An overview of lattice reduction algorithms

Damien Stehlé

ENS de Lyon

July 7th 2016

Two main uses of lattice reduction in crypto

Context 1: cryptanalysis of lattice-based cryptosystems.

- NTRU (encryption & signature)
- All cryptosystems based on SIS and LWE and their ring variants

$\Rightarrow$ Huge dims, small entries, very expensive task

Context 2: cryptanalysis of other cryptosystems.

- Variants of RSA (with Coppersmith's method)

$\Rightarrow$ Large dims, huge entries, quite cheap

Two main uses of lattice reduction in crypto

Context 1: cryptanalysis of lattice-based cryptosystems.

- NTRU (encryption & signature)
- All cryptosystems based on SIS and LWE and their ring variants

$\Rightarrow$ Huge dims, small entries, very expensive task

Context 2: cryptanalysis of other cryptosystems.

- Variants of RSA (with Coppersmith's method)

$\Rightarrow$ Large dims, huge entries, much less costly

Two main uses of lattice reduction in crypto

Context 1: cryptanalysis of lattice-based cryptosystems.

- NTRU (encryption & signature)
- All cryptosystems based on SIS and LWE and their ring variants

$\Rightarrow$ Huge dims, small entries, very expensive task

Context 2: cryptanalysis of other cryptosystems.

- Variants of RSA (with Coppersmith's method)

$\Rightarrow$ Large dims, huge entries, much less costly

Two main uses of lattice reduction in crypto

Context 1: cryptanalysis of lattice-based cryptosystems.

- NTRU (encryption & signature)
- All cryptosystems based on SIS and LWE and their ring variants

$\Rightarrow$ Huge dims, small entries, very expensive task

Context 2: cryptanalysis of other cryptosystems.

- Variants of RSA (with Coppersmith's method)

$\Rightarrow$ Large dims, huge entries, much less costly

Roadmap

Goal of this talk

An introductive overview on lattice reduction algorithms

1. **Background on lattices**
2. The lattice reduction framework
3. Strong but slow: BKZ
4. Solving the SIS problem
5. Weak but fast: LLL

Euclidean lattices

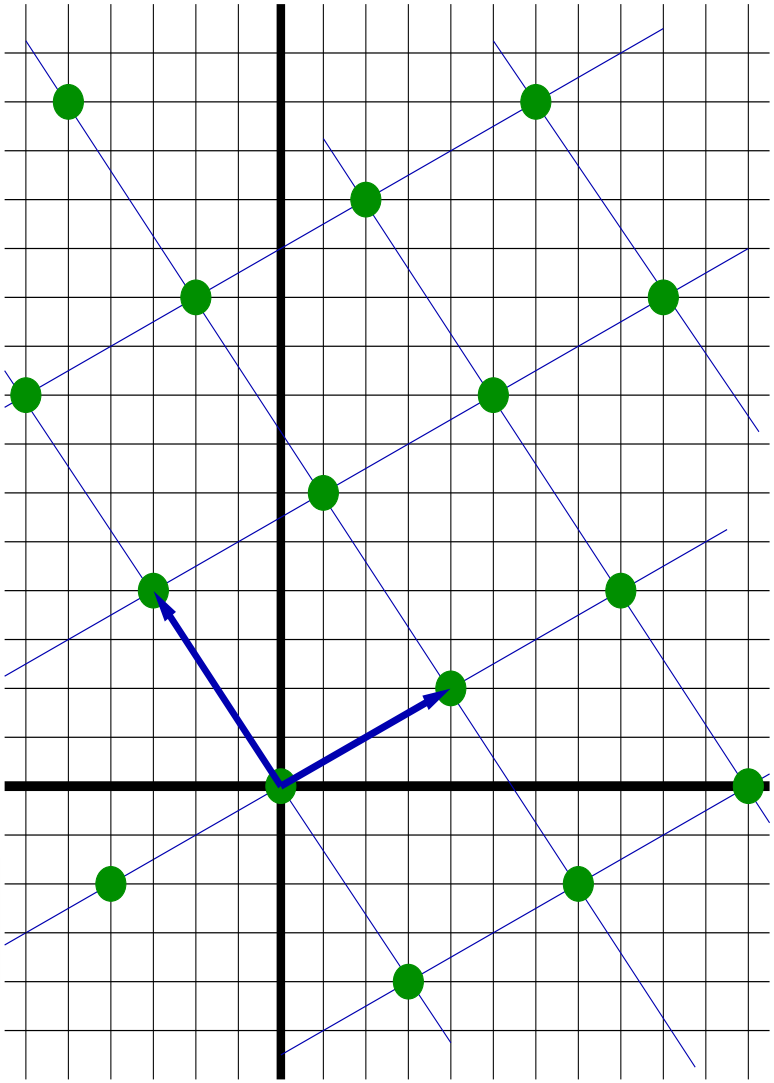
Lattice $\equiv \left\{\sum_{i\leq n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\right\}$,
for linearly indep. $\mathbf{b}_i$'s in $\mathbb{R}^n$,
referred to as **lattice basis**

Bases are **not unique**, but can be obtained from one another by integer transforms of determinant ± 1:

$$\begin{bmatrix} -2 & 1 \\ 10 & 6 \end{bmatrix} = \begin{bmatrix} 4 & -3 \\ 2 & 4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$$

Lattice reduction

Find a short basis, given an arbitrary one



Euclidean lattices

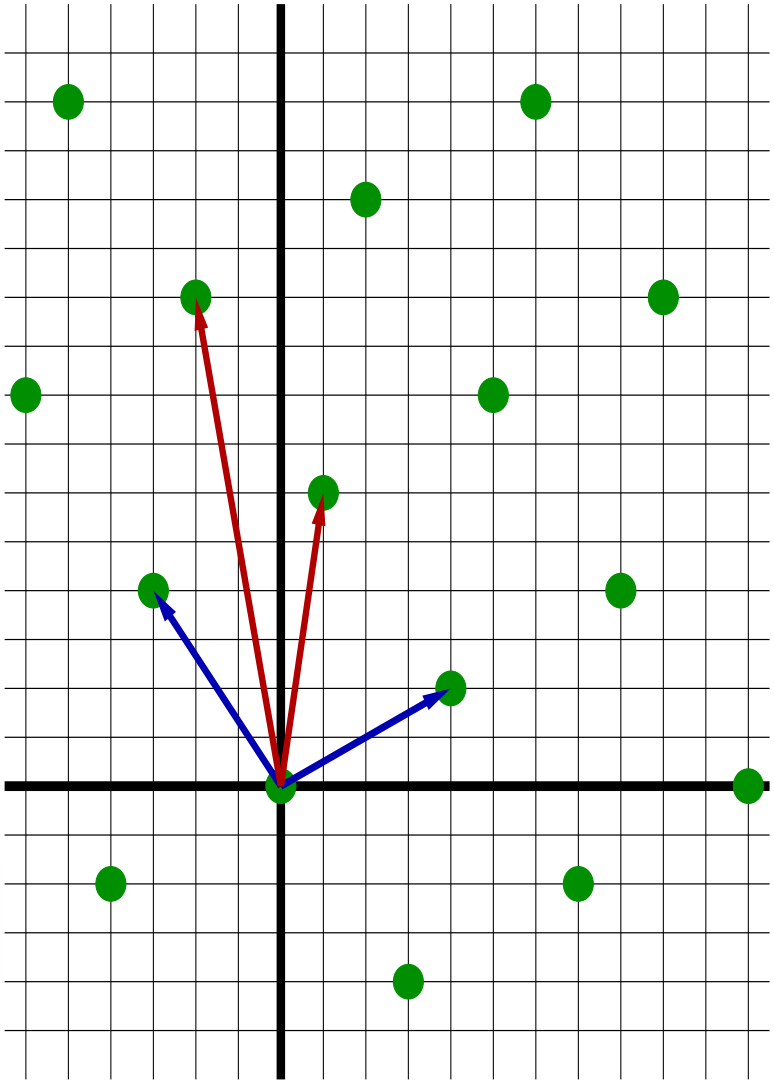
Lattice $\equiv \left\{\sum_{i\leq n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\right\}$,
for linearly indep. $\mathbf{b}_i$'s in $\mathbb{R}^n$,
referred to as **lattice basis**

Bases are **not unique**, but can be obtained from one another by integer transforms of determinant ± 1:

$$\begin{bmatrix} -2 & 1 \\ 10 & 6 \end{bmatrix} = \begin{bmatrix} 4 & -3 \\ 2 & 4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$$

Lattice reduction

Find a short basis, given an arbitrary one

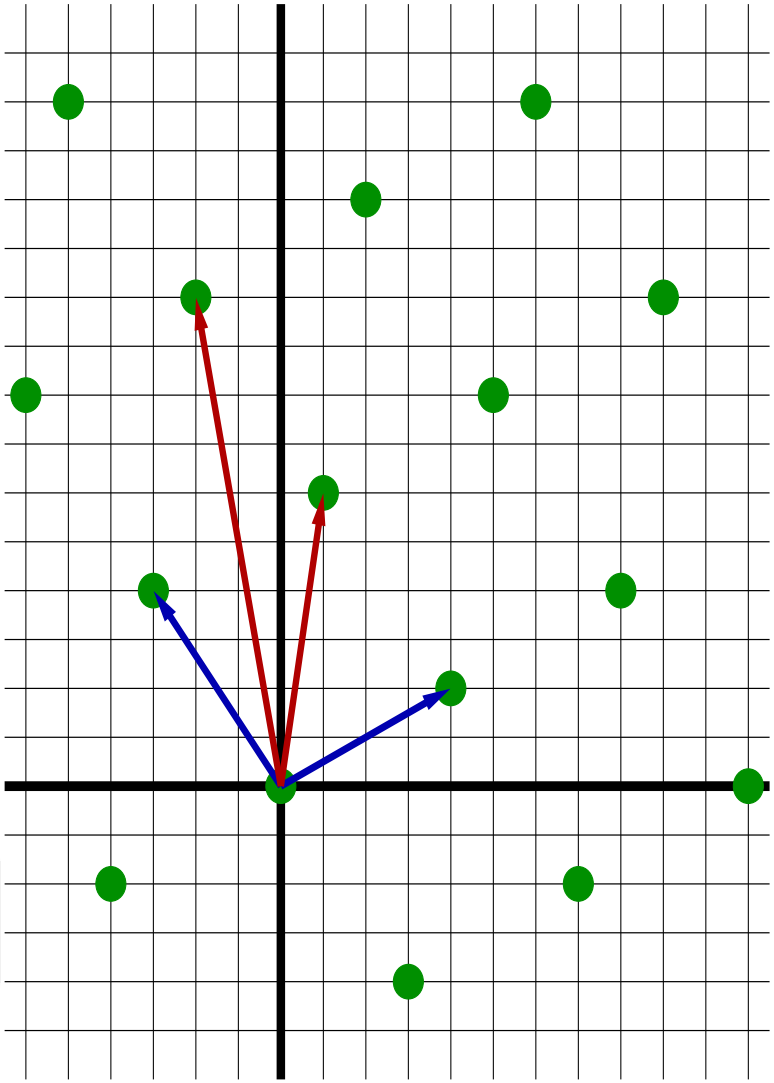


Euclidean lattices

Lattice $\equiv \left\{\sum_{i\leq n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\right\}$,
for linearly indep. $\mathbf{b}_i$'s in $\mathbb{R}^n$,
referred to as **lattice basis**

Bases are **not unique**, but can be obtained from one another by integer transforms of determinant ± 1:

$$\begin{bmatrix} -2 & 1 \\ 10 & 6 \end{bmatrix} = \begin{bmatrix} 4 & -3 \\ 2 & 4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$$



Lattice reduction

Find a short basis, given an arbitrary one

Lattice invariants

Minimum:
$\lambda(L) = \min(\|\mathbf{b}\| : \mathbf{b} \in L \setminus \mathbf{0})$

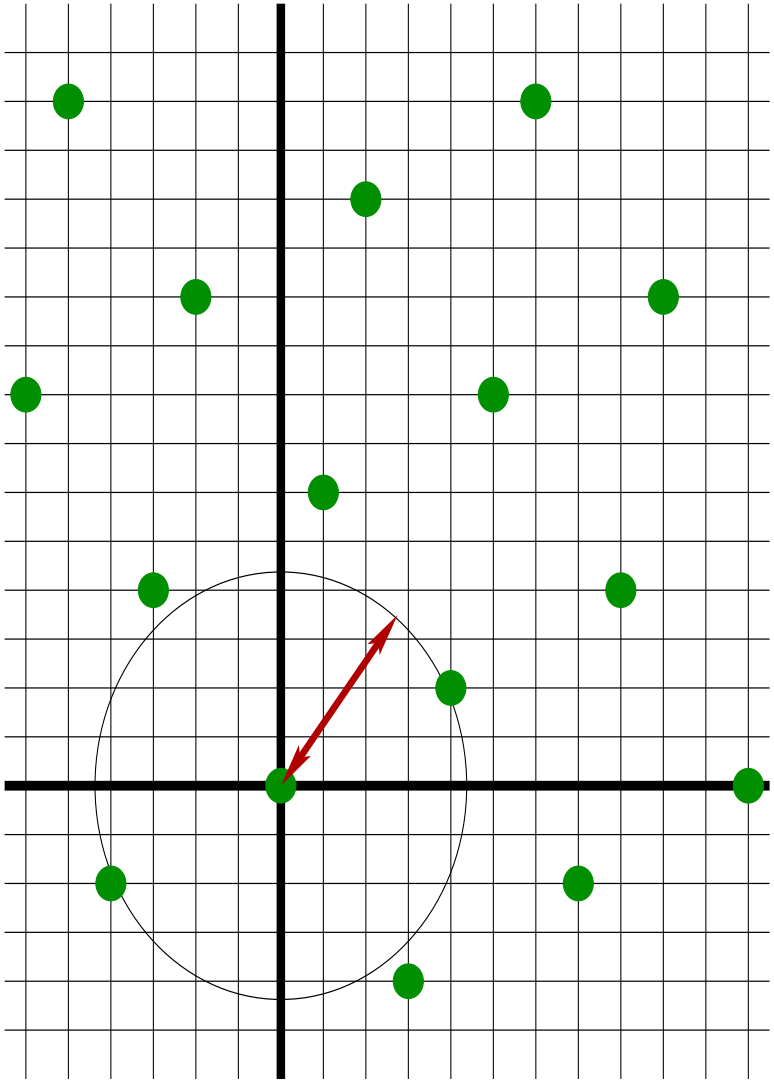
Determinant:
$\det L = |\det(\mathbf{b}_i)_i|$, for any basis

Minkowski theorem

$\lambda(L) \leq \sqrt{n} \cdot (\det L)^{\frac{1}{n}}$, for any L of dim n

Lattice reduction

Find a basis that is short compared to $\lambda(L)$ and/or $(\det L)^{\frac{1}{n}}$



Lattice invariants

Minimum:
$\lambda(L) = \min(\|\mathbf{b}\| : \mathbf{b} \in L \setminus \mathbf{0})$

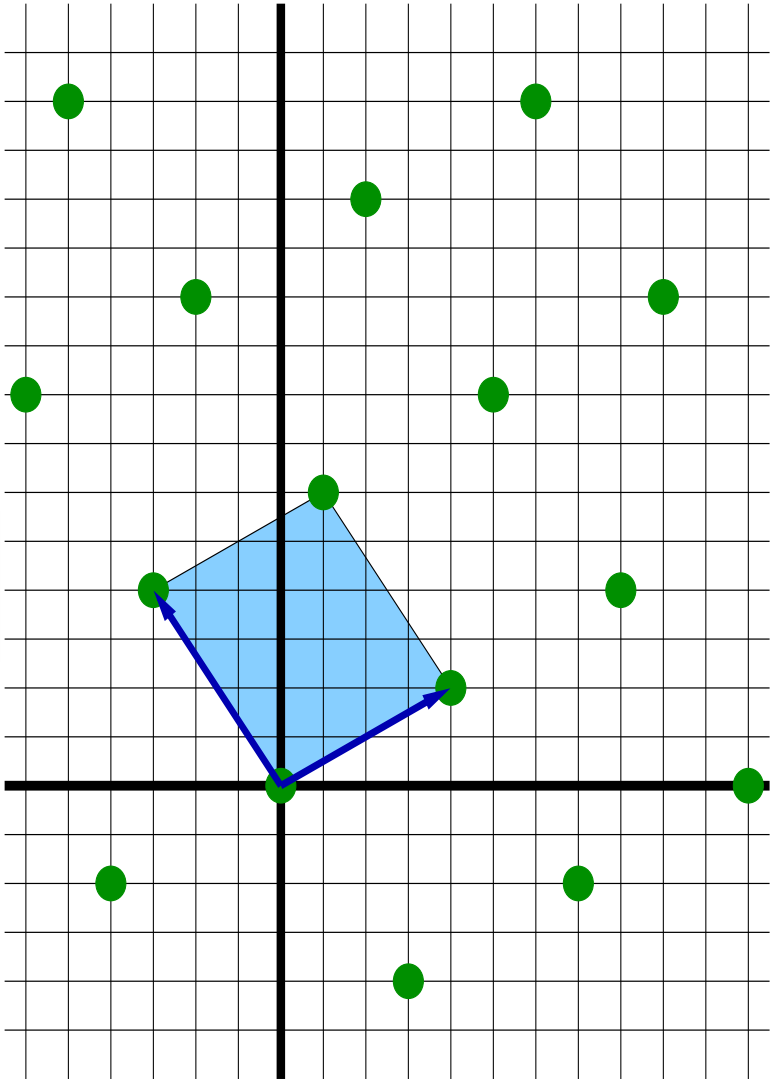
Determinant:
$\det L = |\det(\mathbf{b}_i)_i|$, for any basis

