$

$$\forall i = 1, \dots, d-1, \quad x_i \in \left[\frac{1}{t^2} - \mu_i, +\infty \right[.$$

C'est un système dynamique probabiliste dans $\mathbb{R}^{d-1}$, à trou, chaque transformation ayant un point fixe attractif

Modèle M3 : systèmes dynamiques probabilistes

ρ est fonction de r_i , ν_i est uniforme dans $[-\frac{1}{2}, \frac{1}{2}]$

On pose :

$$x_i := r_i^2 = \frac{\ell_{i+1}^2}{\ell_i^2} \leftarrow \text{variable}, \quad \tilde{x}_i := \tilde{r}_i^2 = \frac{\tilde{\ell}_{i+1}^2}{\tilde{\ell}_i^2}, \quad \mu_i := \nu_i^2 \quad \text{avec} \quad \nu_i \in_R [-\frac{1}{2}, \frac{1}{2}]$$

M3(t)

$$\text{Si } x_i < \frac{1}{t^2} - \mu_i$$

$$\tilde{x}_{i-1} = x_{i-1}(x_i + \mu_i)$$

$$\tilde{x}_i = \frac{x_i}{(x_i + \mu_i)^2}$$

$$\tilde{x}_{i+1} = x_{i+1}(x_i + \mu_i)$$

Générer un nouveau μ_i

- Le facteur de décroissance : $\rho = x_i + \mu_i$
- M3(t) modélise et analyse l'algorithme LLL(t) (également pour $t = 1$)
- Le trou du système M2(t, μ) (condition d'arrêt) est

$(x_1, \dots, x_{d-1}), (\mu_1, \dots, \mu_{d-1})$ tels que

$$\forall i = 1, \dots, d-1, \quad x_i \in \left[\frac{1}{t^2} - \mu_i, +\infty \right[.$$

C'est un système dynamique probabiliste dans $\mathbb{R}^{d-1}$, à trou, chaque transformation ayant un point fixe attractif

Modèle M3 : systèmes dynamiques probabilistes

ρ est fonction de r_i , ν_i est uniforme dans $[-\frac{1}{2}, \frac{1}{2}]$

On pose :

$$x_i := r_i^2 = \frac{\ell_{i+1}^2}{\ell_i^2} \leftarrow \text{variable}, \quad \tilde{x}_i := \tilde{r}_i^2 = \frac{\tilde{\ell}_{i+1}^2}{\tilde{\ell}_i^2}, \quad \mu_i := \nu_i^2 \quad \text{avec} \quad \nu_i \in_R [-\frac{1}{2}, \frac{1}{2}]$$

M3(t)

$$\text{Si } x_i < \frac{1}{t^2} - \mu_i$$

$$\tilde{x}_{i-1} = x_{i-1} (x_i + \mu_i)$$

$$\tilde{x}_i = \frac{x_i}{(x_i + \mu_i)^2}$$

$$\tilde{x}_{i+1} = x_{i+1} (x_i + \mu_i)$$

Générer un nouveau μ_i

- Le facteur de décroissance : $\rho = x_i + \mu_i$
- M3(t) modélise et analyse l'algorithme LLL(t) (également pour $t = 1$)
- Le trou du système M2(t, μ) (condition d'arrêt) est

$(x_1, \dots, x_{d-1}), (\mu_1, \dots, \mu_{d-1})$ tels que

$$\forall i = 1, \dots, d-1, \quad x_i \in \left[\frac{1}{t^2} - \mu_i, +\infty \right[.$$

C'est un système dynamique probabiliste dans $\mathbb{R}^{d-1}$, à trou, chaque transformation ayant un point fixe attractif

Modèle M3 : résultats

Aucun !

Conjecture d'un résultat

Posons $K(\mathbf{x})$ le nombre d'itérations sur l'entrée $\mathbf{x} = (x_1, \dots, x_{d-1})$

Posons P_0 le potentiel tel que $P(\mathbf{x}) \geq P_0 \Rightarrow \mathbf{x}$ est "presque réduit"

Considérons une suite i.i.d. de variables aléatoires $(\mu_i = \nu_i^2)_i$ telle que ν_i suit la même loi uniforme sur $[-\frac{1}{2}, \frac{1}{2}]$.

Considérons le temps d'arrêt $T(\mathbf{x})$ défini comme le plus petit k tel que

$$P_0 \prod_{i=1}^k \mu_i < P(\mathbf{x}).$$

Conjecture : $\mathbb{E}[K] \sim \mathbb{E}[T]$

Conjecture d'un résultat

Posons $K(\mathbf{x})$ le nombre d'itérations sur l'entrée $\mathbf{x} = (x_1, \dots, x_{d-1})$

Posons P_0 le potentiel tel que $P(\mathbf{x}) \geq P_0 \Rightarrow \mathbf{x}$ est “presque réduit”

Considérons une suite i.i.d. de variables aléatoires $(\mu_i = \nu_i^2)_i$ telle que ν_i suit la même loi uniforme sur $[-\frac{1}{2}, \frac{1}{2}]$.

