The Nearest-Colattice algorithm

Time-approximation tradeoff for Approx-CVP

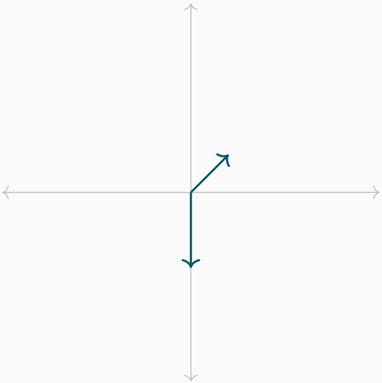
Thomas Espitau and Paul Kirchner

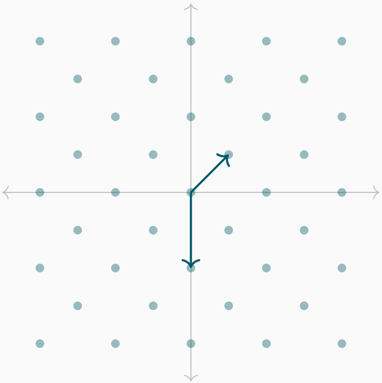
ANTS 2020

Lattices, (H)SVP, CVP

Lattices in a nutshell

Lattices in a nutshell

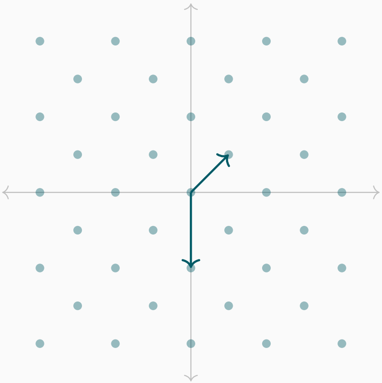




Lattice

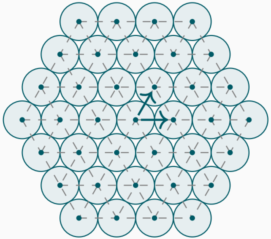
A (Euclidean) lattice L is a *discrete* subgroup of an Euclidean space (say $\mathbb{R}^n$).

Lattices in a nutshell



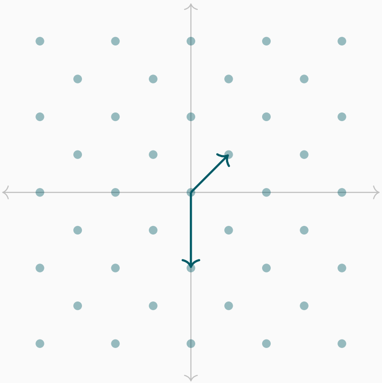
Lattice

A (Euclidean) lattice L is a *discrete* subgroup of an Euclidean space (say $\mathbb{R}^n$).



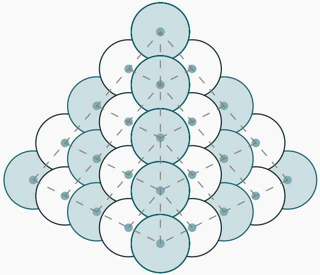
Sphere Packing problem

Lattices in a nutshell



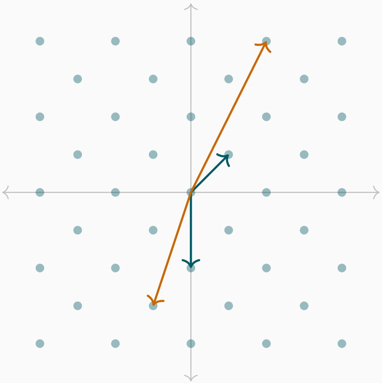
Lattice

A (Euclidean) lattice L is a *discrete* subgroup of an Euclidean space (say $\mathbb{R}^n$).



Sphere Packing problem

Lattices in a nutshell



Lattice

A (Euclidean) lattice L is a *discrete* subgroup of an Euclidean space (say $\mathbb{R}^n$).