Minkowski theorem

$\lambda(L) \leq \sqrt{n} \cdot (\det L)^{\frac{1}{n}}$, for any L of dim n

Lattice reduction

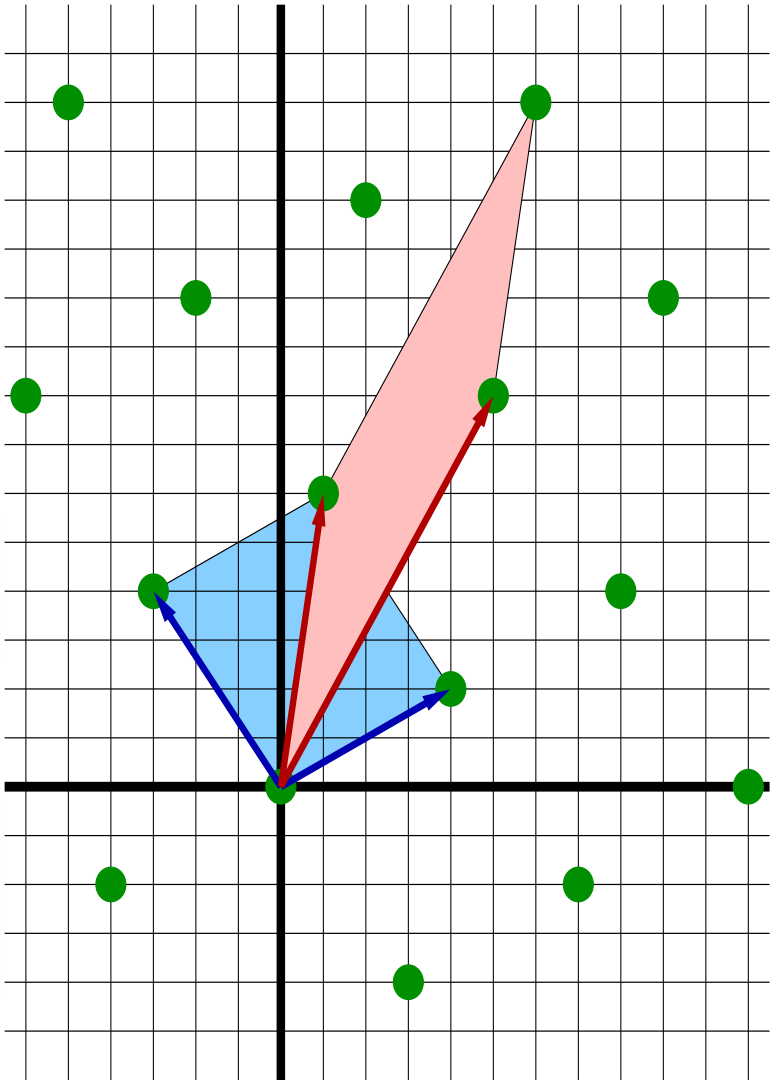
Find a basis that is short compared to $\lambda(L)$ and/or $(\det L)^{\frac{1}{n}}$



Lattice invariants

Minimum:
$\lambda(L) = \min(\|\mathbf{b}\| : \mathbf{b} \in L \setminus \mathbf{0})$

Determinant:
$\det L = |\det(\mathbf{b}_i)_i|$, for any basis



Minkowski theorem

$\lambda(L) \leq \sqrt{n} \cdot (\det L)^{\frac{1}{n}}$, for any L of dim n

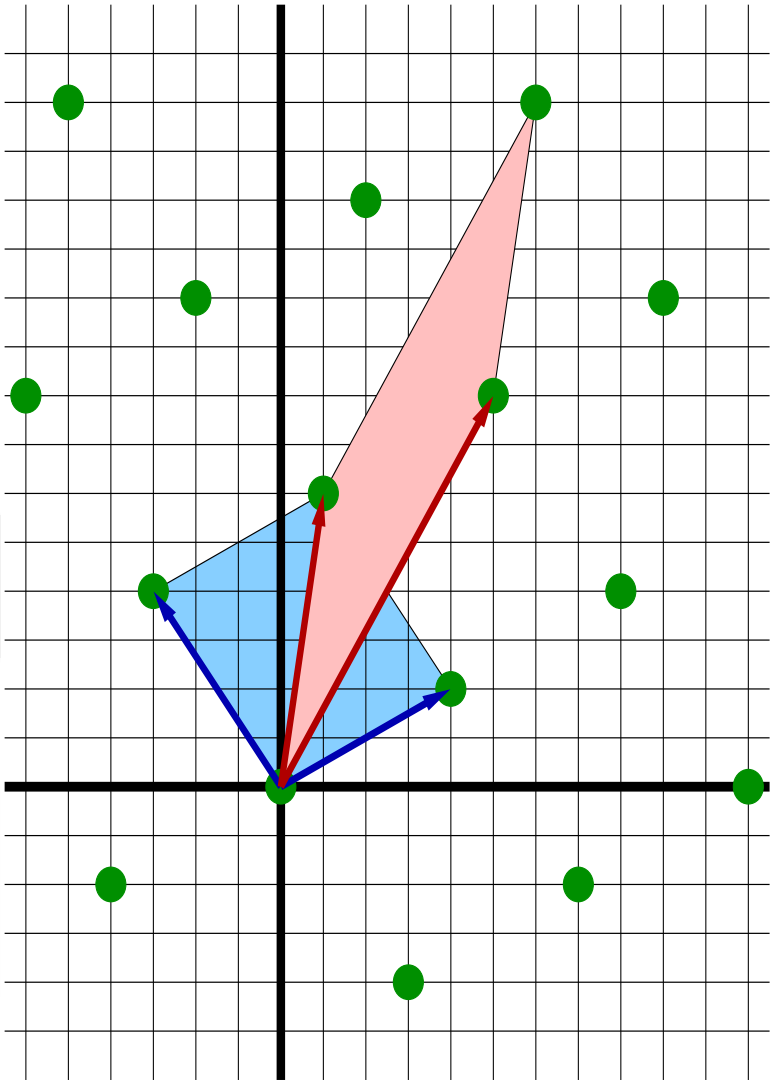
Lattice reduction

Find a basis that is short compared to $\lambda(L)$ and/or $(\det L)^{\frac{1}{n}}$

Lattice invariants

Minimum:
$\lambda(L) = \min(\|\mathbf{b}\| : \mathbf{b} \in L \setminus \mathbf{0})$

Determinant:
$\det L = |\det(\mathbf{b}_i)_i|$, for any basis



Minkowski theorem

$\lambda(L) \leq \sqrt{n} \cdot (\det L)^{\frac{1}{n}}$, for any L of dim n

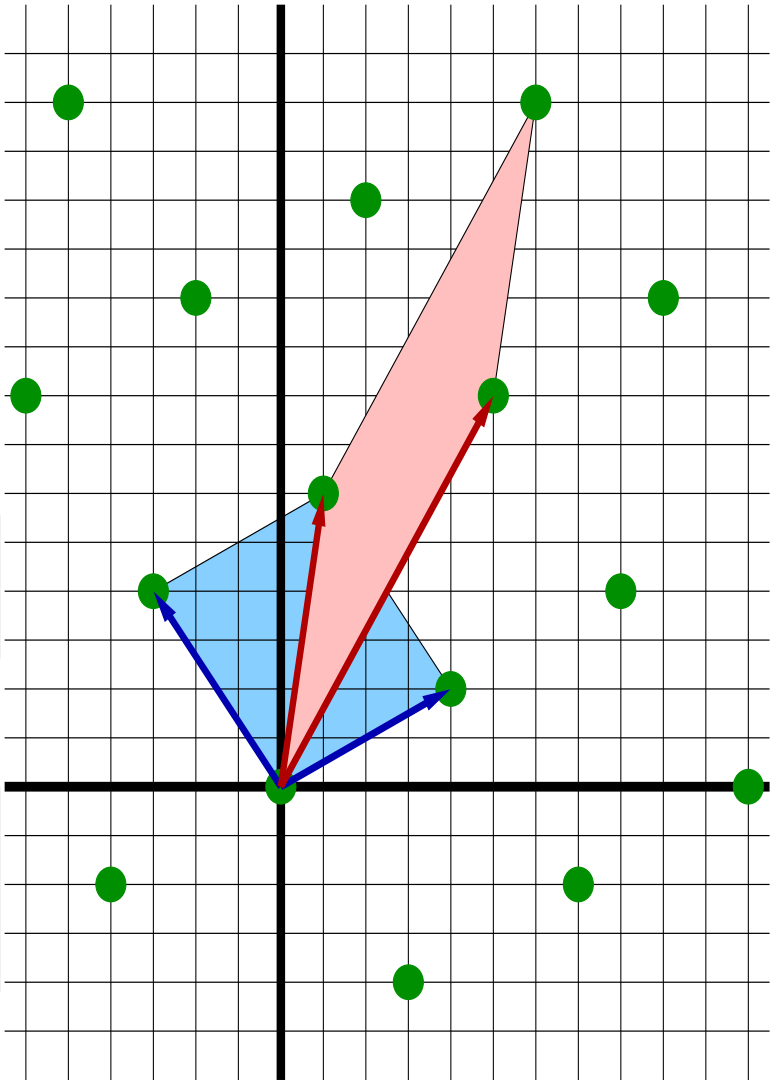
Lattice reduction

Find a basis that is short compared to $\lambda(L)$ and/or $(\det L)^{\frac{1}{n}}$

Lattice invariants

Minimum:
$\lambda(L) = \min(\|\mathbf{b}\| : \mathbf{b} \in L \setminus \mathbf{0})$

Determinant:
$\det L = |\det(\mathbf{b}_i)_i|$, for any basis



Minkowski theorem

$\lambda(L) \leq \sqrt{n} \cdot (\det L)^{\frac{1}{n}}$, for any L of dim n

Lattice reduction

Find a basis that is short compared to $\lambda(L)$ and/or $(\det L)^{\frac{1}{n}}$

Computational problems on lattices

The Shortest Vector Problem: SVP_γ

Given a basis of L, find $\mathbf{b} \in L$ s.t.: $0 < \|\mathbf{b}\| \leq \gamma \cdot \lambda(L)$

Effective Minkowski theorem: HSVP_γ

Given a basis of L, find $\mathbf{b} \in L$ s.t.: $0 < \|\mathbf{b}\| \leq \gamma \cdot (\det L)^{\frac{1}{n}}$

Many other problems: BDD_γ, CVP_γ, uSVP_γ, SIVP_γ, etc.

Computational problems on lattices

The Shortest Vector Problem: SVP_γ

Given a basis of L, find $\mathbf{b} \in L$ s.t.: $0 < \|\mathbf{b}\| \leq \gamma \cdot \lambda(L)$

Effective Minkowski theorem: HSVP_γ

Given a basis of L, find $\mathbf{b} \in L$ s.t.: $0 < \|\mathbf{b}\| \leq \gamma \cdot (\det L)^{\frac{1}{n}}$

Many other problems: BDD_γ, CVP_γ, uSVP_γ, SIVP_γ, etc.

Computational problems on lattices

The Shortest Vector Problem: SVP_γ

Given a basis of L, find $\mathbf{b} \in L$ s.t.: $0 < \|\mathbf{b}\| \leq \gamma \cdot \lambda(L)$

Effective Minkowski theorem: HSVP_γ

Given a basis of L, find $\mathbf{b} \in L$ s.t.: $0 < \|\mathbf{b}\| \leq \gamma \cdot (\det L)^{\frac{1}{n}}$

Many other problems: BDD_γ, CVP_γ, uSVP_γ, SIVP_γ, etc.

The bad news...

- SVP_γ is NP-hard for $\gamma = O(1)$ (under random. red.) [Ajt98]
- SVP_γ, HSVP_γ, BDD_γ... in P for $\gamma = 2^{\Omega\left(n\frac{\log\log n}{\log n}\right)}$ [Sch87,MiVo10]

When $\gamma \geq n^{\Omega(1)}$, the cost of the best known algorithm is:

$$\mathcal{P}oly(m, \log \|B\|) \cdot \left(1 + \frac{n}{\log \gamma}\right)^{O\left(1 + \frac{n}{\log \gamma}\right)}.$$

For $\gamma \leq \mathcal{P}oly(n)$, the cost is $\mathcal{P}oly(m, \log \|B\|) \cdot 2^{O(n)}$.

The bad news...

- SVP_γ is NP-hard for $\gamma = O(1)$ (under random. red.) [Ajt98]
- SVP_γ, HSVP_γ, BDD_γ... in P for $\gamma = 2^{\Omega\left(n\frac{\log\log n}{\log n}\right)}$ [Sch87,MiVo10]

When $\gamma \geq n^{\Omega(1)}$, the cost of the best known algorithm is:

$$\mathcal{P}oly(m, \log \|B\|) \cdot \left(1 + \frac{n}{\log \gamma}\right)^{O\left(1 + \frac{n}{\log \gamma}\right)}.$$

For $\gamma \leq \mathcal{P}oly(n)$, the cost is $\mathcal{P}oly(m, \log \|B\|) \cdot 2^{O(n)}$.

Roadmap

1. Background on lattices
2. **The lattice reduction framework**
3. Strong but slow: BKZ
4. Solving the SIS problem
5. Weak but fast: LLL

From n^2 to $n(n+1)/2$ variables: QR-factorisation

Goal of lattice reduction

Given $B \in \mathbb{R}^{n \times n}$ full-rank, find $U \in \mathrm{GL}_n(\mathbb{Z})$ s.t. $B \cdot U$ has small coeffs, i.e., its columns have small euclidean norms

As $\|\cdot\|$ is invariant under rotations, we may work on triangular matrices:

$$B = \underbrace{Q}_{\text{orthogonal}} \cdot \underbrace{R}_{\text{up-triangular}} \qquad \text{(QR-factorisation)}$$

- r_{11} is the norm of $\mathbf{b}_1$
- r_{ii} is the norm of the proj. of $\mathbf{b}_i$ orthogonally to $(\mathbf{b}_j)_{j<i}$
- This is equivalent to Gram-Schmidt Orthogonalisation

From n^2 to $n(n+1)/2$ variables: QR-factorisation

Goal of lattice reduction

Given $B \in \mathbb{R}^{n \times n}$ full-rank, find $U \in \mathrm{GL}_n(\mathbb{Z})$ s.t. $B \cdot U$ has small coeffs, i.e., its columns have small euclidean norms

As $\|\cdot\|$ is invariant under rotations, we may work on triangular matrices:

$$B = \underbrace{Q}_{\text{orthogonal}} \cdot \underbrace{R}_{\text{up-triangular}} \qquad \textbf{(QR-factorisation)}$$

- r_{11} is the norm of $\mathbf{b}_1$
- r_{ii} is the norm of the proj. of $\mathbf{b}_i$ orthogonally to $(\mathbf{b}_j)_{j<i}$
- This is equivalent to Gram-Schmidt Orthogonalisation

From n^2 to $n(n+1)/2$ variables: QR-factorisation

Goal of lattice reduction

Given $B \in \mathbb{R}^{n \times n}$ full-rank, find $U \in \mathrm{GL}_n(\mathbb{Z})$ s.t. $B \cdot U$ has small coeffs, i.e., its columns have small euclidean norms

As $\|\cdot\|$ is invariant under rotations, we may work on triangular matrices:

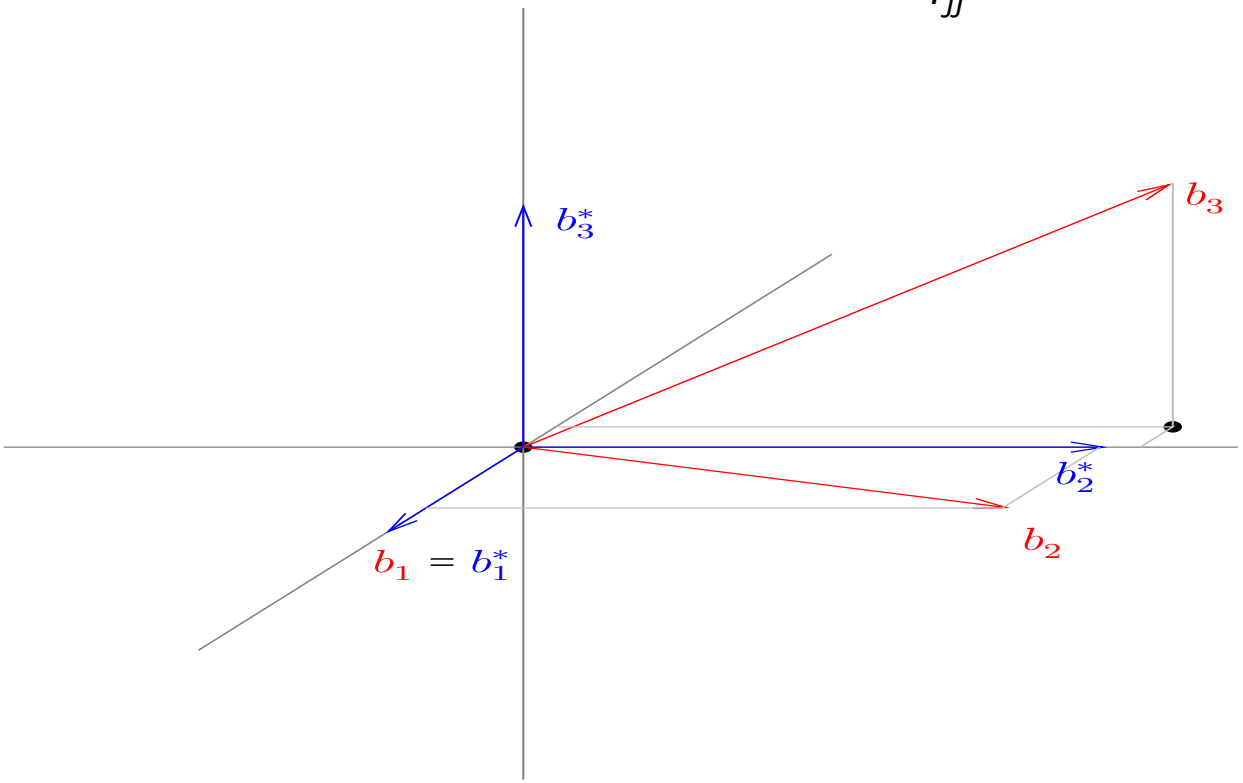
$$B = \underbrace{Q}_{\text{orthogonal}} \cdot \underbrace{R}_{\text{up-triangular}} \qquad \textbf{(QR-factorisation)}$$

- r_{11} is the norm of $\mathbf{b}_1$
- r_{ii} is the norm of the proj. of $\mathbf{b}_i$ orthogonally to $(\mathbf{b}_j)_{j<i}$
- This is equivalent to Gram-Schmidt Orthogonalisation

QR-factorisation and GSO

QR is equivalent to Gram-Schmidt Orthogonalisation:

- For all i, $\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j<i} \mu_{ij} \mathbf{b}_j^*$ is the projection of $\mathbf{b}_i$ orthogonally to $\mathrm{Span}_{\mathbb{R}}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})$.
- We have $\|\mathbf{b}_i^*\| = r_{ii}$ and $\mu_{ij} = \frac{r_{ji}}{r_{jj}}$ for $i > j$.



From $n(n+1)/2$ to n variables: size-reduction

Goal of lattice reduction

Given $R \in \mathbb{R}^{n \times n}$ up-triangular, find $U \in \mathrm{GL}_n(\mathbb{Z})$
s.t. $R \cdot U$ has a small R-factor

$$\begin{bmatrix} \ddots & \vdots & \dots & \vdots & \dots \\ & r_{ii} & \dots & r_{ij} & \dots \\ & & \ddots & \vdots & \dots \\ & & & r_{jj} & \dots \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & & & & \\ & 1 & & -\lfloor \frac{r_{ij}}{r_{ii}} \rceil & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \ddots \end{bmatrix} \Rightarrow \left| r_{ij}^{(new)} \right| \leq \frac{|r_{ii}|}{2}$$

But other coefficients of R may have changed:
If they were small, they may not be small anymore...

