Classical and quantum 3 and 4-sieves to solve SVP with low memory

Johanna Loyer and André Chailloux

Inria Paris

PQCrypto. August 16, 2023.

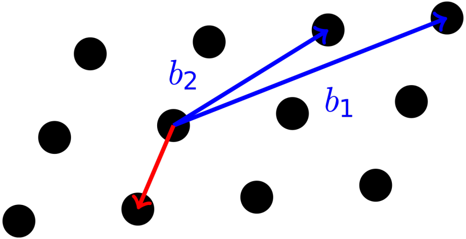
Lattice and SVP

Lattice

Given a basis $B = (\vec{b_1}, ..., \vec{b_d})$, the lattice $\mathcal{L}$ generated by B is the set of all integer linear combinations of its basis vectors: $\mathcal{L}(B) = \left\{ \sum_{i=1}^{d} z_i \vec{b_i},\ z_i \in \mathbb{Z} \right\}$.

Shortest Vector Problem (SVP)

Given a lattice $\mathcal{L}$, find the shortest non-zero vector $\vec{v} \in \mathcal{L}$.



Motivation to solve SVP

Cryptography

- NP-hard problem, hard in average, believed to be quantum-resistant.
- Problems derived from SVP: LWE, SIS, NTRU...
- Cryptosystems based on them: Kyber, Dilithium, Falcon (NIST standardization), ...

Motivation to solve SVP

Cryptography

- NP-hard problem, hard in average, believed to be quantum-resistant.
- Problems derived from SVP: LWE, SIS, NTRU...
- Cryptosystems based on them: Kyber, Dilithium, Falcon (NIST standardization), ...

Cryptanalysis

- Broken if we can find a reduced basis of the lattice.
- BKZ algorithm returns a reduced basis using an SVP-solver.

$\Rightarrow$ The security of these cryptosystems directly relies on the complexity of solving SVP.

Overview

1. Lattice sieving

2. Filtering
   New code for filtering

3. Framework to solve SVP and complexity results

1. Lattice sieving

Sieving

Heuristic: Lattice vectors are uniformly random in $\mathbb{R}^d$.

- Random vectors of norm $\leq R$ are w.h.p. of norm R.
- Validated by experiments [NV08] for long vectors.

Sieving

Heuristic: Lattice vectors are uniformly random in $\mathbb{R}^d$.

- Random vectors of norm $\leq R$ are w.h.p. of norm R.
- Validated by experiments [NV08] for long vectors.

Sieving step

Input: list L of N lattice vectors of norm at most R ; $\gamma < 1$.
Output: list L_{out} of N lattice vectors of norm at most $\gamma R < R$.

Sieving

Heuristic: Lattice vectors are uniformly random in $\mathbb{R}^d$.

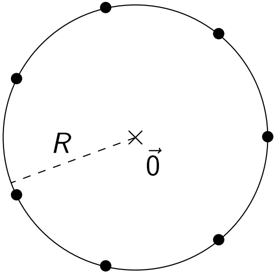
- Random vectors of norm $\leq R$ are w.h.p. of norm R.
- Validated by experiments [NV08] for long vectors.

Sieving step

Input: list L of N lattice vectors of norm at most R ; $\gamma < 1$.

Output: list L_{out} of N lattice vectors of norm at most $\gamma R < R$.

Initialization:
Generate N lattice vectors
of norm $\leq R$
(Klein's algorithm)



Sieving

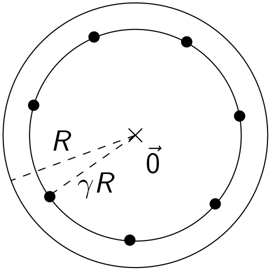
Heuristic: Lattice vectors are uniformly random in $\mathbb{R}^d$.

- Random vectors of norm $\leq R$ are w.h.p. of norm R.
- Validated by experiments [NV08] for long vectors.

Sieving step

Input: list L of N lattice vectors of norm at most R ; $\gamma < 1$.
Output: list L_{out} of N lattice vectors of norm at most $\gamma R < R$.

After 1 iteration:
vectors of norm at most γR



Sieving

Heuristic: Lattice vectors are uniformly random in $\mathbb{R}^d$.