- Finding a shortest vector: hard
- Solving lattice problems depends on the quality of the basis
  - Size of vectors
  - Orthogonality defect of basis

Closest vector problem

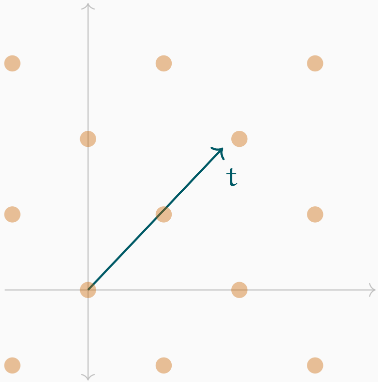
- Given a lattice Λ and a vector t in the ambient space:

  Retrieve the closest vector of Λ to t.

Closest vector problem

- Given a lattice Λ and a vector t in the ambient space:

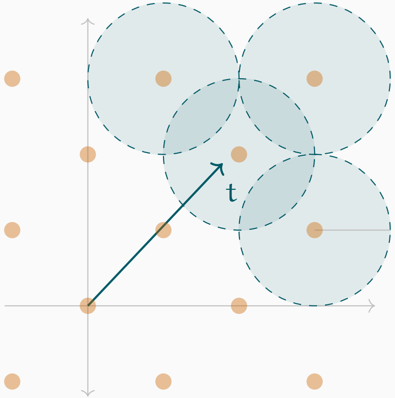
  Retrieve the closest vector of Λ to t.



Closest vector problem

- Given a lattice Λ and a vector t in the ambient space:

  Retrieve the closest vector of Λ to t.

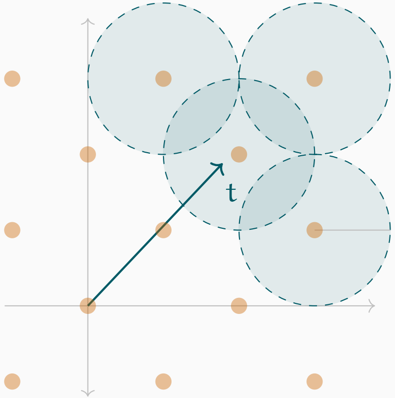


Closest vector problem

- Given a lattice Λ and a vector t in the ambient space:

  Retrieve the closest vector of Λ to t.

- Decode at distance $\mu(\Lambda)$.



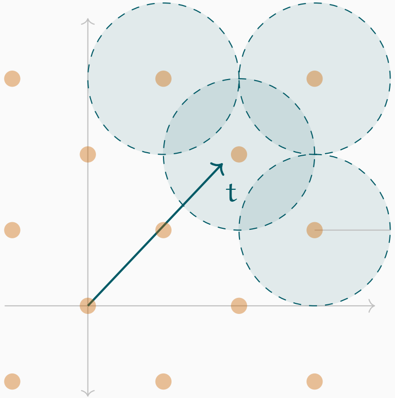
Closest vector problem

- Given a lattice Λ and a vector t in the ambient space:

  Retrieve the closest vector of Λ to t.

- Decode at distance $\mu(\Lambda)$.

- Solving the problem exactly is hard
  - Emumeration ([HS]) $n^{n/2}$
  - Sieve (proved) ([ADS]) $(2+o(1))^n$
  - Sieve (heur.) ([BDGL]) $\left(\frac{4}{3}+o(1)\right)^{\frac{n}{2}}$



Relaxed version: γ-Approx-CVP: *find* $v \in \Lambda$ *at distance at most* $\gamma \min_{w\in\Lambda} \|w - t\|$

Hierarchy for SVP/CVP

1	$\dots$	$\beta^{\frac{n}{2\beta}}$	$\dots$	2^n
$(4/3 + o(1))^n$		$(4/3 + o(1))^\beta$		Poly

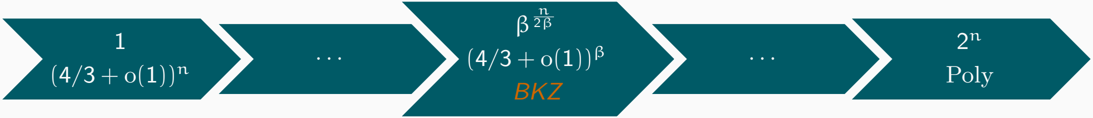
- Hierarchy given by BKZ algorithm: hinges on the call of (exact) SVP oracle in dimension to β to solve the relaxed problem in dimension n