From $n(n+1)/2$ to n variables: size-reduction

Goal of lattice reduction

Given $R \in \mathbb{R}^{n \times n}$ up-triangular, find $U \in \mathrm{GL}_n(\mathbb{Z})$
s.t. $R \cdot U$ has a small R-factor

$$\begin{bmatrix} \ddots & \vdots & \dots & \vdots & \dots \\ & r_{ii} & \dots & r_{ij} & \dots \\ & & \ddots & \vdots & \dots \\ & & & r_{jj} & \dots \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & & & & \\ & 1 & & -\lfloor \frac{r_{ij}}{r_{ii}} \rceil & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \ddots \end{bmatrix} \Rightarrow \left| r_{ij}^{(new)} \right| \leq \frac{|r_{ii}|}{2}$$

But other coefficients of R may have changed:
If they were small, they may not be small anymore...

From $n(n+1)/2$ to n variables: size-reduction

Goal of lattice reduction

Given $R \in \mathbb{R}^{n \times n}$ up-triangular, find $U \in \mathrm{GL}_n(\mathbb{Z})$
s.t. $R \cdot U$ has a small R-factor

$$\begin{bmatrix} \ddots & \vdots & \dots & \vdots & \dots \\ & r_{ii} & \dots & r_{ij} & \dots \\ & & \ddots & \vdots & \dots \\ & & & r_{jj} & \dots \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & & & & \\ & 1 & & -\lfloor \frac{r_{ij}}{r_{ii}} \rceil & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \ddots \end{bmatrix} \Rightarrow \left| r_{ij}^{(new)} \right| \leq \frac{|r_{ii}|}{2}$$

But other coefficients of R may have changed:
If they were small, they may not be small anymore...

From $n(n+1)/2$ to n variables: size-reduction

Goal of lattice reduction

Given $R \in \mathbb{R}^{n \times n}$ up-triangular, find $U \in \mathrm{GL}_n(\mathbb{Z})$
s.t. $R \cdot U$ has a small R-factor

$$\begin{bmatrix} \ddots & \vdots & \dots & \vdots & \dots \\ & r_{ii} & \dots & r_{ij} & \dots \\ & & \ddots & \vdots & \dots \\ & & & r_{jj} & \dots \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & & & & \\ & 1 & & -\lfloor \frac{r_{ij}}{r_{ii}} \rceil & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \ddots \end{bmatrix} \Rightarrow \left| r_{ij}^{(new)} \right| \leq \frac{|r_{ii}|}{2}$$

But other coefficients of R may have changed:
If they were small, they may not be small anymore...

From $n(n+1)/2$ to n variables: size-reduction

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

This modifies the top of the j-th column.
$\Rightarrow$ Proceed from bottom to top.

Size-reduced basis

A basis is said size-reduced if $|r_{ij}| \leq r_{ii}/2$ for all $i < j$

- Size-reduction grants control of the off-diagonal coeffs
- It can be performed with $\mathcal{P}oly(n, \log \|B\|)$ cost

From $n(n+1)/2$ to n variables: size-reduction

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

This modifies the top of the j-th column.
$\Rightarrow$ Proceed from bottom to top.

Size-reduced basis

A basis is said size-reduced if $|r_{ij}| \leq r_{ii}/2$ for all $i < j$

- Size-reduction grants control of the off-diagonal coeffs
- It can be performed with $\mathcal{P}oly(n, \log \|B\|)$ cost

From $n(n+1)/2$ to n variables: size-reduction

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

This modifies the top of the j-th column.
$\Rightarrow$ Proceed from bottom to top.

Size-reduced basis

A basis is said size-reduced if $|r_{ij}| \leq r_{ii}/2$ for all $i < j$

- Size-reduction grants control of the off-diagonal coeffs
- It can be performed with $\mathcal{P}oly(n, \log \|B\|)$ cost

Where are we now?

Goal of lattice reduction

Given $R \in \mathbb{R}^{n \times n}$ up-triangular, find $U \in \mathrm{GL}_n(\mathbb{Z})$
s.t. the R-factor of $R \cdot U$ has small diagonal coeffs

What does it mean, as the product of the r_{ii}'s is constant?
We want to

- make the first r_{ii}'s small,
- make the r_{ii}'s balanced, or preventing them from decreasing fast

Where are we now?

Goal of lattice reduction

Given $R \in \mathbb{R}^{n \times n}$ up-triangular, find $U \in \mathrm{GL}_n(\mathbb{Z})$
s.t. the R-factor of $R \cdot U$ has small diagonal coeffs

What does it mean, as the product of the r_{ii}'s is constant?
We want to

- make the first r_{ii}'s small,
- make the r_{ii}'s balanced, or preventing them from decreasing fast

The best we can do: HKZ

HKZ-reduction

R up-triangular is HKZ-reduced if

- $r_{11} = \lambda(L)$ with $L = \sum_i \mathbb{Z}\mathbf{r}_i$
- and $(r_{ij})_{i,j>1}$ is HKZ-reduced

In the worst case, we have, for all $i \leq n$:

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

Fixing r_{nn} fixes the other r_{ii}'s.
As this is all multiplicative, we use $x_i = \log r_{ii}$ instead.

The best we can do: HKZ

HKZ-reduction

R up-triangular is HKZ-reduced if

- $r_{11} = \lambda(L)$ with $L = \sum_i \mathbb{Z}\mathbf{r}_i$
- and $(r_{ij})_{i,j>1}$ is HKZ-reduced

In the worst case, we have, for all $i \leq n$:

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

Fixing r_{nn} fixes the other r_{ii}'s.
As this is all multiplicative, we use $x_i = \log r_{ii}$ instead.

The best we can do: HKZ

HKZ-reduction

R up-triangular is HKZ-reduced if

- $r_{11} = \lambda(L)$ with $L = \sum_i \mathbb{Z}\mathbf{r}_i$
- and $(r_{ij})_{i,j>1}$ is HKZ-reduced

The best we can do: HKZ

HKZ-reduction

R up-triangular is HKZ-reduced if

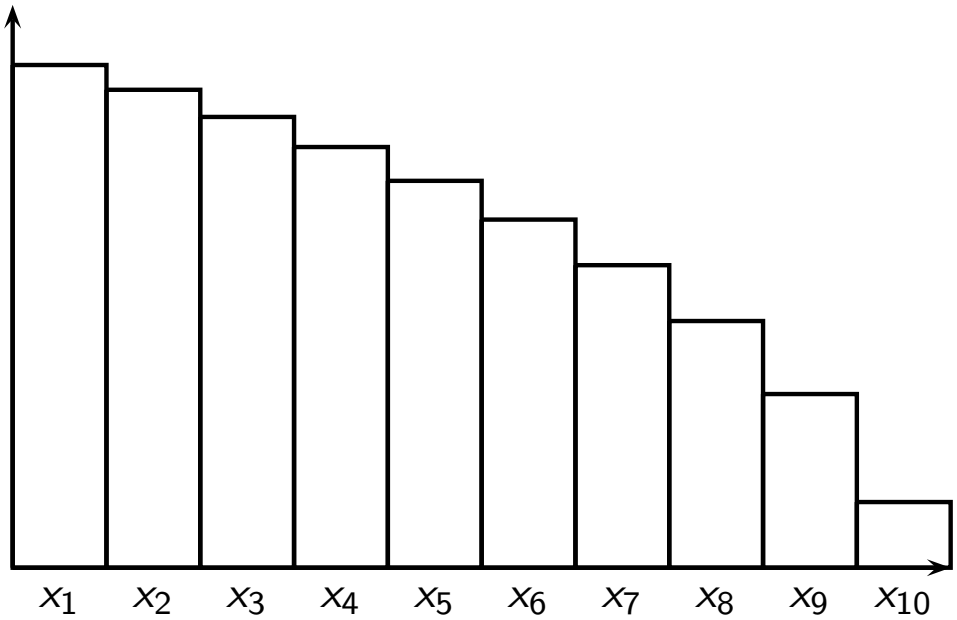
- $r_{11} = \lambda(L)$ with $L = \sum_i \mathbb{Z}\mathbf{r}_i$
- and $(r_{ij})_{i,j>1}$ is HKZ-reduced

Worst-case HKZ profile:

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

Cost of HKZ:

- Computationally eq. to SVP
- Time & space $2^{O(n)}$



The best we can do: HKZ

HKZ-reduction

R up-triangular is HKZ-reduced if

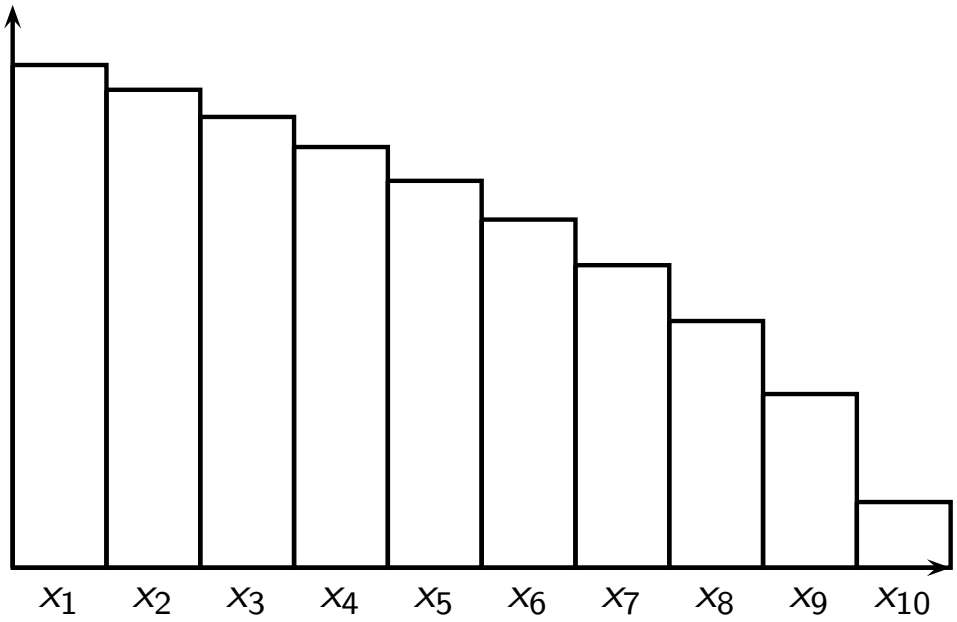
- $r_{11} = \lambda(L)$ with $L = \sum_i \mathbb{Z}\mathbf{r}_i$
- and $(r_{ij})_{i,j>1}$ is HKZ-reduced

Worst-case HKZ profile:

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

Cost of HKZ:

- Computationally eq. to SVP
- Time & space $2^{O(n)}$



Lattice reduction: the rules of the game

Goal of lattice reduction

Given $R \in \mathbb{R}^{n \times n}$ up-triangular, find $U \in \mathrm{GL}_n(\mathbb{Z})$
s.t. the R-factor of $R \cdot U$ has small diagonal coeffs

HKZ too costly... What can we do?

1. Swap two consecutive vectors s.t. $r_{i+1,i+1} \ll r_{i,i}$ [LLL82]
2. Balance the diag. coeffs locally by applying lattice reduction (e.g., HKZ) to a diag. submatrix of R [Sch87]

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

Lattice reduction: the rules of the game

Goal of lattice reduction

Given $R \in \mathbb{R}^{n \times n}$ up-triangular, find $U \in \mathrm{GL}_n(\mathbb{Z})$
s.t. the R-factor of $R \cdot U$ has small diagonal coeffs

HKZ too costly... What can we do?

1. Swap two consecutive vectors s.t. $r_{i+1,i+1} \ll r_{i,i}$ [LLL82]
2. Balance the diag. coeffs locally by applying lattice reduction (e.g., HKZ) to a diag. submatrix of R [Sch87]

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

Lattice reduction: the rules of the game

Goal of lattice reduction

Given $R \in \mathbb{R}^{n \times n}$ up-triangular, find $U \in \mathrm{GL}_n(\mathbb{Z})$
s.t. the R-factor of $R \cdot U$ has small diagonal coeffs

HKZ too costly... What can we do?

1. Swap two consecutive vectors s.t. $r_{i+1,i+1} \ll r_{i,i}$ [LLL82]
2. Balance the diag. coeffs locally by applying lattice reduction (e.g., HKZ) to a diag. submatrix of R [Sch87]

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

Roadmap

1. Background on lattices
2. The lattice reduction framework
3. **Strong but slow: BKZ**
4. Solving the SIS problem
5. Weak but fast: LLL

BKZ: A global reduction approach

[Sch87,ScEu91]: Do HKZ on k-dim. diagonal submatrices of R

BKZ$_k$, simplified version

Input: $R \in \mathbb{R}^{n \times n}$ up-triangular
Repeat ... times
  For i from 1 to $n - k + 1$ do
    HKZ-reduce the k-dim sub-matrix of R starting at r_{ii}
    Update the R-factor and size-reduce it

- How many iterations?
- What is the output quality?

BKZ: A global reduction approach

[Sch87,ScEu91]: Do HKZ on k-dim. diagonal submatrices of R

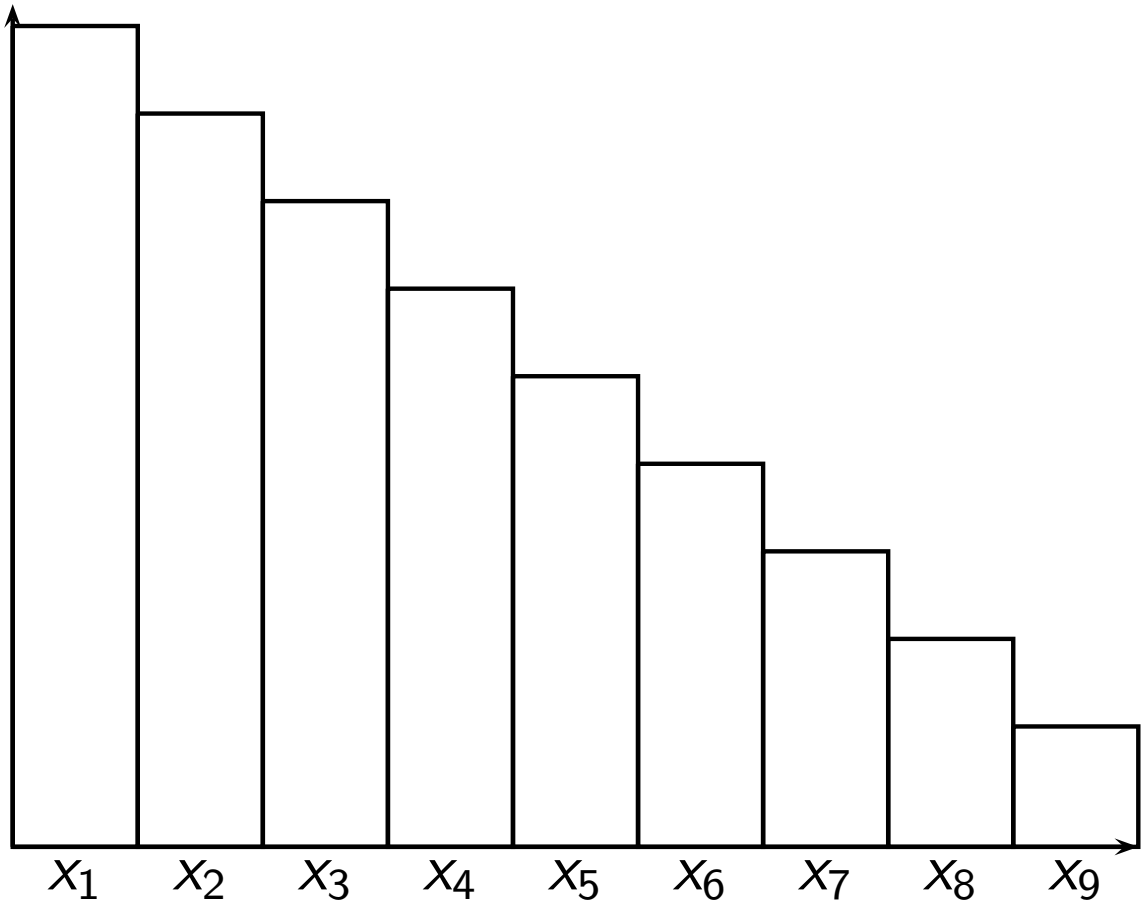
BKZ_k, simplified version

Input: $R \in \mathbb{R}^{n \times n}$ up-triangular
Repeat ... times
  For i from 1 to $n - k + 1$ do
    HKZ-reduce the k-dim sub-matrix of R starting at r_{ii}
    Update the R-factor and size-reduce it

- How many iterations?
- What is the output quality?