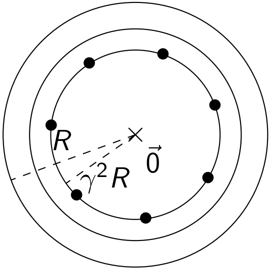
- Random vectors of norm $\leq R$ are w.h.p. of norm R.
- Validated by experiments [NV08] for long vectors.

Sieving step

Input: list L of N lattice vectors of norm at most R ; $\gamma < 1$.

Output: list L_{out} of N lattice vectors of norm at most $\gamma R < R$.

After 2 iterations:
vectors of norm at most $\gamma^2 R$



Sieving

Heuristic: Lattice vectors are uniformly random in $\mathbb{R}^d$.

- Random vectors of norm $\leq R$ are w.h.p. of norm R.
- Validated by experiments [NV08] for long vectors.

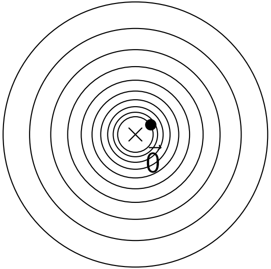
Sieving step

Input: list L of N lattice vectors of norm at most R ; $\gamma < 1$.

Output: list L_{out} of N lattice vectors of norm at most $\gamma R < R$.

After poly(d) iterations:
norm at most $\gamma^{\mathrm{poly}(d)} R$.

Short vector found!

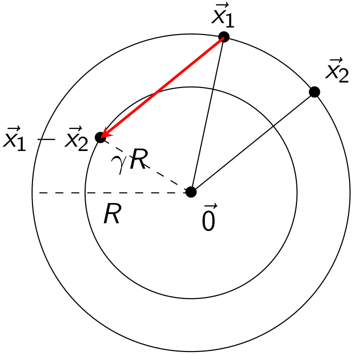


Sieving step

Nguyen-Vidick sieve [NV08] (2-sieve)

```
for $(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2) \in L \times L$ :
    if $\|\vec{\mathbf{x}}_1 - \vec{\mathbf{x}}_2\| \leqslant \gamma R$ :
        add $\vec{\mathbf{x}}_1 - \vec{\mathbf{x}}_2$ to $L_{out}$
```

Sphere of dimension d and radius R:

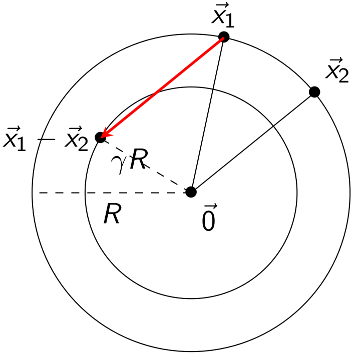


Sieving step

Nguyen-Vidick sieve [NV08] (2-sieve)

```
for $(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2) \in L \times L$ :
    if $\|\vec{\mathbf{x}}_1 - \vec{\mathbf{x}}_2\| \leqslant \gamma R$ :
        add $\vec{\mathbf{x}}_1 - \vec{\mathbf{x}}_2$ to $L_{out}$
```

Sphere of dimension d and radius R:



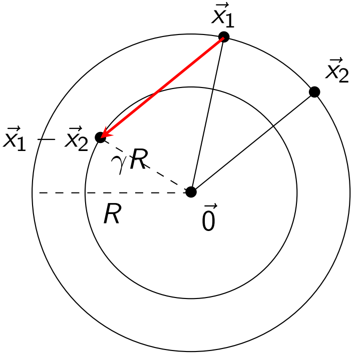
If $\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2 \in \mathcal{L}$ then $\vec{\mathbf{x}}_1 - \vec{\mathbf{x}}_2 \in \mathcal{L}$.

Sieving step

Nguyen-Vidick sieve [NV08] (2-sieve)

```
for $(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2) \in L \times L$ :
    if $\|\vec{\mathbf{x}}_1 - \vec{\mathbf{x}}_2\| \leqslant \gamma R$ :
        add $\vec{\mathbf{x}}_1 - \vec{\mathbf{x}}_2$ to $L_{out}$
```

Sphere of dimension d and radius R:



If $\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2 \in \mathcal{L}$ then $\vec{\mathbf{x}}_1 - \vec{\mathbf{x}}_2 \in \mathcal{L}$.