Considérons le temps d'arrêt $T(\mathbf{x})$ défini comme le plus petit k tel que

$$P_0 \prod_{i=1}^k \mu_i < P(\mathbf{x}).$$

Conjecture : $\mathbb{E}[K] \sim \mathbb{E}[T]$

Conjecture d'un résultat

Posons $K(\mathbf{x})$ le nombre d'itérations sur l'entrée $\mathbf{x} = (x_1, \dots, x_{d-1})$

Posons P_0 le potentiel tel que $P(\mathbf{x}) \geq P_0 \Rightarrow \mathbf{x}$ est “presque réduit”

Considérons une suite i.i.d. de variables aléatoires $(\mu_i = \nu_i^2)_i$ telle que ν_i suit la même loi uniforme sur $[-\frac{1}{2}, \frac{1}{2}]$.

Considérons le temps d'arrêt $T(\mathbf{x})$ défini comme le plus petit k tel que

$$P_0 \prod_{i=1}^k \mu_i < P(\mathbf{x}).$$

Conjecture : $\mathbb{E}[K] \sim \mathbb{E}[T]$

Conjecture d'un résultat

Posons $K(\mathbf{x})$ le nombre d'itérations sur l'entrée $\mathbf{x} = (x_1, \dots, x_{d-1})$

Posons P_0 le potentiel tel que $P(\mathbf{x}) \geq P_0 \Rightarrow \mathbf{x}$ est "presque réduit"

Considérons une suite i.i.d. de variables aléatoires $(\mu_i = \nu_i^2)_i$ telle que ν_i suit la même loi uniforme sur $[-\frac{1}{2}, \frac{1}{2}]$.

Considérons le temps d'arrêt $T(\mathbf{x})$ défini comme le plus petit k tel que

$$P_0 \prod_{i=1}^k \mu_i < P(\mathbf{x}).$$

Conjecture : $\mathbb{E}[K] \sim \mathbb{E}[T]$

Conjecture d'un résultat

Posons $K(\mathbf{x})$ le nombre d'itérations sur l'entrée $\mathbf{x} = (x_1, \dots, x_{d-1})$

Posons P_0 le potentiel tel que $P(\mathbf{x}) \geq P_0 \Rightarrow \mathbf{x}$ est "presque réduit"

Considérons une suite i.i.d. de variables aléatoires $(\mu_i = \nu_i^2)_i$ telle que ν_i suit la même loi uniforme sur $[-\frac{1}{2}, \frac{1}{2}]$.

Considérons le temps d'arrêt $T(\mathbf{x})$ défini comme le plus petit k tel que

$$P_0 \prod_{i=1}^k \mu_i < P(\mathbf{x}).$$

Conjecture : $\mathbb{E}[K] \sim \mathbb{E}[T]$

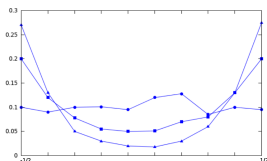
- 1 Algorithme LLL
 - Réduction des réseaux
 - Exemple d'application
 - Algorithme LLL
- 2 Modélisations de l'algorithme LLL et de ses entrées
 - Modèles d'exécution (part 1)
 - Modèles d'entrée
- 3 Résultats obtenus dans les différents modèles
 - Modèle M1 : cfg/tas de sable
 - Modèle M1 et modèles d'entrée
 - Modèle M2 : système dynamique de $\mathbb{R}^d$
 - Modèle M3 : systèmes dynamiques probabilistes de $\mathbb{R}^d$
- 4 Conclusion

Conclusion

- Une classe de modèles simplifiés pour l'exécution de l'algorithme :
 - du plus simple : cfg
 - au plus compliqué : l'algorithme LLL
- Étude d'un modèle semi-simplifié d'exécution :
 - une analyse totale pour $d = 3$
où l'analyse du véritable algorithme LLL est déjà mal comprise
 - analyse partielle en dimension générale.
- Modélisation des familles de réseaux cryptographiques, dans le cadre d'un cfg
 - “tas plein” : dit réseaux d'Ajtai
 - “uni-tas” : sac-à-dos ou réseaux NTRU
 - “tas à trous” : de Coppersmith :

Perspectives

- Modélisation d'autres algorithmes de réduction de réseaux (DeepLLL)
- Pour le modèle M2, prouver la conjecture, qui permettrait d'avoir un modèle simplifié pour l'algorithme LLL associé au paramètre $t = 1$
- Analyser le modèle M3
- Modèle de l'évolution du coefficient sous diagonal (Brigitte?)



Merci pour votre attention !