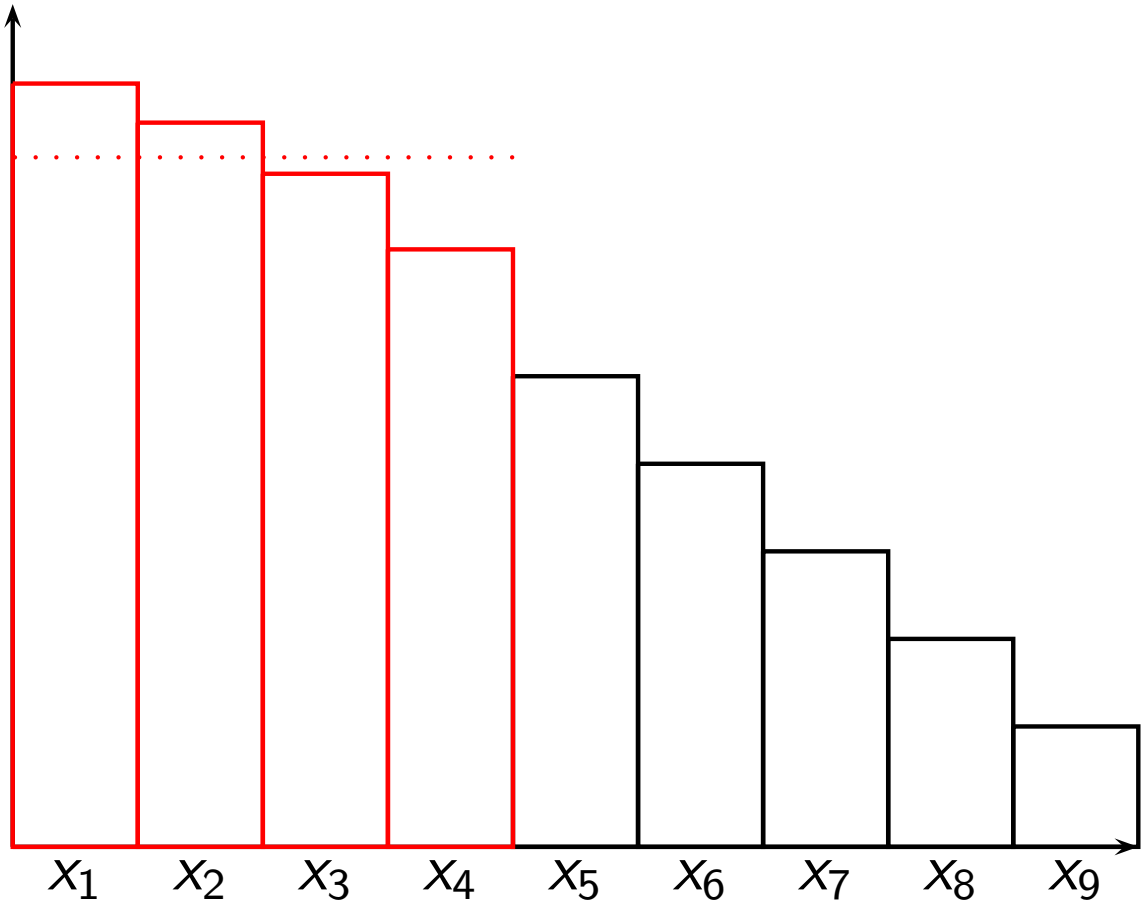
A sandpile model for BKZ_k [HaPuSt11]

Regularity assumption: Each HKZ-reduction gives a worst-case profile



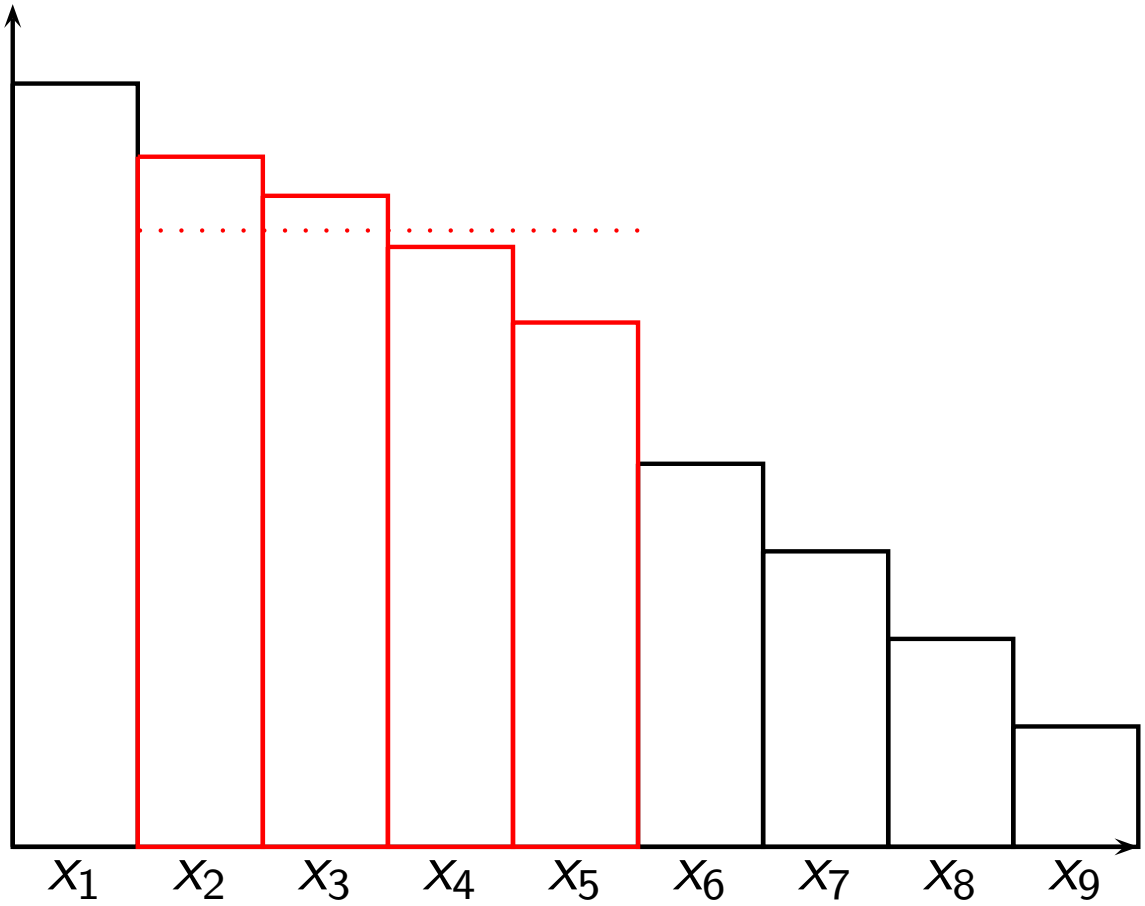
A sandpile model for BKZ_k [HaPuSt11]

Regularity assumption: Each HKZ-reduction gives a worst-case profile



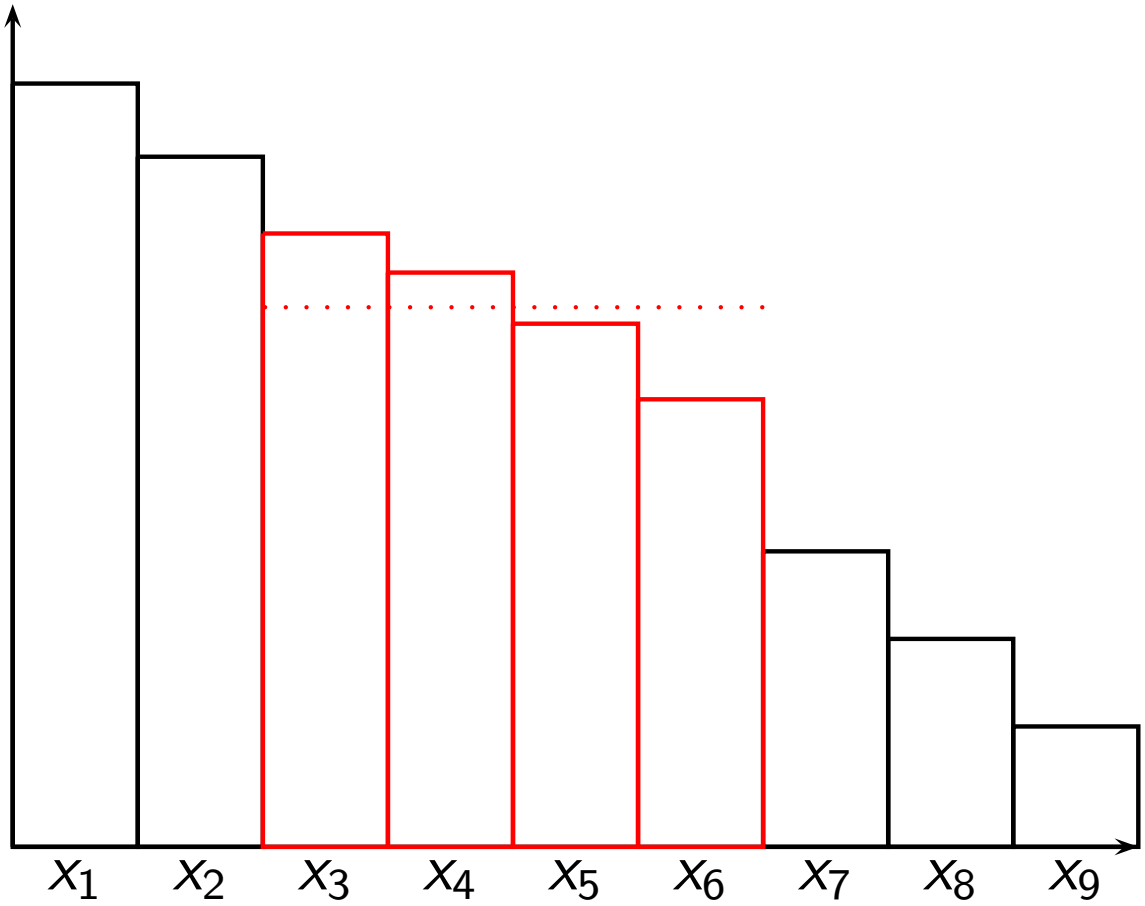
A sandpile model for BKZ_k [HaPuSt11]

Regularity assumption: Each HKZ-reduction gives a worst-case profile



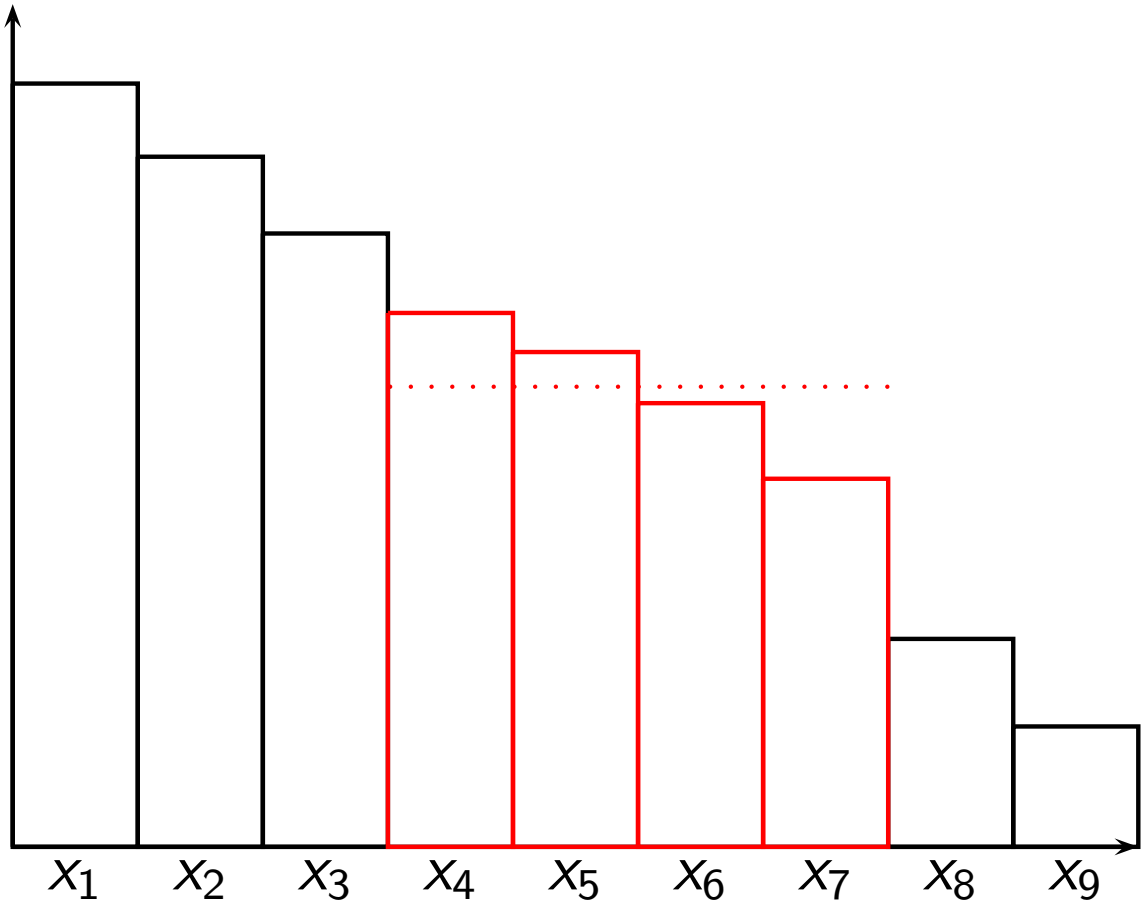
A sandpile model for BKZ_k [HaPuSt11]

Regularity assumption: Each HKZ-reduction gives a worst-case profile



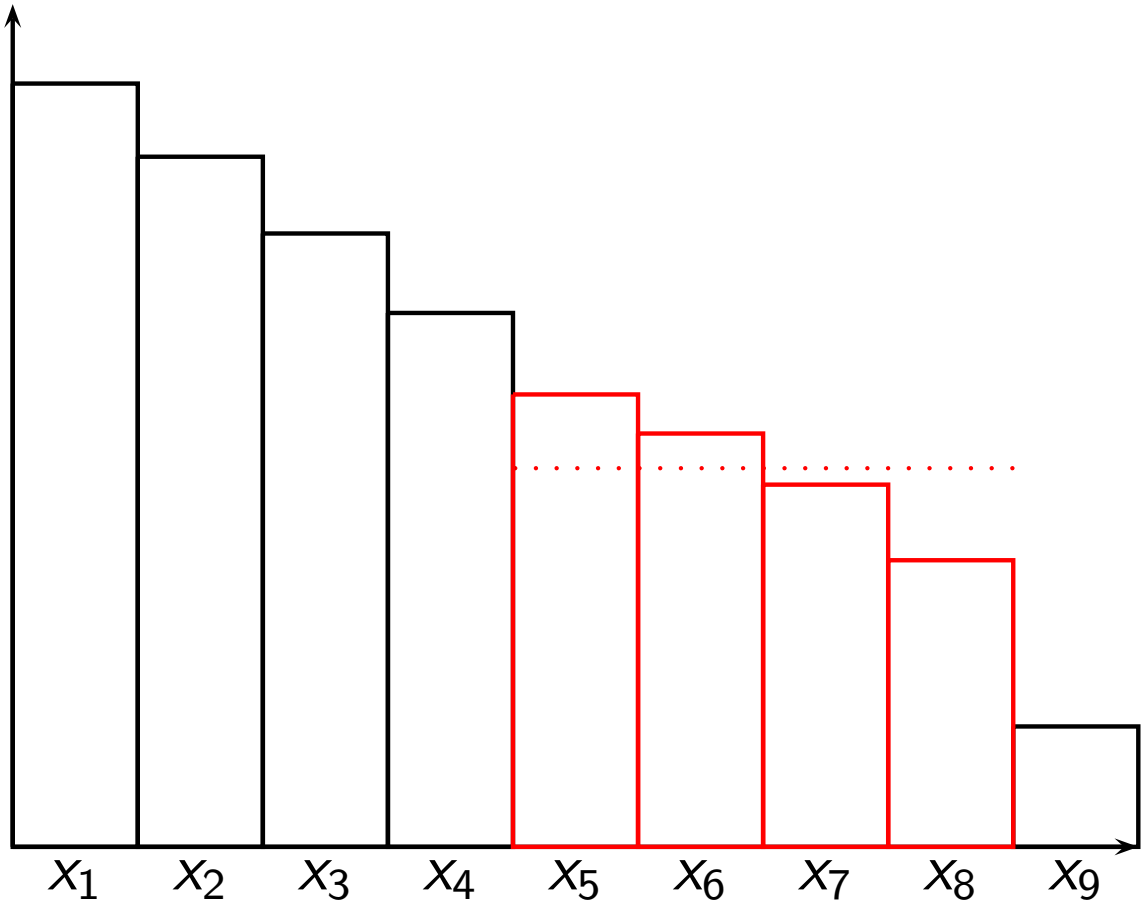
A sandpile model for BKZ$_k$ [HaPuSt11]

Regularity assumption: Each HKZ-reduction gives a worst-case profile



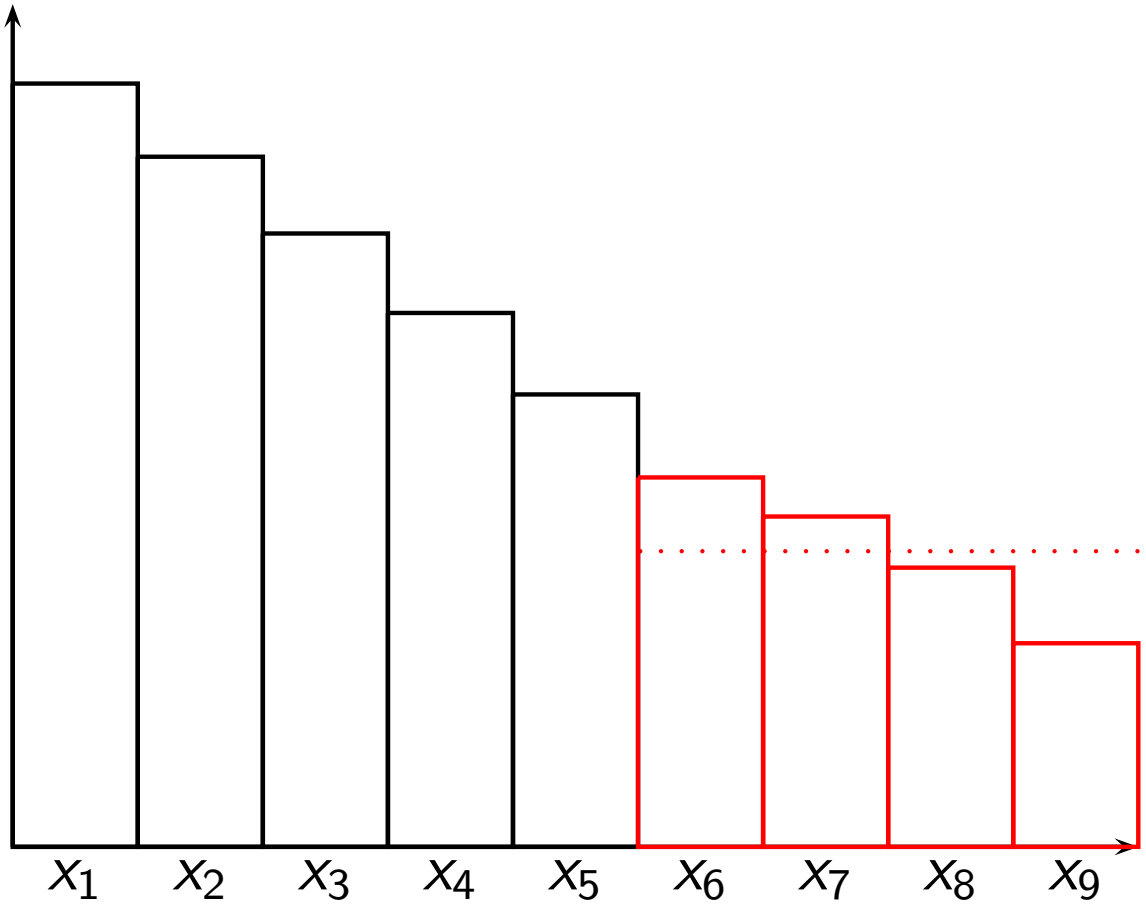
A sandpile model for BKZ_k [HaPuSt11]