Hierarchy for SVP/CVP

1	$\cdots$	$\beta^{\frac{n}{2\beta}}$	$\cdots$	2^n
$(4/3+o(1))^n$		$(4/3+o(1))^\beta$		Poly
Enum/Sieve				

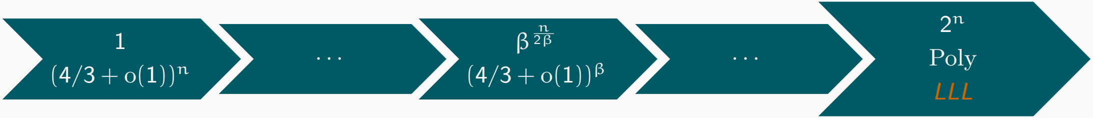
- Hierarchy given by BKZ algorithm: hinges on the call of (exact) SVP oracle in dimension to β to solve the relaxed problem in dimension n

Hierarchy for SVP/CVP



- Hierarchy given by BKZ algorithm: hinges on the call of (exact) SVP oracle in dimension to β to solve the relaxed problem in dimension n

Hierarchy for SVP/CVP



- Hierarchy given by BKZ algorithm: hinges on the call of (exact) SVP oracle in dimension to β to solve the relaxed problem in dimension n

Hierarchy for SVP/CVP

1	$\dots$	$\beta^{\frac{n}{2\beta}}$	$\dots$	2^n
$(4/3+o(1))^n$		$(4/3+o(1))^\beta$		Poly

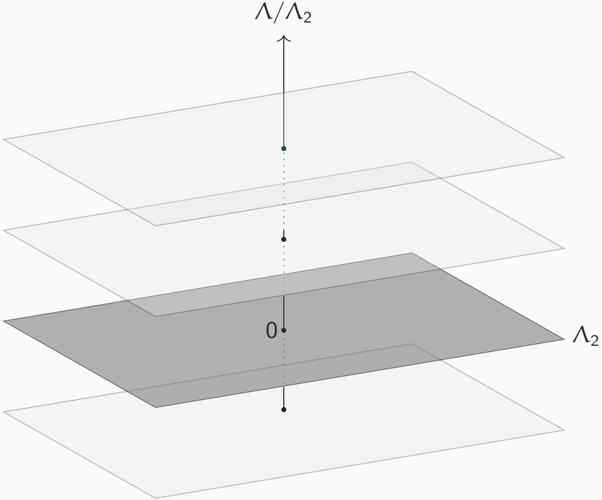
- Hierarchy given by BKZ algorithm: hinges on the call of (exact) SVP oracle in dimension to β to solve the relaxed problem in dimension n
- Exists for Approx-CVP by Kannan's embedding, but use the reduction to SVP, and does not allow preprocess.

$$\text{CVP in } \Lambda \text{ for } t = e + v \longrightarrow \text{SVP to reveal } e \text{ in } \left[\begin{array}{c|c} \Lambda & 0 \\ \hline t & K \end{array}\right]$$

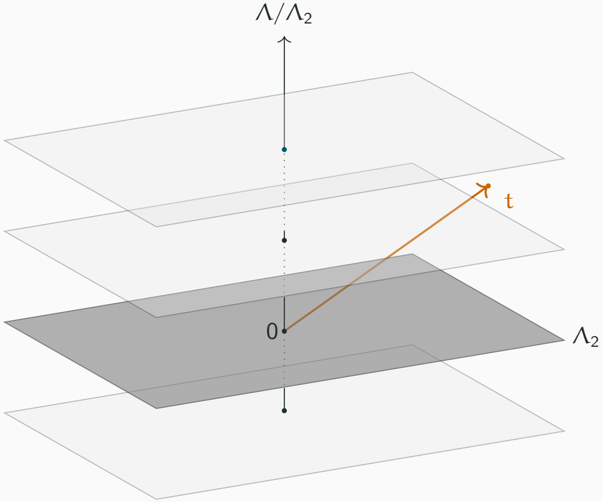
Hierarchy for SVP/CVP

- Hierarchy given by BKZ algorithm: hinges on the call of (exact) SVP oracle in dimension to β to solve the relaxed problem in dimension n
- Exists for Approx-CVP by Kannan's embedding, but use the reduction to SVP, and does not allow preprocess.
- Natural approach ? (i.e. using an oracle CVP)