Condition of reduction:
For $\gamma = 1$, $\|\vec{\mathbf{x}}_1\| = \|\vec{\mathbf{x}}_2\| = R$,

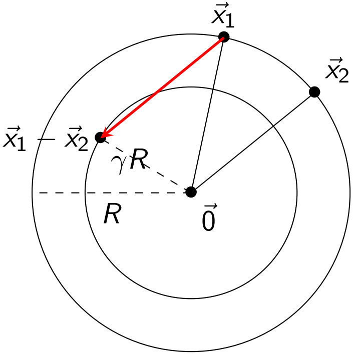
$$\|\vec{\mathbf{x}}_1 - \vec{\mathbf{x}}_2\| \leqslant \gamma R$$
$$\Leftrightarrow \text{Angle}(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2) \leqslant \tfrac{\pi}{3}$$

Sieving step

Nguyen-Vidick sieve [NV08] (2-sieve)

```
for $(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2) \in L \times L$ :
    if $\|\vec{\mathbf{x}}_1 - \vec{\mathbf{x}}_2\| \leqslant \gamma R$ :
        add $\vec{\mathbf{x}}_1 - \vec{\mathbf{x}}_2$ to $L_{out}$
```

Sphere of dimension d and radius R:



If $\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2 \in \mathcal{L}$ then $\vec{\mathbf{x}}_1 - \vec{\mathbf{x}}_2 \in \mathcal{L}$.

Condition of reduction:
For $\gamma = 1$, $\|\vec{\mathbf{x}}_1\| = \|\vec{\mathbf{x}}_2\| = R$,

$$\|\vec{\mathbf{x}}_1 - \vec{\mathbf{x}}_2\| \leqslant \gamma R$$
$$\Leftrightarrow \text{Angle}(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2) \leqslant \tfrac{\pi}{3}$$
$$\Leftrightarrow \tfrac{1}{R^2} \langle \vec{\mathbf{x}}_1 | \vec{\mathbf{x}}_2 \rangle \geq \tfrac{1}{2}.$$

Sieving step

3-sieve

```
for $(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \vec{\mathbf{x}}_3) \in L^3$ :
    if $\|\vec{\mathbf{x}}_1 + \vec{\mathbf{x}}_2 + \vec{\mathbf{x}}_3\| \leqslant \gamma R$ :
        add $\vec{\mathbf{x}}_1 + \vec{\mathbf{x}}_2 + \vec{\mathbf{x}}_3$ to $L_{out}$
```

4-sieve

```
for $(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \vec{\mathbf{x}}_3, \vec{\mathbf{x}}_4) \in L^4$
    if $\|\vec{\mathbf{x}}_1 + \vec{\mathbf{x}}_2 + \vec{\mathbf{x}}_3 + \vec{\mathbf{x}}_4\| \leqslant \gamma R$ :
        add $\vec{\mathbf{x}}_1 + \vec{\mathbf{x}}_2 + \vec{\mathbf{x}}_3 + \vec{\mathbf{x}}_4$ to $L_{out}$
```

k-sieve

```
for $(\vec{\mathbf{x}}_1, ..., \vec{\mathbf{x}}_k) \in L^k$
    if $\|\vec{\mathbf{x}}_1 + ... + \vec{\mathbf{x}}_k\| \leqslant \gamma R$ :
        add $\vec{\mathbf{x}}_1 + ... + \vec{\mathbf{x}}_k$ to $L_{out}$
```

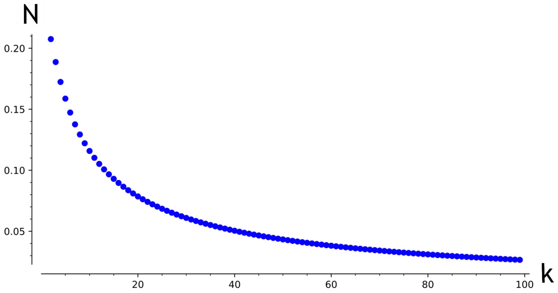
Minimal size of the list L

Sieving step

Input: List of N lattice vectors

Output: List of N reduced lattice vectors

$\Rightarrow$ We need that N reduced vectors are computable from the N input vectors.