Regularity assumption: Each HKZ-reduction gives a worst-case profile



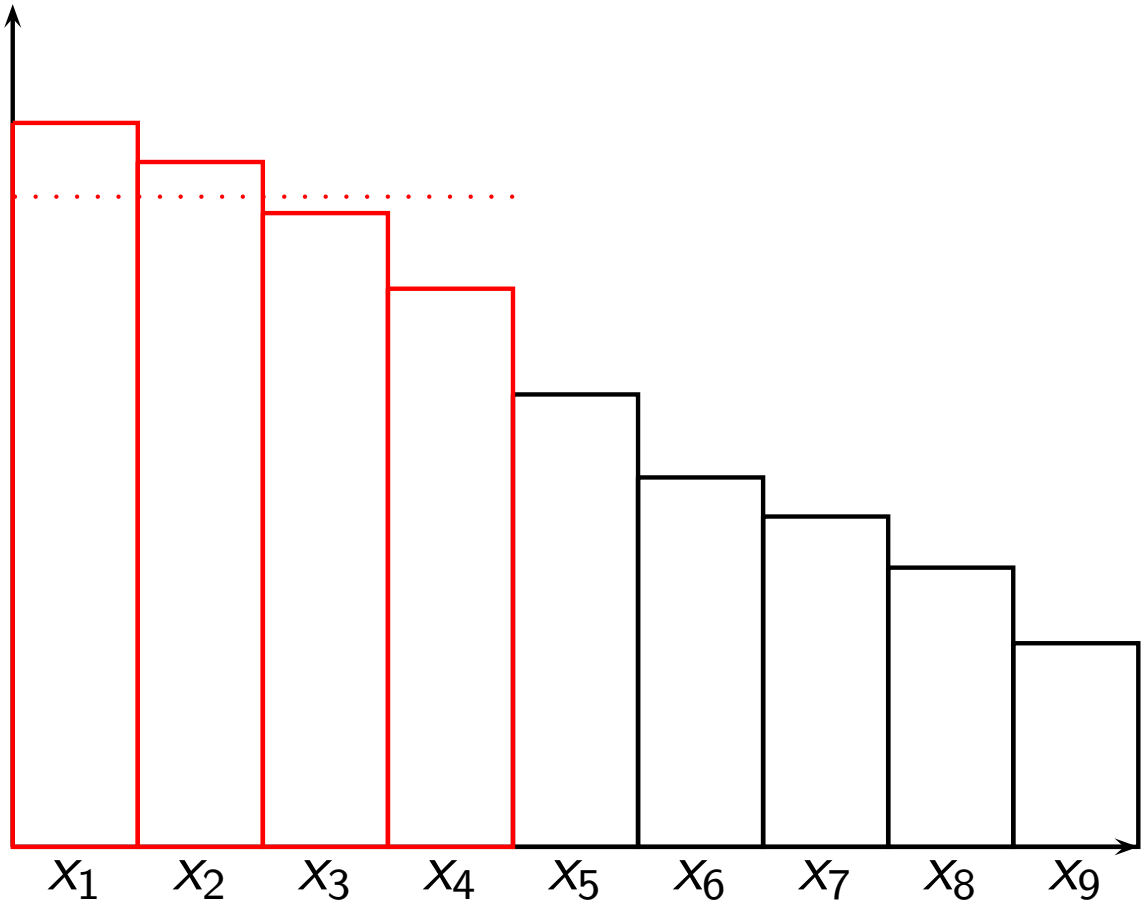
A sandpile model for BKZ_k [HaPuSt11]

Regularity assumption: Each HKZ-reduction gives a worst-case profile

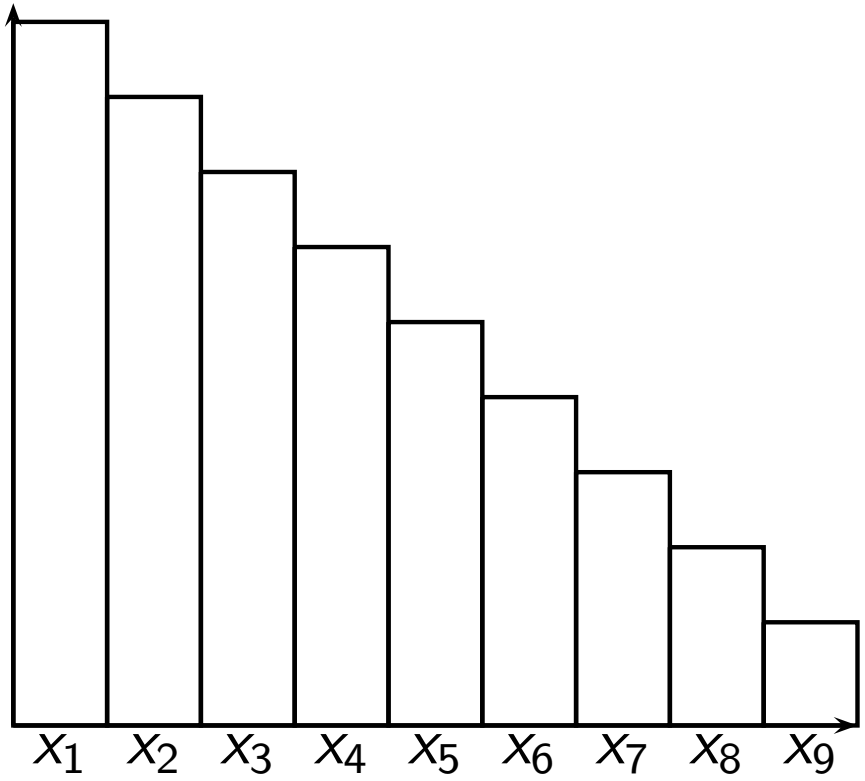


A sandpile model for BKZ_k [HaPuSt11]

Regularity assumption: Each HKZ-reduction gives a worst-case profile



BKZ_k's sandpile as a dynamic system



$$X = (x_1, \ldots, x_n)^T$$

$$X_{0.5} \leftarrow A_1 X$$

$$X_1 \leftarrow A_1 X + \Gamma_1$$

$$X_2 \leftarrow A_2 X_1 + \Gamma_2$$

$$\ldots$$

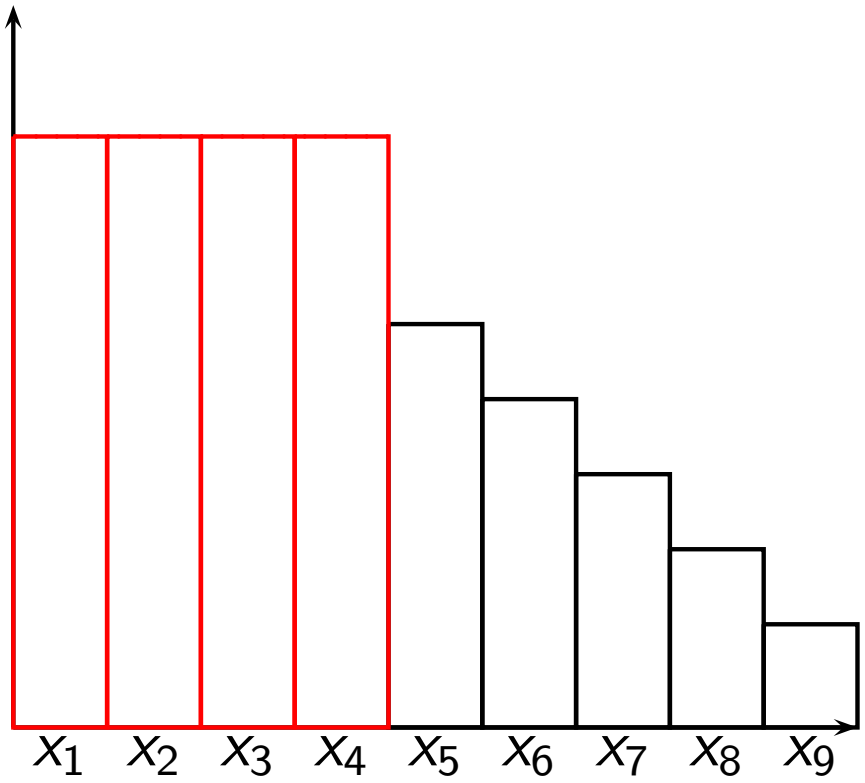
$$X_j = A_j X_j + \Gamma_j$$

with $j = n - k + 1$

A full tour:

$$X' \leftarrow AX + \Gamma$$

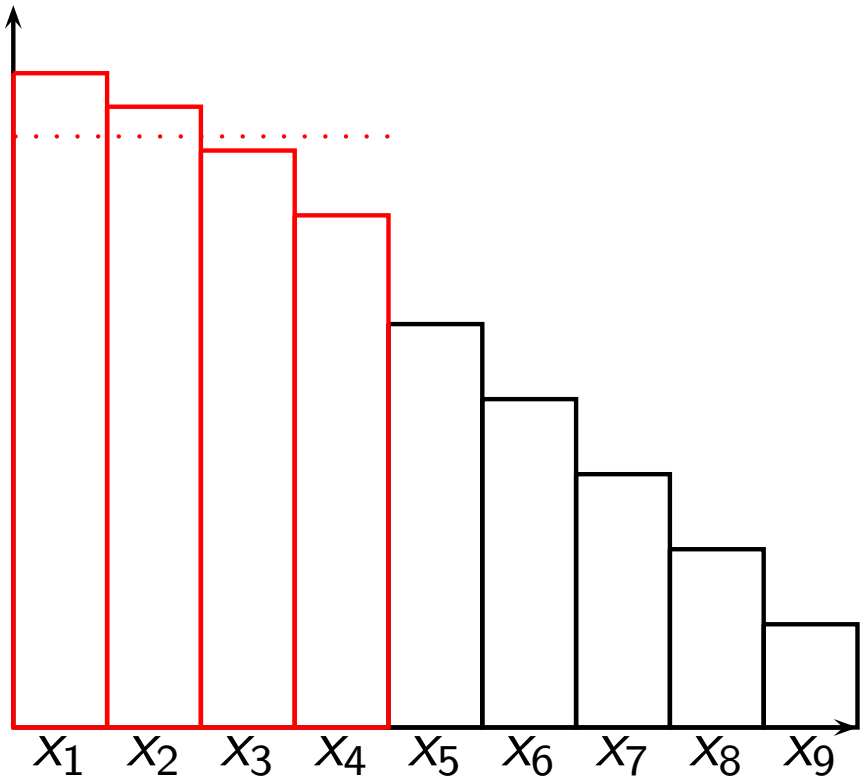
BKZ_k's sandpile as a dynamic system



$$X = (x_1, \ldots, x_n)^T$$
$$X_{0.5} \leftarrow A_1 X$$
$$X_1 \leftarrow A_1 X + \Gamma_1$$
$$X_2 \leftarrow A_2 X_1 + \Gamma_2$$
$$\ldots$$
$$X_j = A_j X_j + \Gamma_j$$
with $j = n - k + 1$

A full tour:
$$X' \leftarrow AX + \Gamma$$

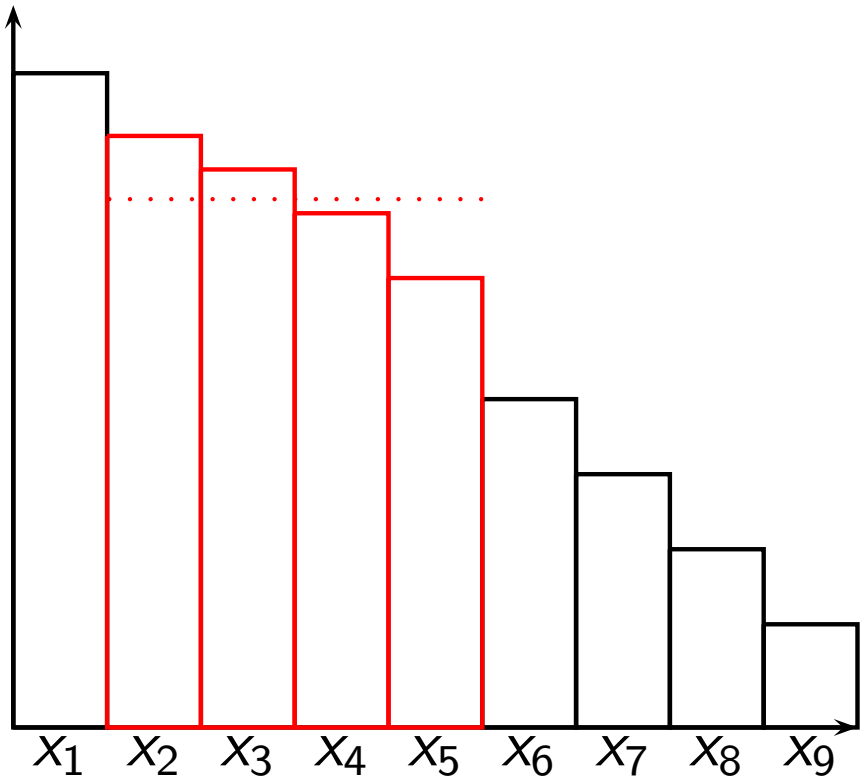
BKZ_k's sandpile as a dynamic system



$$X = (x_1, \ldots, x_n)^T$$
$$X_{0.5} \leftarrow A_1 X$$
$$X_1 \leftarrow A_1 X + \Gamma_1$$
$$X_2 \leftarrow A_2 X_1 + \Gamma_2$$
$$\ldots$$
$$X_j = A_j X_j + \Gamma_j$$
with $j = n - k + 1$

A full tour:
$$X' \leftarrow AX + \Gamma$$

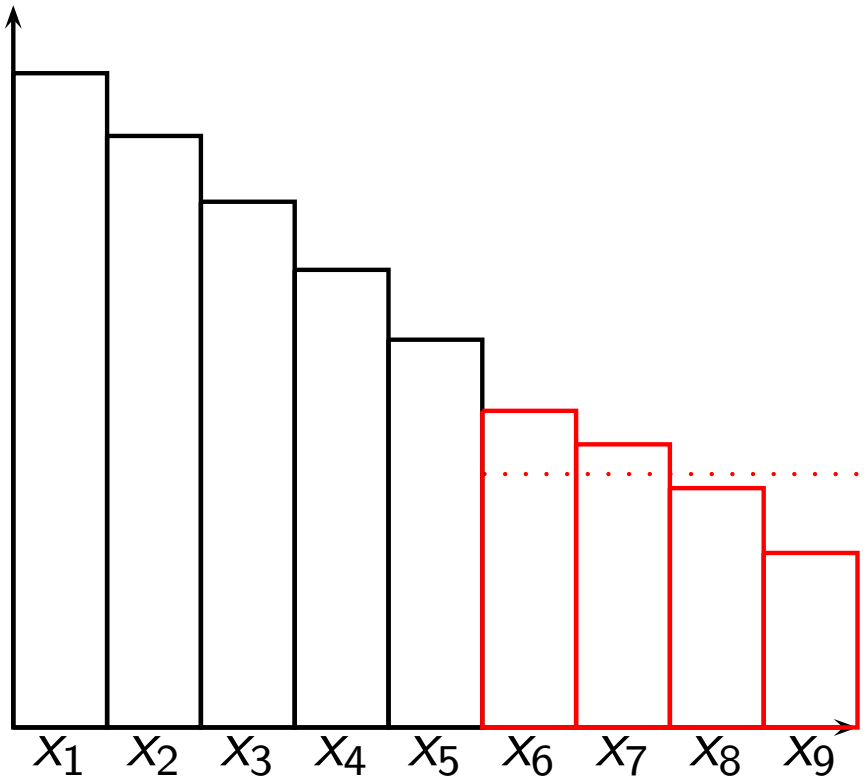
BKZ_k's sandpile as a dynamic system



$$X = (x_1, \ldots, x_n)^T$$
$$X_{0.5} \leftarrow A_1 X$$
$$X_1 \leftarrow A_1 X + \Gamma_1$$
$$X_2 \leftarrow A_2 X_1 + \Gamma_2$$
$$\ldots$$
$$X_j = A_j X_j + \Gamma_j$$
with $j = n - k + 1$

A full tour:
$$X' \leftarrow AX + \Gamma$$

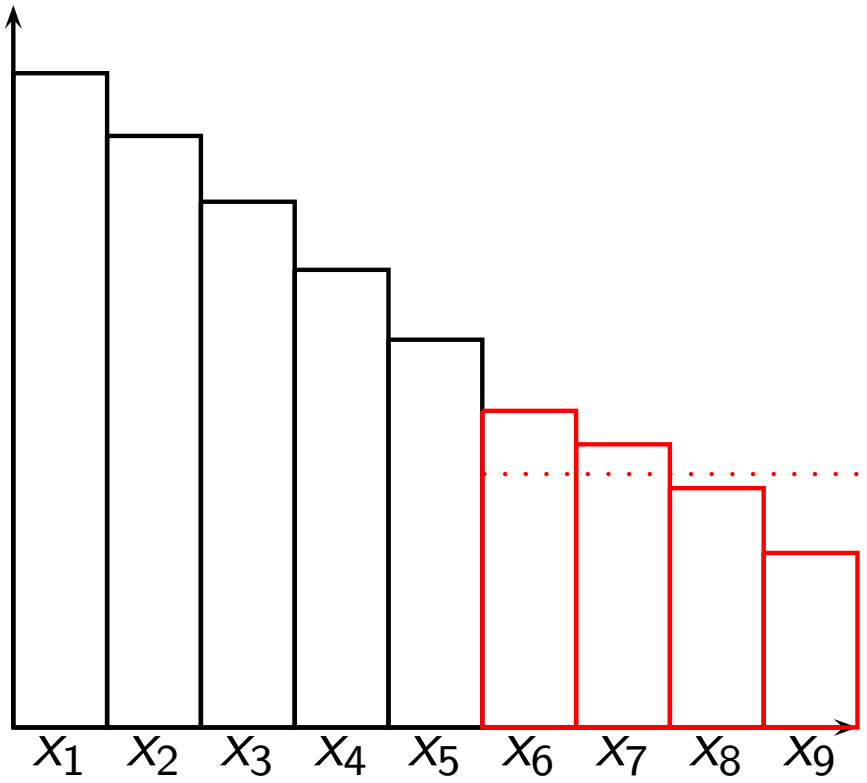
BKZ_k's sandpile as a dynamic system



$$X = (x_1, \ldots, x_n)^T$$
$$X_{0.5} \leftarrow A_1 X$$
$$X_1 \leftarrow A_1 X + \Gamma_1$$
$$X_2 \leftarrow A_2 X_1 + \Gamma_2$$
$$\ldots$$
$$X_j = A_j X_j + \Gamma_j$$
with $j = n - k + 1$