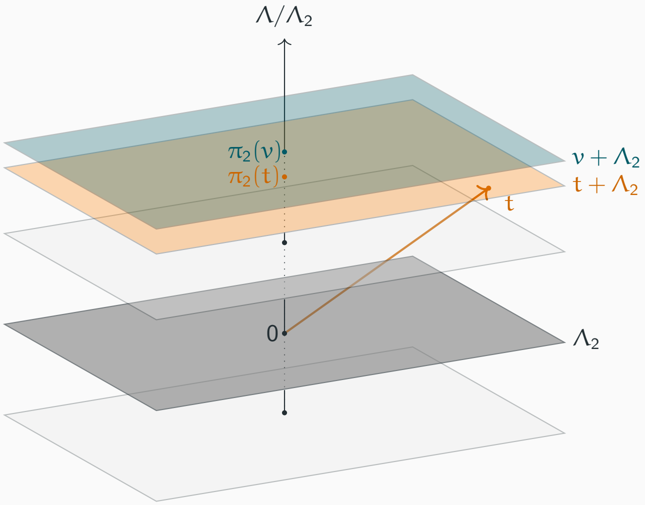
From Babai's nearest plane...



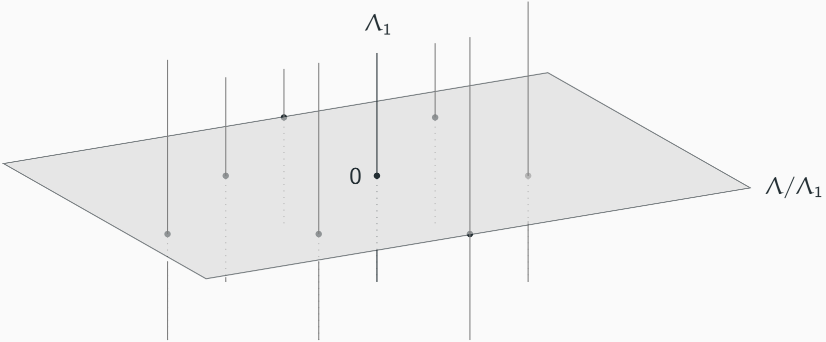
From Babai's nearest plane...



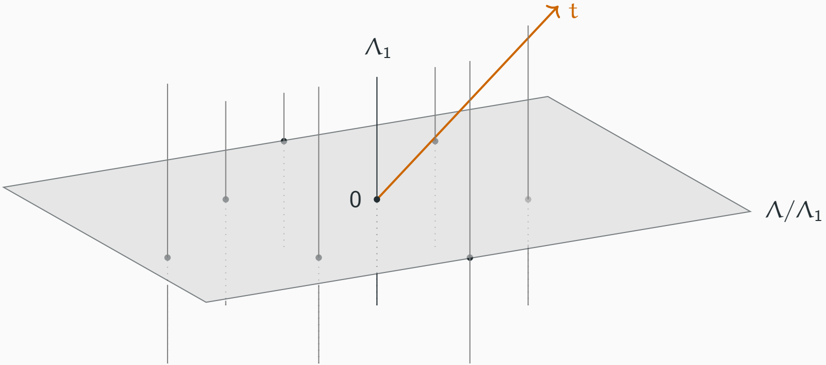
From Babai's nearest plane...



