Sieving: Finding Short Lattice Vectors using Space

Phong Nguyễn

Inria
INVENTORS FOR THE DIGITAL WORLD

ENS
ÉCOLE NORMALE SUPÉRIEURE

Provable vs Heuristic

- Sieving comes in two flavours:
  - Provable algorithm with rigorous analysis
  - Heuristic algorithm where not much is known. These have the best claimed running times.

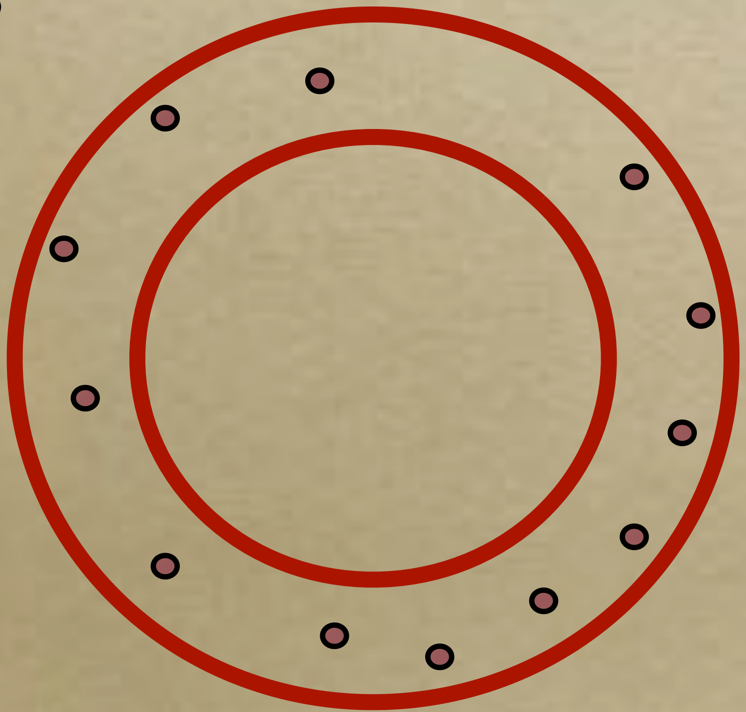
Practical Sieves

Practical Sieves

- Sieve algorithms were believed to be impractical until [NgVi08]: « Sieve algorithms for the shortest vector problem are practical ».

Intuition

- You have a huge number m of lattice vectors $v_1, \ldots, v_m$ inside the ball of radius R
- Can you transform these vectors to decrease R?



Insight

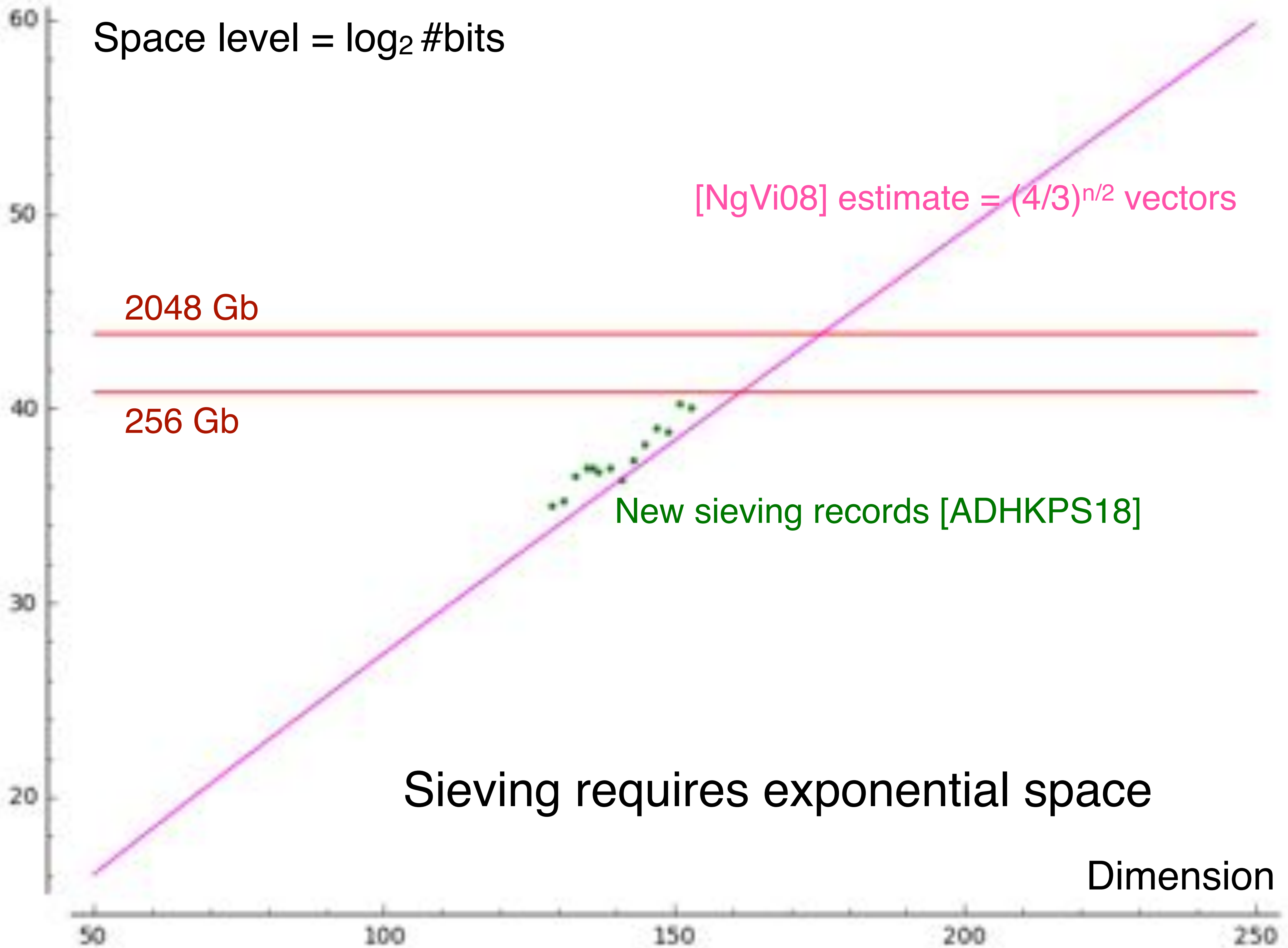
- For any $R' < R$, there exists a subset C of $V = \{v_i\}$ such that the sets $\mathrm{Ball}(c,R') \cap V$ form a partition of V:
  - Each v_i belongs to some $\mathrm{Ball}(c,R')$ where $c \in C$.
  - The balls $\mathrm{Ball}(c,R')$ do not overlap when c ranges over C.