A full tour:
$$X' \leftarrow AX + \Gamma$$

BKZ_k's sandpile as a dynamic system



$$X = (x_1, \ldots, x_n)^T$$
$$X_{0.5} \leftarrow A_1 X$$
$$X_1 \leftarrow A_1 X + \Gamma_1$$
$$X_2 \leftarrow A_2 X_1 + \Gamma_2$$
$$\ldots$$
$$X_j = A_j X_j + \Gamma_j$$
with $j = n - k + 1$

A full tour:
$$X' \leftarrow AX + \Gamma$$

Analysis of BKZ in the sandpile model

Discrete-time affine dynamical system, for one loop iteration

$$X \leftarrow AX + \Gamma$$

- Reducedness of the output $\Rightarrow$ fixed points

- Speed of convergence $\Rightarrow$ eigenvalues of $A^T A$
  - $\Rightarrow$ the convergence is geometric, for the x_i's
  - $\Rightarrow$ for fixed k and n, there are $O(\log\log\|B\|)$ iterations

Analysis of BKZ in the sandpile model

Discrete-time affine dynamical system, for one loop iteration

$$X \leftarrow AX + \Gamma$$

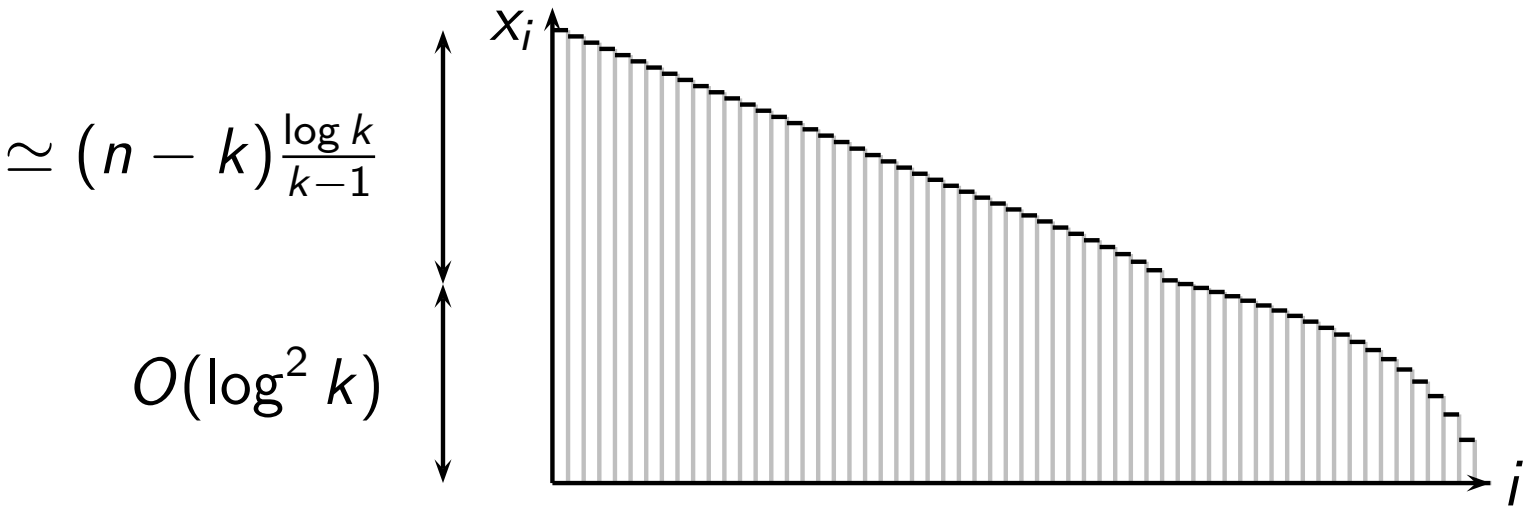
- Reducedness of the output $\Rightarrow$ fixed points
- Speed of convergence $\Rightarrow$ eigenvalues of $A^T A$
  - $\Rightarrow$ the convergence is geometric, for the x_i's
  - $\Rightarrow$ for fixed k and n, there are $O(\log\log\|B\|)$ iterations

Analysis of BKZ in the sandpile model

Discrete-time affine dynamical system, for one loop iteration

$$X \leftarrow AX + \Gamma$$

- Reducedness of the output $\Rightarrow$ fixed points



- Speed of convergence $\Rightarrow$ eigenvalues of $A^T A$

  $\Rightarrow$ the convergence is geometric, for the x_i's

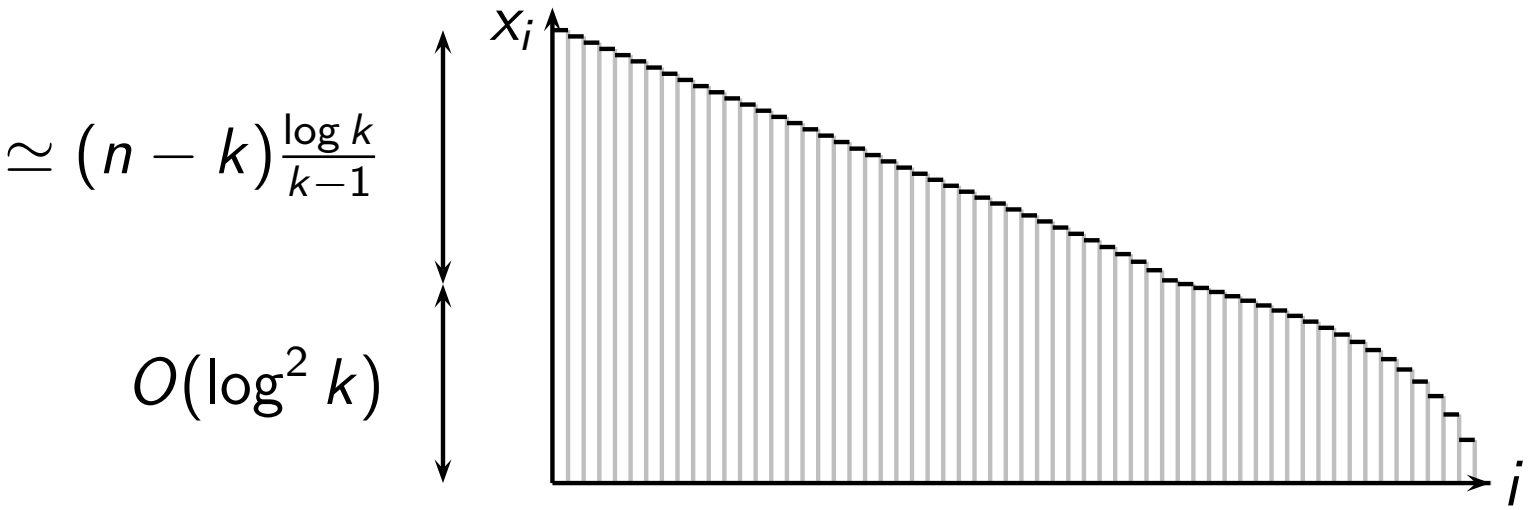
  $\Rightarrow$ for fixed k and n, there are $O(\log\log \|B\|)$ iterations

Analysis of BKZ in the sandpile model

Discrete-time affine dynamical system, for one loop iteration

$$X \leftarrow AX + \Gamma$$

- Reducedness of the output $\Rightarrow$ fixed points



- Speed of convergence $\Rightarrow$ eigenvalues of $A^T A$
  - $\Rightarrow$ the convergence is geometric, for the x_i's
  - $\Rightarrow$ for fixed k and n, there are $O(\log\log\|B\|)$ iterations

Analysis of BKZ [HaPuSt11,Neumaier16]

One can solve $X = AX + \Gamma$, find the eigenvalues of $A^T A$, and remove the regularity assumption... cumbersome...

Neumaier's reducedness parameter

$$\nu := \max_{i \leq n-k} \frac{1}{n-i} \left(\frac{\sum_{j \leq i} x_j}{i} - \frac{\sum_{j \leq n} x_j}{n} \right).$$

- $(\sum_{j \leq i} x_j)/i$ is a smoothed proxy for x_i.
- Taking $i = 1$ gives $\|\mathbf{b}_1\| \leq \exp(\nu)^{n-1} \cdot (\det B)^{1/n}$.
- The definition is justified by the fact we expect the x_i's to decrease linearly after reduction

Analysis of BKZ [HaPuSt11,Neumaier16]

One can solve $X = AX + \Gamma$, find the eigenvalues of $A^T A$, and remove the regularity assumption... cumbersome...

Neumaier's reducedness parameter

$$\nu := \max_{i \leq n-k} \frac{1}{n-i} \left(\frac{\sum_{j \leq i} x_j}{i} - \frac{\sum_{j \leq n} x_j}{n} \right).$$

- $(\sum_{j \leq i} x_j)/i$ is a smoothed proxy for x_i.
- Taking $i = 1$ gives $\|\mathbf{b}_1\| \leq \exp(\nu)^{n-1} \cdot (\det B)^{1/n}$.
- The definition is justified by the fact we expect the x_i's to decrease linearly after reduction

Analysis of BKZ [HaPuSt11,Neumaier16]

One can solve $X = AX + \Gamma$, find the eigenvalues of $A^T A$, and remove the regularity assumption... cumbersome...

Neumaier's reducedness parameter

$$\nu := \max_{i \leq n-k} \frac{1}{n-i} \left(\frac{\sum_{j \leq i} x_j}{i} - \frac{\sum_{j \leq n} x_j}{n} \right).$$

- $(\sum_{j \leq i} x_j)/i$ is a smoothed proxy for x_i.
- Taking $i = 1$ gives $\|\mathbf{b}_1\| \leq \exp(\nu)^{n-1} \cdot (\det B)^{1/n}$.
- The definition is justified by the fact we expect the x_i's to decrease linearly after reduction

Cost of BKZ

Neumaier's reducedness parameter

$$\nu := \max_{i \leq n-k} \frac{1}{n-i} \left(\frac{\sum_{j \leq i} x_j}{i} - \frac{\sum_{j \leq n} x_j}{n} \right).$$

Cost of BKZ

At every tour before reaching the fix-point:

$$\nu \text{ decreases by a factor } \leq 1 - k^2/n^2.$$

$\Rightarrow$ BKZ requires $O(n \cdot \frac{n^2}{k^2} \cdot \log\log \|B\|)$ calls to an SVP oracle to (essentially) reach the fix-point.

Neumaier's parameter also allows to analyze variants of BKZ, including SDBKZ [MiWa16].

Cost of BKZ

Neumaier's reducedness parameter

$$\nu := \max_{i \leq n-k} \frac{1}{n-i} \left(\frac{\sum_{j \leq i} x_j}{i} - \frac{\sum_{j \leq n} x_j}{n} \right).$$

Cost of BKZ

At every tour before reaching the fix-point:

$$\nu \text{ decreases by a factor } \leq 1 - k^2/n^2.$$

$\Rightarrow$ BKZ requires $O(n \cdot \frac{n^2}{k^2} \cdot \log\log \|B\|)$ calls to an SVP oracle to (essentially) reach the fix-point.

Neumaier's parameter also allows to analyze variants of BKZ, including SDBKZ [MiWa16].

Roadmap

1. Background on lattices
2. The lattice reduction framework
3. Strong but slow: BKZ
4. **Solving the SIS problem**
5. Weak but fast: LLL

SIS [Ajt96]

The Small Integer Solution Problem

Given a uniform $A \in \mathbb{Z}_q^{m \times n}$, find $\mathbf{x} \in \mathbb{Z}^m$ s.t.:

$$0 < \|\mathbf{x}\| \leq \beta \quad \text{and} \quad \mathbf{x}^t \cdot A = \mathbf{0} \mod q.$$

- Hash functions [Ajt96,LyMi08,PeRo08]
- Commitment scheme [KeTaXa08]
- Digital signatures [GePeVa08,Boy10,Lyu12]

And many more.

SIS [Ajt96]

The Small Integer Solution Problem

Given a uniform $A \in \mathbb{Z}_q^{m \times n}$, find $\mathbf{x} \in \mathbb{Z}^m$ s.t.:

$$0 < \|\mathbf{x}\| \leq \beta \quad \text{and} \quad \mathbf{x}^t \cdot A = \mathbf{0} \mod q.$$

- Hash functions [Ajt96,LyMi08,PeRo08]
- Commitment scheme [KeTaXa08]
- Digital signatures [GePeVa08,Boy10,Lyu12]

And many more.

Viewing SIS as a lattice problem

Given $A \in \mathbb{Z}_q^{m \times n}$, find $\mathbf{x}$ s.t.: $0 < \|\mathbf{x}\| \leq \beta$ and $\mathbf{x}^t \cdot A = \mathbf{0}\ [q]$

Short $\neq 0$ vector in $L = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^t \cdot A = \mathbf{0}\ [q]\}$:

$$\det L = q^n \text{ (with high prob.)}, \quad \dim L = m$$

We may optimize over $m' \leq m$:

- Less freedom, larger smallest solutions
- But smaller lattice dimension

Viewing SIS as a lattice problem

Given $A \in \mathbb{Z}_q^{m \times n}$, find $\mathbf{x}$ s.t.: $0 < \|\mathbf{x}\| \leq \beta$ and $\mathbf{x}^t \cdot A = \mathbf{0}\ [q]$

Short $\neq 0$ vector in $L = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^t \cdot A = \mathbf{0}\ [q]\}$:

$$\det L = q^n \text{ (with high prob.)}, \quad \dim L = m$$

We may optimize over $m' \leq m$:

- Less freedom, larger smallest solutions
- But smaller lattice dimension

Viewing SIS as a lattice problem

Given $A \in \mathbb{Z}_q^{m \times n}$, find $\mathbf{x}$ s.t.: $0 < \|\mathbf{x}\| \leq \beta$ and $\mathbf{x}^t \cdot A = \mathbf{0}\ [q]$

Short $\neq 0$ vector in $L = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^t \cdot A = \mathbf{0}\ [q]\}$:

$$\det L = q^n \text{ (with high prob.)}, \quad \dim L = m$$

We may optimize over $m' \leq m$:

- Less freedom, larger smallest solutions
- But smaller lattice dimension

Solving SIS with BKZ

Short $\neq 0$ vector in $L = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^t \cdot A = \mathbf{0}\ [q]\}$:

$$\det L = q^n \text{ (with high prob.)}, \quad \dim L = m$$

Lattice reduction on L

$$\gamma \cdot q^{\frac{n}{m}} \quad \text{needs to be} \quad \leq \beta$$

$$\text{Cost grows as } (m/\log\gamma)^{O(m/\log\gamma)}.$$

$$\Rightarrow \text{Look for } \min_{m' \leq m} x \log x \quad \text{with} \quad x = \frac{m'}{\log\beta - \frac{n}{m'}\log q}$$

If m is large enough, take $m' \approx \sqrt{n \log q / \log\beta}$.

$$\text{Cost is} \leq \exp\left(O\left(\frac{n\log q}{\log^2\beta} \cdot \log\frac{n\log q}{\log^2\beta}\right)\right)$$

Solving SIS with BKZ

Short $\neq 0$ vector in $L = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^t \cdot A = \mathbf{0}\ [q]\}$:

$$\det L = q^n \text{ (with high prob.)}, \quad \dim L = m$$

Lattice reduction on L

$$\gamma \cdot q^{\frac{n}{m}} \quad \text{needs to be} \quad \leq \beta$$

Cost grows as $(m/\log\gamma)^{O(m/\log\gamma)}$.

$\Rightarrow$ Look for $\min_{m' \leq m} x \log x$ with $x = \frac{m'}{\log\beta - \frac{n}{m'}\log q}$

If m is large enough, take $m' \approx \sqrt{n \log q / \log\beta}$.

$$\text{Cost is} \leq \exp\left(O\left(\frac{n\log q}{\log^2\beta} \cdot \log\frac{n\log q}{\log^2\beta}\right)\right)$$

Solving SIS with BKZ

Short $\neq 0$ vector in $L = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^t \cdot A = \mathbf{0}\ [q]\}$:

$$\det L = q^n \text{ (with high prob.)}, \quad \dim L = m$$

Lattice reduction on L

$$\gamma \cdot q^{\frac{n}{m}} \quad \text{needs to be} \quad \leq \beta$$
$$\text{Cost grows as } (m/\log\gamma)^{O(m/\log\gamma)}.$$

$\Rightarrow$ Look for $\min_{m' \leq m} x \log x$ with $x = \frac{m'}{\log\beta - \frac{n}{m'}\log q}$

If m is large enough, take $m' \approx \sqrt{n \log q / \log\beta}$.

$$\text{Cost is} \leq \exp\left(O\left(\frac{n\log q}{\log^2\beta} \cdot \log\frac{n\log q}{\log^2\beta}\right)\right)$$

Solving SIS with BKZ

Short $\neq 0$ vector in $L = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^t \cdot A = \mathbf{0}\ [q]\}$:

$$\det L = q^n \text{ (with high prob.)}, \quad \dim L = m$$

Lattice reduction on L

$$\gamma \cdot q^{\frac{n}{m}} \quad \text{needs to be} \quad \leq \beta$$
$$\text{Cost grows as } (m/\log\gamma)^{O(m/\log\gamma)}.$$

$\Rightarrow$ Look for $\min_{m' \leq m} x \log x$ with $x = \frac{m'}{\log\beta - \frac{n}{m'}\log q}$

If m is large enough, take $m' \approx \sqrt{n \log q / \log\beta}$.

$$\text{Cost is} \leq \exp\left(O\left(\frac{n\log q}{\log^2\beta} \cdot \log\frac{n\log q}{\log^2\beta}\right)\right)$$

Roadmap

1. Background on lattices
2. The lattice reduction framework
3. Strong but slow: BKZ
4. Solving the SIS problem
5. **Weak but fast: LLL**

LLL as sandpile flattening [MadVaI10]

(Recall that $x_i = \log \|\mathbf{b}_i^*\| = \log r_{ii}$, for $i \leq n$.)

The LLL sandpile flattening strategy is **local**.