Notation: $2^{xd+o(d)}$

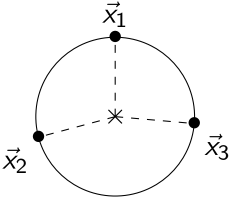
k	Memory N	Time (naive) N^k
2	0.208	0.415
3	0.189	0.566
4	0.173	0.690
5	0.159	0.794
6	0.147	0.884

Configurations

Configurations

Configuration

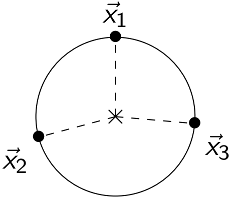
A k-tuple $(\vec{\mathbf{x}}_1, \dots, \vec{\mathbf{x}}_k)$ satisfies configuration $C = (C_{ij})_{i,j} \in \mathbb{R}^{k \times k}$ iff. $\langle \vec{\mathbf{x}}_i | \vec{\mathbf{x}}_j \rangle \leq C_{ij}$.



Configurations

Configuration

A k-tuple $(\vec{\mathbf{x}}_1, ..., \vec{\mathbf{x}}_k)$ satisfies configuration $C = (C_{ij})_{i,j} \in \mathbb{R}^{k \times k}$ iff. $\langle \vec{\mathbf{x}}_i | \vec{\mathbf{x}}_j \rangle \leq C_{ij}$.



Valid configuration C: $(\vec{\mathbf{x}}_1, ..., \vec{\mathbf{x}}_k)$ satisfies $C \Rightarrow \|\vec{\mathbf{x}}_1 + ... + \vec{\mathbf{x}}_k\| \leq \gamma R$

Configurations

Configuration problem

Find all tuples $(\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_k) \in L_1 \times \ldots \times L_k$ satisfying the valid configuration C.

$\Rightarrow$

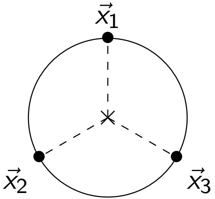
***k*-sieve problem**

Find all reduced vectors $\sum_{i=1}^{k} \vec{\mathbf{x}}_i$, $\vec{\mathbf{x}}_i \in L$ of norm $\leq \gamma R$.

Configurations

Balanced configuration

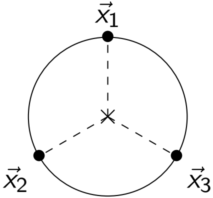
- Fix $C_{ij} = -1/k$ for $i \neq j$
- The most common configuration for reducing k-tuples
  $\Rightarrow$ Minimizes the list size $|L|$.



Configurations

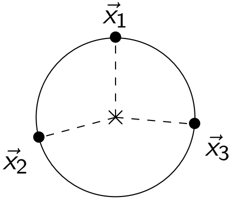
Balanced configuration

- Fix $C_{ij} = -1/k$ for $i \neq j$
- The most common configuration for reducing k-tuples
  $\Rightarrow$ Minimizes the list size $|L|$.



Any valid configuration

- Rarer configurations ⇒ require longer list
- Tuples can be easier to find.



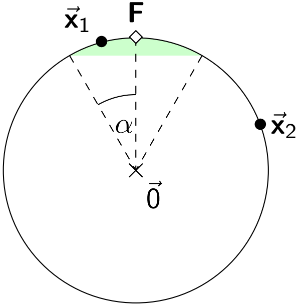
2. Filtering

Filtering

Locality Sentitive Filter

A **filter** of center $\mathbf{F} \in \mathbb{R}^d$ and angle $\alpha \in [0, \pi/2]$ maps a vector $\vec{x}$ to a boolean value:

- 1 if Angle$(\vec{x}, \mathbf{F}) \leqslant \alpha$,
- 0 else.

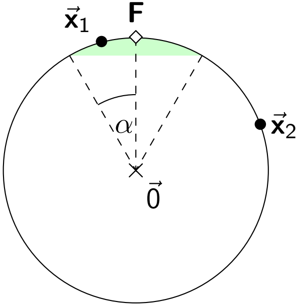


Filtering

Locality Sentitive Filter

A **filter** of center $\mathbf{F} \in \mathbb{R}^d$ and angle $\alpha \in [0, \pi/2]$ maps a vector $\vec{x}$ to a boolean value:

- 1 if Angle$(\vec{x}, \mathbf{F}) \leqslant \alpha$,
- 0 else.