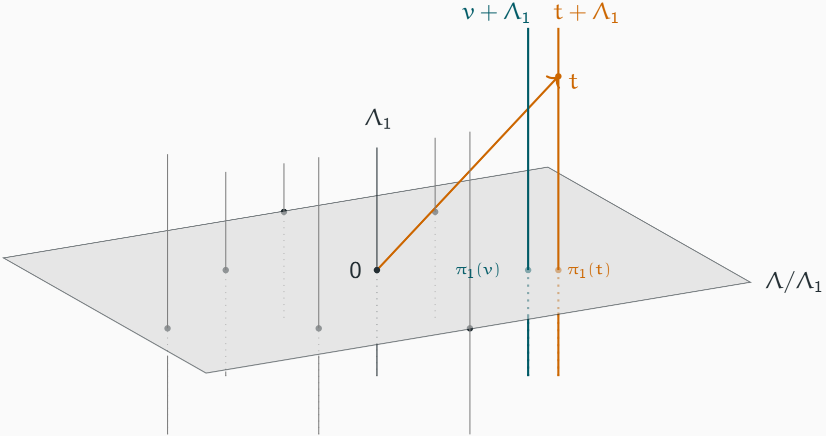
To nearest-2-colattice



To nearest-2-colattice



To nearest-2-colattice



… to a general nearest-colattice algorithm

Algorithm 1: Nearest-collatice

Input : A filtration $\{0\} = \Lambda_0 \subset \Lambda_1 \subset \cdots \subset \Lambda_k = \Lambda$

Output : A vector $v \in \Lambda$

$s \leftarrow -t$

for $i = k$ **downto** 1 **do**

  $s \leftarrow s - \text{Lift}(\textsc{argmin}_{h \in \Lambda_i/\Lambda_{i-1}} \|v - h\|)$

end for

return $t + s$

Quality: $\|x - t\|^2 \leqslant \sum_{i=1}^{k} \mu\left({}^{\Lambda_{i+1}}\!/\!{}_{\Lambda_i}\right)^2$ in time $T_{CVP}(\beta)\text{Poly}(n, \log\|t\|, \log\|B\|)$

Averaged analysis

- For a random lattice of rank n: $\lambda_1 > c\sqrt{n} \implies \mu \leqslant \sqrt{n}$

Averaged analysis

- For a random lattice of rank n: $\lambda_1 > c\sqrt{n} \implies \mu \leqslant \sqrt{n}$

Average behavior

Given a BKZ-β reduced basis and supposing that every sublattice behaves as a random lattice, Nearest-Colattice finds a vector $x \in \Lambda$ such that

$$\|x - t\| \leqslant \Theta(\beta)^{\frac{n}{2\beta}} \mathrm{covol}(\Lambda)^{\frac{1}{n}}$$

in time $T_{CVP}(\beta)\mathrm{Poly}(n, \log \|t\|, \log \|B\|)$.

Averaged analysis

- For a random lattice of rank n: $\lambda_1 > c\sqrt{n} \implies \mu \leqslant \sqrt{n}$

Average behavior

Given a BKZ-β reduced basis and supposing that every sublattice behaves as a random lattice, Nearest-Colattice finds a vector $x \in \Lambda$ such that

$$\|x - t\| \leqslant \Theta(\beta)^{\frac{n}{2\beta}} \mathrm{covol}(\Lambda)^{\frac{1}{n}}$$

in time $T_{CVP}(\beta)\mathrm{Poly}(n, \log\|t\|, \log\|B\|)$.

- BKZ algorithm: Find a vector such that

$$\|v\| \leqslant \Theta(\beta)^{\frac{n}{2\beta}} \mathrm{covol}(\Lambda)^{\frac{1}{n}}$$

in time $T_{SVP}(\beta)\mathrm{Poly}(n, \log\|t\|, \log\|B\|)$.

Applications in Cryptanalysis

- The CVP problem is ubiquitous in cryptanalysis
- Class of signatures schemes (à la GPV)

valid signature $\equiv$ lattice point *close to a public target*

$\rightarrow$ Solving CVP $\Rightarrow$ Forgery

Applications in Cryptanalysis

- The CVP problem is ubiquitous in cryptanalysis
- Class of signatures schemes (à la GPV)

  valid signature $\equiv$ lattice point *close to a public target*

  $\rightarrow$ Solving CVP $\Rightarrow$ Forgery

- Nearest-colattice algorithm $\Rightarrow$ once a reduced basis is found, batch forgery is easy.

- Applies to tradeoff in primal-attack on LWE: allows to use lattice reduction only once to amortize the cost of combinatorial techniques (guessing, small enumeration, ...)

Thank you !