LLL as sandpile flattening [MadVaI10]

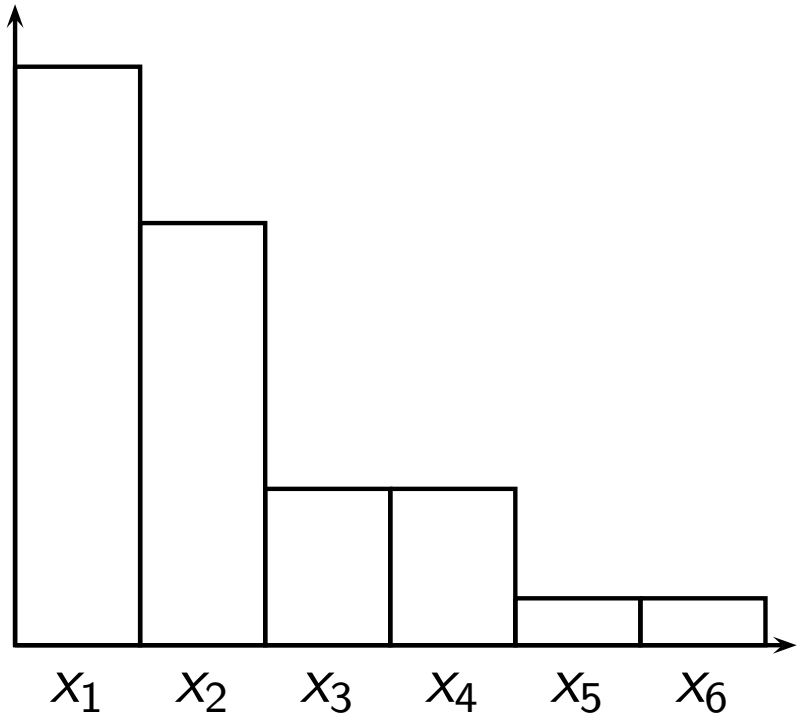
(Recall that $x_i = \log \|\mathbf{b}_i^*\| = \log r_{ii}$, for $i \leq n$.)

The LLL sandpile flattening strategy is **local**.

LLL: if $r_{ii} \gg r_{i+1,i+1}$, do $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

This decreases r_{ii} by at least a constant factor.

This decreases x_i by at least a constant.



LLL as sandpile flattening [MadVaI10]

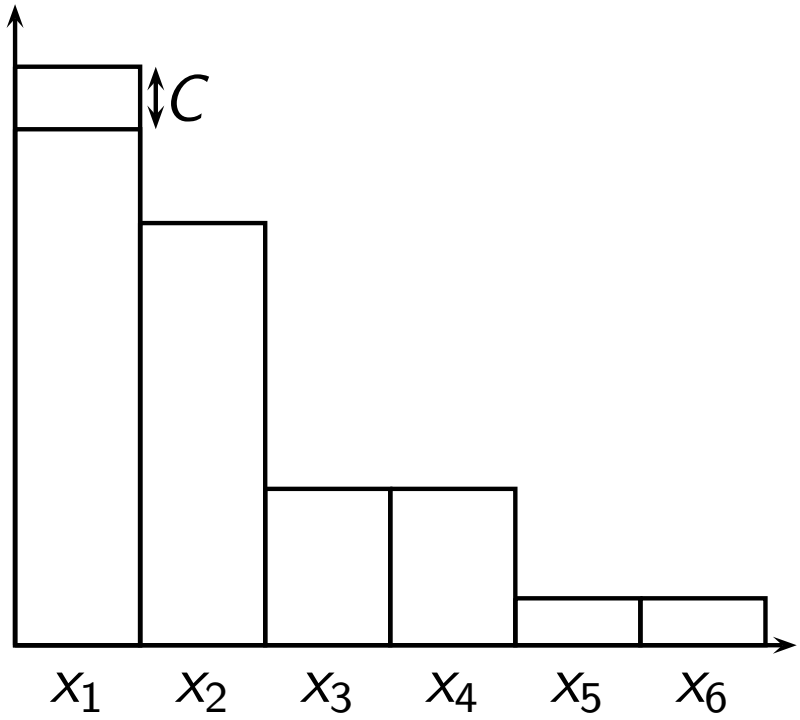
(Recall that $x_i = \log \|\mathbf{b}_i^*\| = \log r_{ii}$, for $i \leq n$.)

The LLL sandpile flattening strategy is **local**.

LLL: if $r_{ii} \gg r_{i+1,i+1}$, do $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

This decreases r_{ii} by at least a constant factor.

This decreases x_i by at least a constant.



LLL as sandpile flattening [MadVaI10]

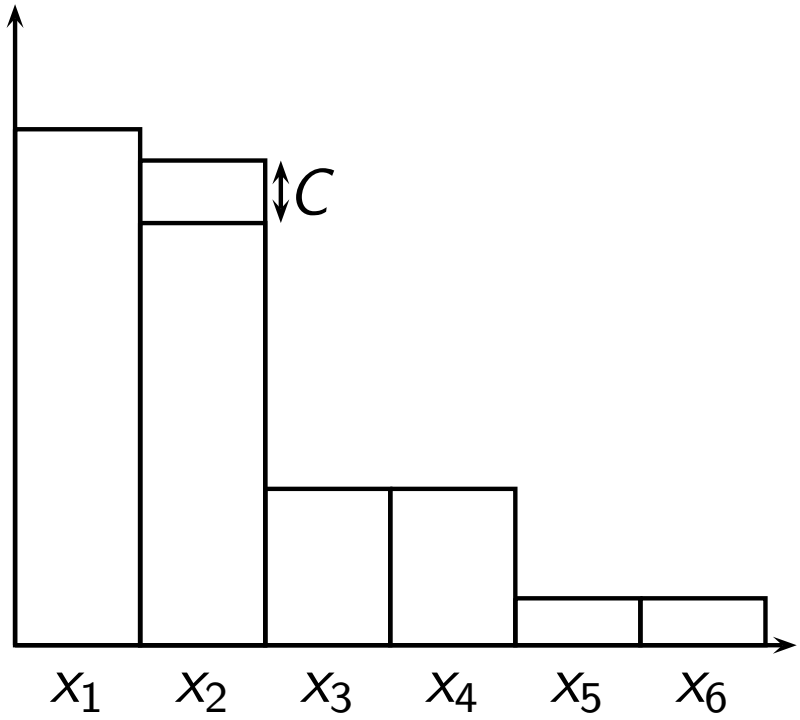
(Recall that $x_i = \log \|\mathbf{b}_i^*\| = \log r_{ii}$, for $i \leq n$.)

The LLL sandpile flattening strategy is **local**.

LLL: if $r_{ii} \gg r_{i+1,i+1}$, do $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

This decreases r_{ii} by at least a constant factor.

This decreases x_i by at least a constant.



LLL as sandpile flattening [MadVaI10]

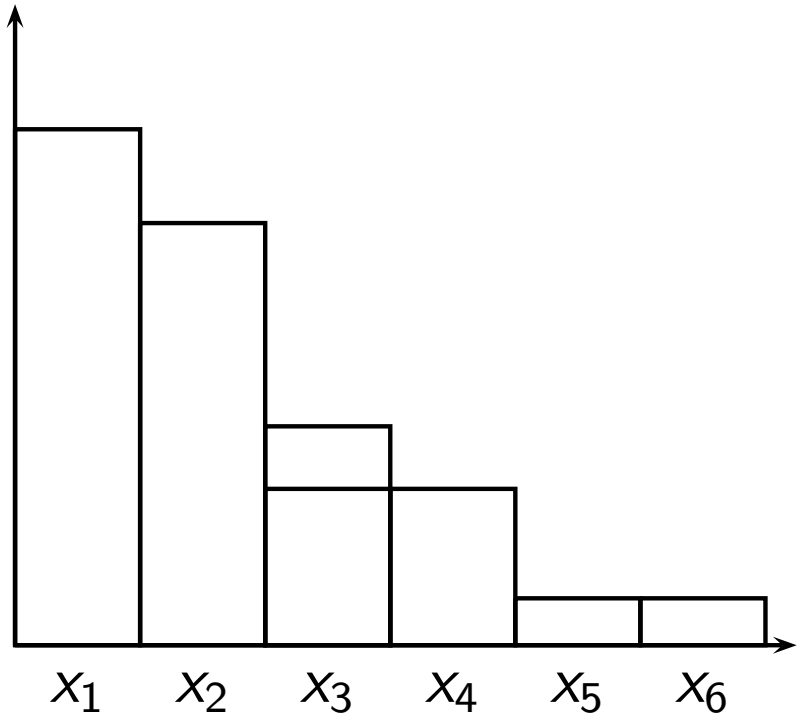
(Recall that $x_i = \log \|\mathbf{b}_i^*\| = \log r_{ii}$, for $i \leq n$.)

The LLL sandpile flattening strategy is **local**.

LLL: if $r_{ii} \gg r_{i+1,i+1}$, do $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

This decreases r_{ii} by at least a constant factor.

This decreases x_i by at least a constant.



LLL as sandpile flattening [MadVaI10]

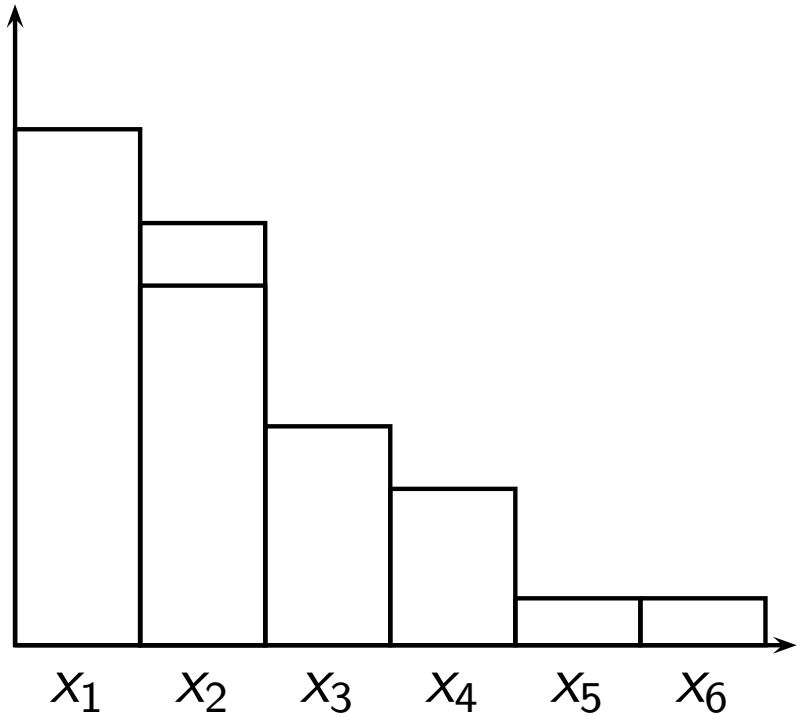
(Recall that $x_i = \log \|\mathbf{b}_i^*\| = \log r_{ii}$, for $i \leq n$.)

The LLL sandpile flattening strategy is **local**.

LLL: if $r_{ii} \gg r_{i+1,i+1}$, do $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

This decreases r_{ii} by at least a constant factor.

This decreases x_i by at least a constant.



LLL as sandpile flattening [MadVaI10]

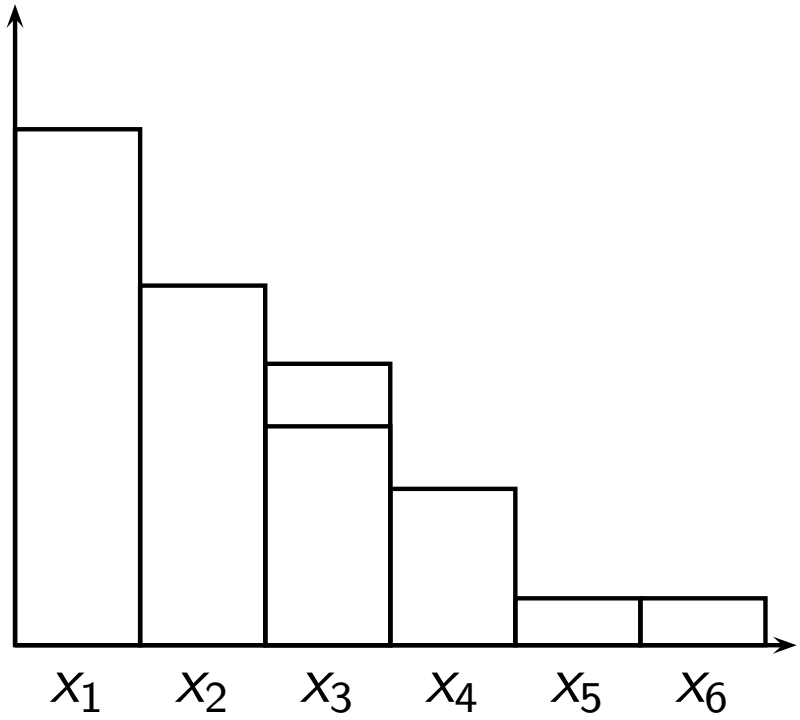
(Recall that $x_i = \log \|\mathbf{b}_i^*\| = \log r_{ii}$, for $i \leq n$.)

The LLL sandpile flattening strategy is **local**.

LLL: if $r_{ii} \gg r_{i+1,i+1}$, do $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

This decreases r_{ii} by at least a constant factor.

This decreases x_i by at least a constant.



LLL as sandpile flattening [MadVaI10]

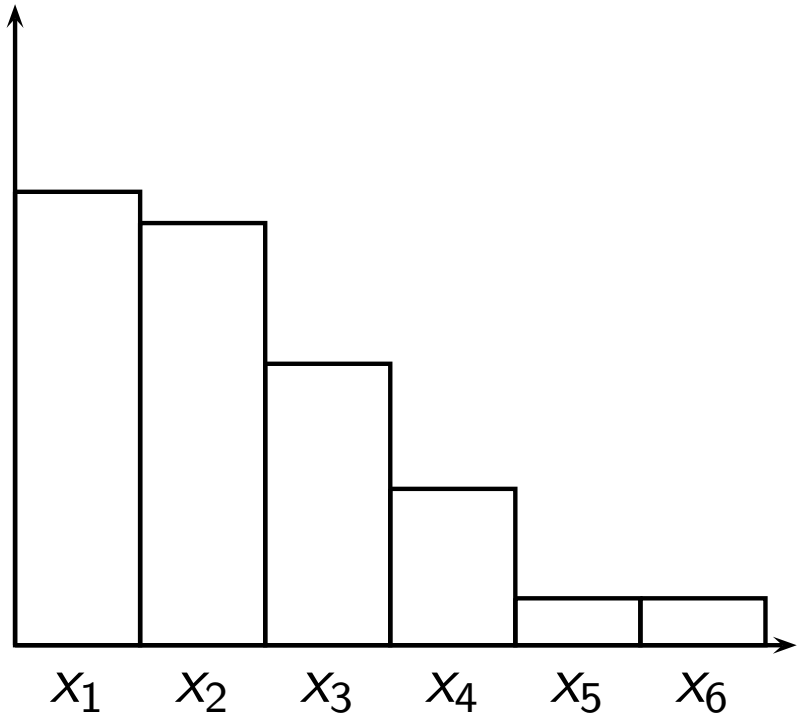
(Recall that $x_i = \log \|\mathbf{b}_i^*\| = \log r_{ii}$, for $i \leq n$.)

The LLL sandpile flattening strategy is **local**.

LLL: if $r_{ii} \gg r_{i+1,i+1}$, do $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

This decreases r_{ii} by at least a constant factor.

This decreases x_i by at least a constant.



LLL as sandpile flattening [MadVaI10]

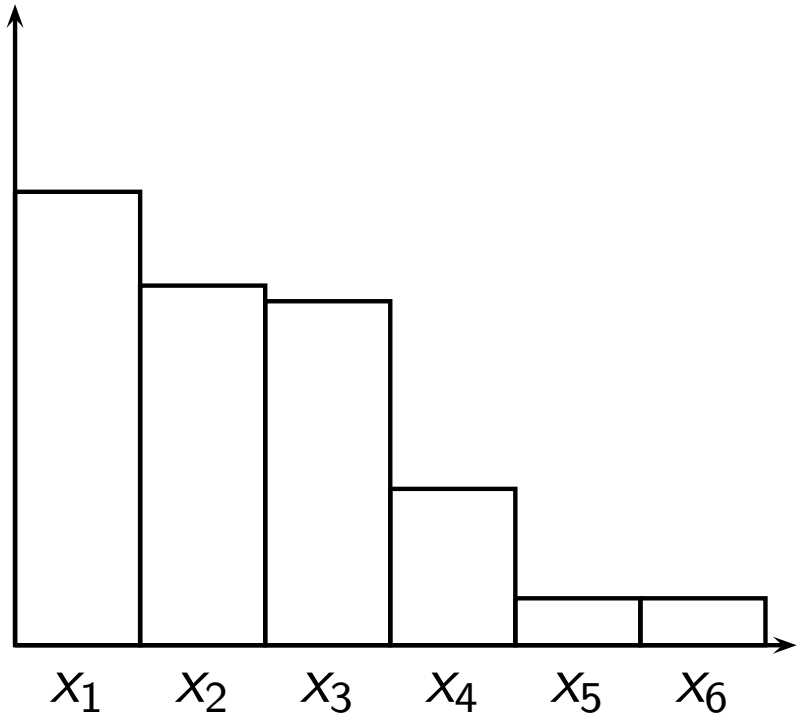
(Recall that $x_i = \log \|\mathbf{b}_i^*\| = \log r_{ii}$, for $i \leq n$.)

The LLL sandpile flattening strategy is **local**.

LLL: if $r_{ii} \gg r_{i+1,i+1}$, do $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

This decreases r_{ii} by at least a constant factor.

This decreases x_i by at least a constant.



LLL as sandpile flattening [MadVaI10]

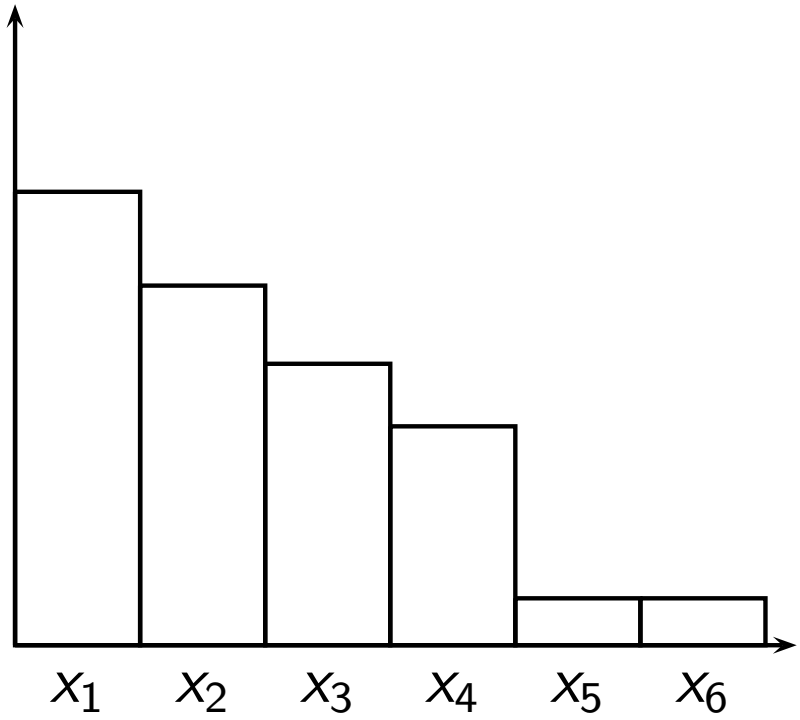
(Recall that $x_i = \log \|\mathbf{b}_i^*\| = \log r_{ii}$, for $i \leq n$.)