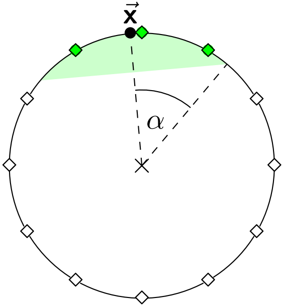
Each filter is associated with a set that we can fill with vectors.

Filtering - Random Product Code

Random Product Code (RPC) of parameters (d, m, B)

$$\mathfrak{C} = Q \cdot (\mathfrak{C}_1 \times \cdots \times \mathfrak{C}_m) \subset \mathbb{R}^d$$

- $\mathfrak{C}_1, \ldots, \mathfrak{C}_m$: sets of B vectors in $\mathbb{R}^{d/m}$ sampled unif. & indep. random of norm $\sqrt{1/m}$
- Q uniformly random rotation over $\mathbb{R}^d$



Codewords $\diamond$

- Uniformly distributed over the sphere
- Each codeword = center of one filter
- Decode $\vec{\mathbf{x}}$ in efficient time (subexp. or poly)

Filtering - Solving SVP

2-sieve

For each vector: search a reducing vector within the whole list L.

2-sieve with filtering [BDGL16]

1. Generate the filters ▷ Sample a Random Product Code
2. Add each vector to its filters of angle at most α. ▷ List decoding algorithm
3. For each vector : search a reducing vector within its filters.

Filtering - Solving SVP

2-sieve

For each vector: search a reducing vector within the whole list L.

2-sieve with filtering [BDGL16]

1. Generate the filters $\triangleright$ Sample a Random Product Code
2. Add each vector to its filters of angle at most α. $\triangleright$ List decoding algorithm
3. For each vector : search a reducing vector within its filters.
   - Classically or by Grover's search

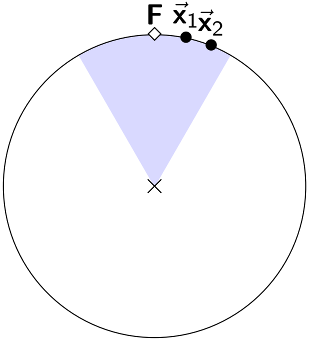
Time complexity (for minimal memory $N = 2^{0.208d+o(d)}$):

Classical 2-sieve: $2^{0.415d+o(d)}$ Quantum 2-sieve: $2^{0.312d+o(d)}$

With filtering: $2^{0.292d+o(d)}$ With filtering: $2^{0.265d+o(d)}$

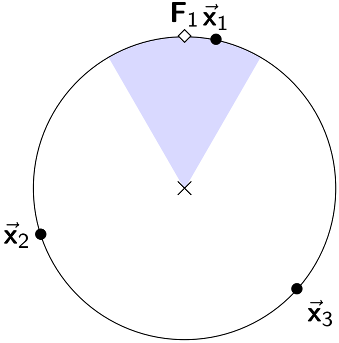
Filtering strategy for the 2-sieve

Constraint: $\langle \vec{\mathbf{x}}_1 | \vec{\mathbf{x}}_2 \rangle \geq \frac{1}{2}$



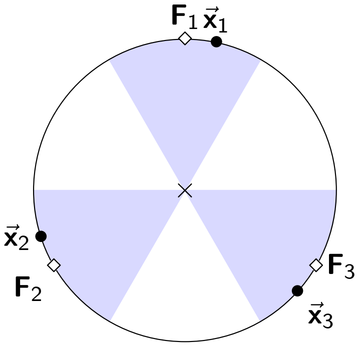
New tailored filtering for the k-sieve

Constraints: $\langle \vec{\mathbf{x}}_i | \vec{\mathbf{x}}_j \rangle \leq C_{ij}$



New tailored filtering for the *k*-sieve

Constraints: $\langle \vec{\mathbf{x}}_i | \vec{\mathbf{x}}_j \rangle \leq C_{ij}$