Generic Sieve

- Generate exponentially many lattice vectors $v_1,\ldots,v_m$ inside the ball of radius R. Choose $\varepsilon>0$.
- While no v_i is short enough
  - Compute all the pairs $v_i - v_j$ whose norm is $\leq (1-\varepsilon)R$.
  - Replace the v_i's by these pairs, and update R

Optimizations

- [NgVi08] heuristically estimates that approximately $m=(4/3)^{n/2}$ vectors are required in practice.
- A naive sieve [NgVi08] runs in time quadratic in m.
- In the past five years, several methods based on nearest neighbor search do it in subquadratic time.



Space level = log2 #bits
[NgVi08] estimate = (4/3)n/2 vectors
2048 Gb
256 Gb
New sieving records [ADHKPS18]
Sieving requires exponential space
Dimension
60
50
40
30
20
50
100
150
200
250

Provable Sieves

Provable Sieves

- [AKS01]: One can solve SVP in randomized time and space $2^{O(n)}$.
- [Sc03] claimed that $O(n) \geq 30n$.
- [NgVi08]: [AKS01] can run in time $2^{5.9n}$ and space $2^{2.95n}$
- [MiVo10]: SVP can be solved in deterministic time 4^n and space 2^n.
- [ADRS15]: 2^n-Time/Space algorithm.

Gaussian Sampling [ADRS15]

- **Th**: One can output $2^{n/2}$ random lattice points from any discrete Gaussian distribution in time/space $2^{n+o(n)}$.
  - The algorithm is somewhat simpler than AKS, and can be viewed as a randomized version of Mordell's algorithm.

Structure

- It can be viewed is a sieve algorithm.
  - Sample Gaussian lattice points where the s parameter gets smaller and smaller.

The Key Lemma

- **Lemma**: Let L be a lattice. If u and v chosen from the discrete Gaussian distribution over (L/2,s), then u+v conditioned over $u+v \in L$ has discrete Gaussian distribution over $(L, s\sqrt{2})$.
- Proof: Simple calculations.

Remark

- It is normal that it works for s beyond the smoothing parameter: for such s, discrete Gaussians behave like continuous Gaussians.
  - u+v has discrete Gaussian distribution over $(L/2, s\sqrt{2})$ [MP13].
  - Then conditioned over $u+v \in L$, it becomes the discrete Gaussian distribution over $(L, s\sqrt{2})$

Remark

- What is surprising is that it works for arbitrary s.
  - If we restrict s to beyond the smoothing parameter, then it works for any overlattice, not just L/2.
  - But without restriction, only L/2 seems to work!

Overview

- Let $\bar{L}=2^{-1}L$ and $G=(\mathbf{Z}/2\mathbf{Z})^n$ then $\bar{L}/L \simeq G$.
- Suppose you can generate Gaussian samples over (L,s). Then you can generate samples over $(\bar{L},s/2)$.
- Keep generating samples u and v over $(\bar{L},s/2)$ until $u+v\in L$. Then this $u+v$ has discrete Gaussian distribution over $(L,s/\sqrt{2})$.
    - Then s has been reduced by $\sqrt{2}$!
- [ADRS15] makes this much more efficient.