The LLL sandpile flattening strategy is **local**.

LLL: if $r_{ii} \gg r_{i+1,i+1}$, do $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

This decreases r_{ii} by at least a constant factor.

This decreases x_i by at least a constant.



LLL as sandpile flattening [MadVaI10]

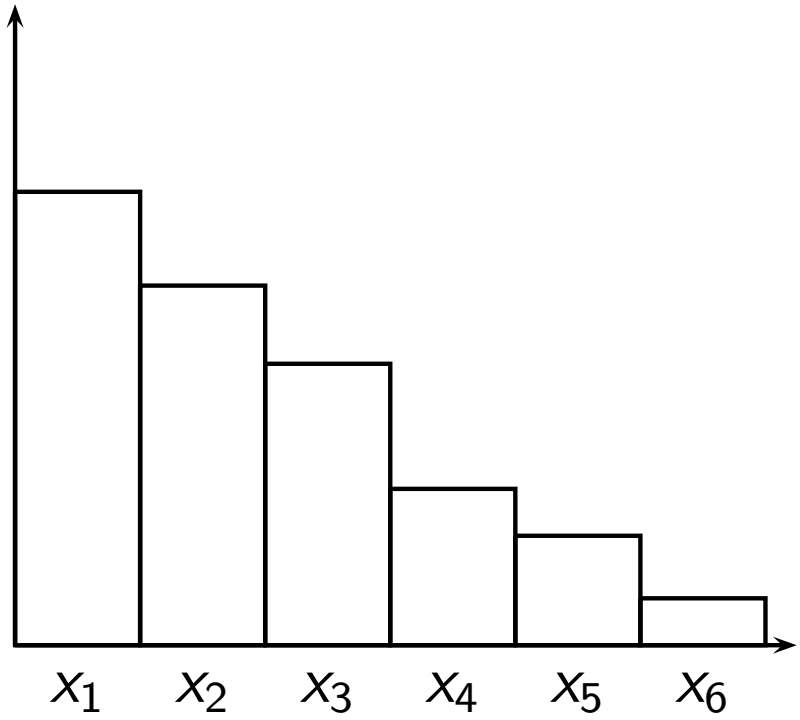
(Recall that $x_i = \log \|\mathbf{b}_i^*\| = \log r_{ii}$, for $i \leq n$.)

The LLL sandpile flattening strategy is **local**.

LLL: if $r_{ii} \gg r_{i+1,i+1}$, do $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

This decreases r_{ii} by at least a constant factor.

This decreases x_i by at least a constant.



LLL as sandpile flattening [MadVaI10]

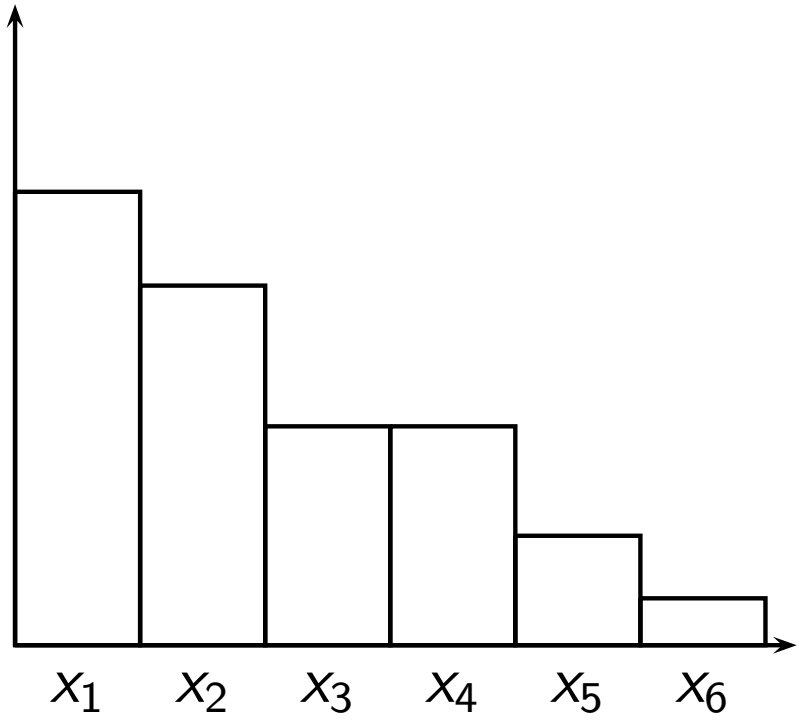
(Recall that $x_i = \log \|\mathbf{b}_i^*\| = \log r_{ii}$, for $i \leq n$.)

The LLL sandpile flattening strategy is **local**.

LLL: if $r_{ii} \gg r_{i+1,i+1}$, do $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

This decreases r_{ii} by at least a constant factor.

This decreases x_i by at least a constant.



LLL as sandpile flattening [MadVaI10]

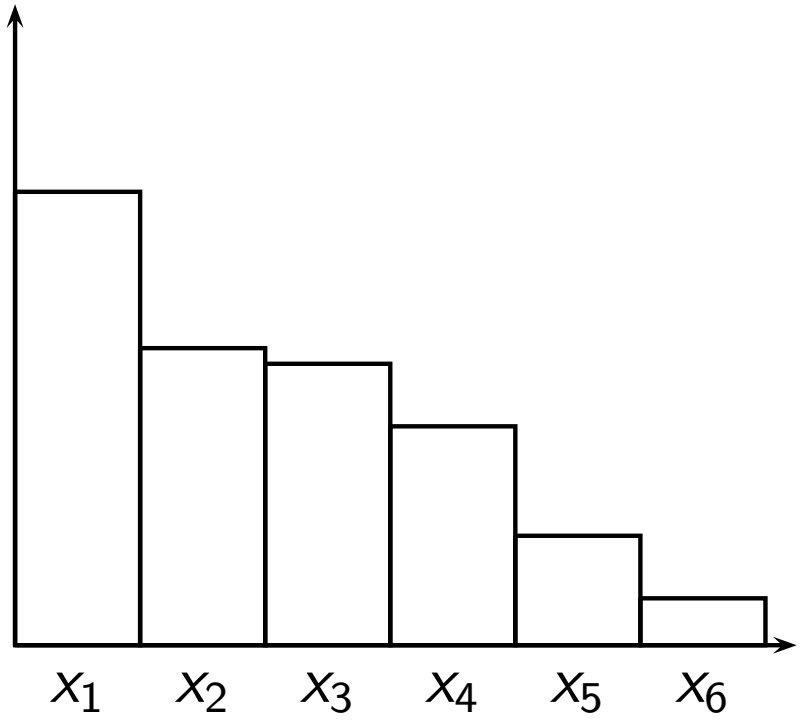
(Recall that $x_i = \log \|\mathbf{b}_i^*\| = \log r_{ii}$, for $i \leq n$.)

The LLL sandpile flattening strategy is **local**.

LLL: if $r_{ii} \gg r_{i+1,i+1}$, do $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

This decreases r_{ii} by at least a constant factor.

This decreases x_i by at least a constant.



LLL as sandpile flattening [MadVaI10]

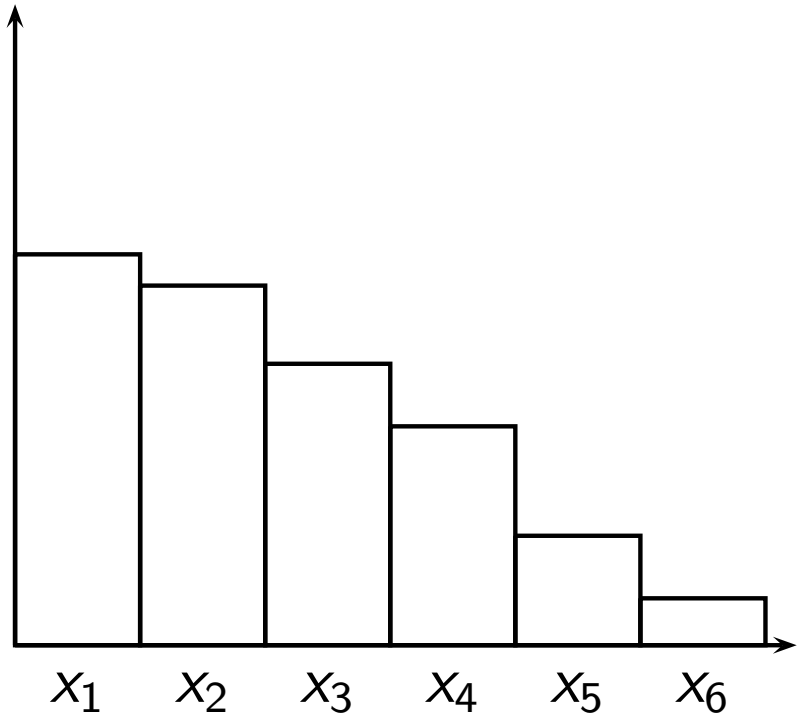
(Recall that $x_i = \log \|\mathbf{b}_i^*\| = \log r_{ii}$, for $i \leq n$.)

The LLL sandpile flattening strategy is **local**.

LLL: if $r_{ii} \gg r_{i+1,i+1}$, do $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$.

This decreases r_{ii} by at least a constant factor.

This decreases x_i by at least a constant.



Convergence of LLL

The LLL potential

$$\Pi := \prod_{i \leq n} (n - i + 1) \cdot x_i.$$

- It is the weighted amount of sand to be moved to the right.
- For each swap, it decreases by at least a constant.

Number of loop iterations of LLL

There are $O(n^2 \log \|B\|)$ loop iterations before completion.

Convergence of LLL

The LLL potential

$$\Pi := \prod_{i \leq n} (n - i + 1) \cdot x_i.$$

- It is the weighted amount of sand to be moved to the right.
- For each swap, it decreases by at least a constant.

Number of loop iterations of LLL

There are $O(n^2 \log \|B\|)$ loop iterations before completion.

Convergence of LLL

The LLL potential

$$\Pi := \sum_{i \leq n} (n - i + 1) \cdot x_i.$$

- It is the weighted amount of sand to be moved to the right.
- For each swap, it decreases by at least a constant.

Number of loop iterations of LLL

There are $O(n^2 \log \|B\|)$ loop iterations before completion.

Cost of LLL

Text-book LLL:

- $O(n^2 \log \|B\|)$ loop iterations
- $O(n^2)$ arithmetic operations per iteration
- GSO rationals have bit-lengths $O(n \log \|B\|)$

$\Rightarrow$ Cost is $\widetilde{O}(n^5 \log^2 \|B\|)$

Improvements:

- Use floating-point GSO [NgSt05]:
$$\widetilde{O}(n^4 \log^2 \|B\|)$$
- Recursively use approximations for B (like fast gcd) [NoStVe11]:
$$\widetilde{O}(n^3 \log \|B\|)$$

Cost of LLL

Text-book LLL:

- $O(n^2 \log \|B\|)$ loop iterations
- $O(n^2)$ arithmetic operations per iteration
- GSO rationals have bit-lengths $O(n \log \|B\|)$

$\Rightarrow$ Cost is $\widetilde{O}(n^5 \log^2 \|B\|)$

Improvements:

- Use floating-point GSO [NgSt05]:
$$\widetilde{O}(n^4 \log^2 \|B\|)$$
- Recursively use approximations for B (like fast gcd) [NoStVi11]:
$$\widetilde{O}(n^5 \log \|B\|)$$

Cost of LLL

Text-book LLL:

- $O(n^2 \log \|B\|)$ loop iterations
- $O(n^2)$ arithmetic operations per iteration
- GSO rationals have bit-lengths $O(n \log \|B\|)$

$\Rightarrow$ Cost is $\widetilde{O}(n^5 \log^2 \|B\|)$

Improvements:

- Use floating-point GSO [NgSt05]:
$$\widetilde{O}(n^4 \log^2 \|B\|)$$
- Recursively use approximations for B (like fast gcd) [NoStVi11]:
$$\widetilde{O}(n^5 \log \|B\|)$$

Faster LLL-type reduction [NeSt16]

Ideas:

- Use a BKZ-like **global** strategy
- In the k-dimensional blocks, make a recursive call
- Make the blocks overlap by half only
- At the bottom of the recursion, use a quasi-linear 2-dimensional algorithm.

This is solving a local-global dilemma:

- Global sandpile flattening strategy
- Stay local, so that working dimension is small

Faster LLL-type reduction [NeSt16]

Ideas:

- Use a BKZ-like **global** strategy
- In the k-dimensional blocks, make a recursive call
- Make the blocks overlap by half only
- At the bottom of the recursion, use a quasi-linear 2-dimensional algorithm.

This is solving a local-global dilemma:

- Global sandpile flattening strategy
- Stay local, so that working dimension is small

Faster LLL-type reduction

Analysis:

- Using Neumaier's parameter: $O(n^2/k^2)$ tours.
- In a tour, we have $O(n/k)$ recursive calls
- Size-reduction and GSO update after a call: $O(n^2 k)$ arithmetic operations

Number of arithmetic operations (including 2-dimensional reductions): $\widetilde{O}(n^3)$.

Total cost: $\widetilde{O}(n^4 \log \|B\|)$

Faster LLL-type reduction

Analysis:

- Using Neumaier's parameter: $O(n^2/k^2)$ tours.
- In a tour, we have $O(n/k)$ recursive calls
- Size-reduction and GSO update after a call: $O(n^2 k)$ arithmetic operations

Number of arithmetic operations (including 2-dimensional reductions): $\widetilde{O}(n^3)$.

Total cost: $\widetilde{O}(n^4 \log \|B\|)$

Roadmap

1. Background on lattices
2. The lattice reduction framework
3. Strong but slow: BKZ
4. Solving the SIS problem
5. Weak but fast: LLL

What to remember from this talk?

- Lattice reduction is used to solve the approximate variants of SVP/uSVP/HSVP/SIVP/...
- The process is driven by the r_{ii}'s
- Time $2^k \Leftrightarrow$ approx. factor $\gamma = k^{O(n/k)}$
  or... approx. factor γ in time $(1 + n/\log \gamma)^{O(n/\log \gamma)}$

What to remember from this talk?

- Lattice reduction is used to solve the approximate variants of SVP/uSVP/HSVP/SIVP/...
- The process is driven by the r_{ii}'s
- Time $2^k \Leftrightarrow$ approx. factor $\gamma = k^{O(n/k)}$
  or... approx. factor γ in time $(1 + n/\log \gamma)^{O(n/\log \gamma)}$

What to remember from this talk?

- Lattice reduction is used to solve the approximate variants of SVP/uSVP/HSVP/SIVP/...
- The process is driven by the r_{ii}'s
- Time $2^k \Leftrightarrow$ approx. factor $\gamma = k^{O(n/k)}$
  or... approx. factor γ in time $(1 + n/\log \gamma)^{O(n/\log \gamma)}$

What to remember from this talk?

Two approaches to flatten the sandpile.

- Global (BKZ, fast LLL): $O(n^3 \log\log \|B\|)$ iterations.
- Local (LLL): $O(n^2 \log \|B\|)$ iterations.

- Global approach seems superior
- But in practice, local remains better for LLL reduction
- And also global kicks in only if there are many iterations, whereas local may be cheaper for some instances.

What to remember from this talk?

Two approaches to flatten the sandpile.

- Global (BKZ, fast LLL): $O(n^3 \log\log \|B\|)$ iterations.
- Local (LLL): $O(n^2 \log \|B\|)$ iterations.

- Global approach seems superior
- But in practice, local remains better for LLL reduction
- And also global kicks in only if there are many iterations, whereas local may be cheaper for some instances.

Open problems

1. Practice
2. Understand the relationship between global and local flattening
3. Faster LLL-type reduction: $\widetilde{O}(n^\omega \log \|B\|)$?
4. Go beyond this framework:
   - Why sticking to the input lattice?
   - Why progressive improvements?
5. Find a quantum acceleration

Main open problem

Achieve approx. factor γ in time $\left(\frac{n}{\log \gamma}\right)^{o\left(\frac{n}{\log \gamma}\right)}$, for some γ.

Open problems

1. Practice
2. Understand the relationship between global and local flattening
3. Faster LLL-type reduction: $\widetilde{O}(n^\omega \log \|B\|)$?
4. Go beyond this framework:
   - Why sticking to the input lattice?
   - Why progressive improvements?
5. Find a quantum acceleration

Main open problem

Achieve approx. factor γ in time $\left(\frac{n}{\log \gamma}\right)^{o\left(\frac{n}{\log \gamma}\right)}$, for some γ.

Open problems

1. Practice
2. Understand the relationship between global and local flattening
3. Faster LLL-type reduction: $\widetilde{O}(n^\omega \log \|B\|)$?
4. Go beyond this framework:
   - Why sticking to the input lattice?
   - Why progressive improvements?
5. Find a quantum acceleration

Main open problem

Achieve approx. factor γ in time $\left(\frac{n}{\log \gamma}\right)^{o\left(\frac{n}{\log \gamma}\right)}$, for some γ.

Open problems

1. Practice
2. Understand the relationship between global and local flattening
3. Faster LLL-type reduction: $\widetilde{O}(n^\omega \log \|B\|)$?
4. Go beyond this framework:
   - Why sticking to the input lattice?
   - Why progressive improvements?
5. Find a quantum acceleration

Main open problem

Achieve approx. factor γ in time $\left(\frac{n}{\log \gamma}\right)^{o\left(\frac{n}{\log \gamma}\right)}$, for some γ.

Open problems

1. Practice
2. Understand the relationship between global and local flattening
3. Faster LLL-type reduction: $\widetilde{O}(n^\omega \log \|B\|)$?
4. Go beyond this framework:
   - Why sticking to the input lattice?
   - Why progressive improvements?
5. Find a quantum acceleration

Main open problem

Achieve approx. factor γ in time $\left(\frac{n}{\log \gamma}\right)^{o\left(\frac{n}{\log \gamma}\right)}$, for some γ.

Open problems

1. Practice
2. Understand the relationship between global and local flattening
3. Faster LLL-type reduction: $\widetilde{O}(n^\omega \log \|B\|)$?
4. Go beyond this framework:
   - Why sticking to the input lattice?
   - Why progressive improvements?
5. Find a quantum acceleration

Main open problem

Achieve approx. factor γ in time $\left(\frac{n}{\log \gamma}\right)^{o\left(\frac{n}{\log \gamma}\right)}$, for some γ.