3. Framework for the k-sieve

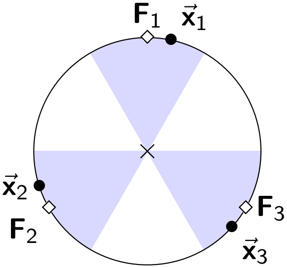
Framework

k-sieve framework to solve SVP

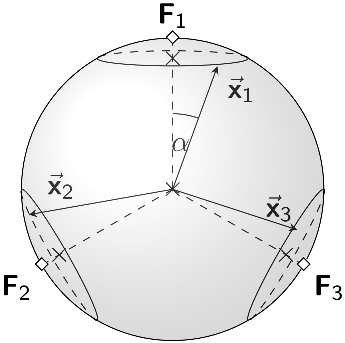
Input: list L of N lattice vectors, parameters k, angle α, configuration C

Output: list L_{out} of N reduced lattice vectors

1. Generate the tuple-filters. **Prefilter** L: add each $\vec{\mathbf{x}} \in L$ to its nearest filter.
2. For each tuple-filter: **Find all solutions** satisfying C within the tuple-filter.
3. Repeat 1. and 2. until $|L_{out}| = N$.

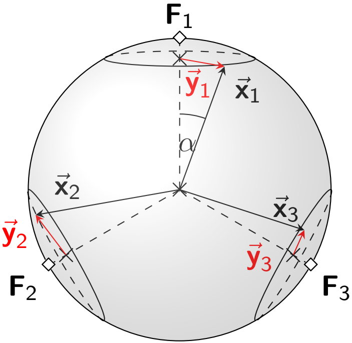


Residual vectors



Search for a tuple $(\vec{\mathbf{x}}_1, ..., \vec{\mathbf{x}}_k)$ satisfying configuration C

Residual vectors



Search for a tuple $(\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_k)$ satisfying configuration C

$\Leftrightarrow$

Search for their residual vectors $(\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_k)$ satisfying configuration $C'_{C,\alpha}$

Framework

k-sieve framework to solve SVP

Input: N lattice vectors, k, α, C
Output: N reduced lattice vectors

1. Prefilter L.
2. For each tuple-filter: **Find all solutions**
3. Repeat.

Subroutine **Find all solutions** within a tuple-filter

Input: Lists $L_1, \ldots, L_k$ of residual vectors, configuration C'.
Output: all tuples $(\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_k) \in L_1 \times \ldots \times L_k$ that satisfy C'.

Framework

k-sieve framework to solve SVP

Input: N lattice vectors, k, α, C
Output: N reduced lattice vectors

1. Prefilter L.
2. For each tuple-filter: **Find all solutions**
3. Repeat.

Subroutine **Find all solutions** within a tuple-filter

Input: Lists $L_1, \ldots, L_k$ of residual vectors, configuration C'.
Output: all tuples $(\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_k) \in L_1 \times \ldots \times L_k$ that satisfy C'.

- Classic 3-sieve
- Quantum 3-sieve
- Classic 4-sieve
- Quantum 4-sieve

Classic 3-sieve — Subroutine

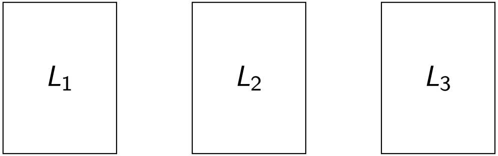
Configuration problem

Input: Lists L_1, L_2, L_3, configuration C'

Output: All the tuples $(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3) \in L_1 \times L_2 \times L_3$ satisfying configuration C'

$(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3)$ satisfies C'

$$\Leftrightarrow \begin{cases} \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_2 \rangle & \leq C'_{12} \\ \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_3 \rangle & \leq C'_{13} \\ \langle \vec{\mathbf{y}}_2 | \vec{\mathbf{y}}_3 \rangle & \leq C'_{23} \end{cases}$$



L_1 L_2 L_3

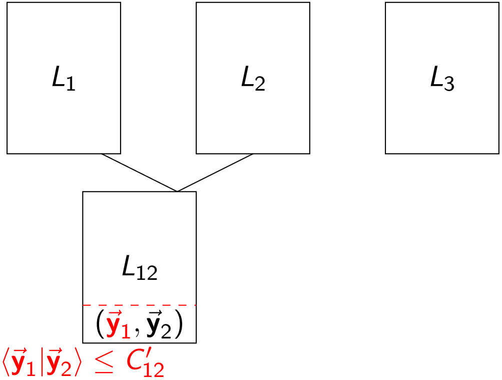
Classic 3-sieve — Subroutine

Configuration problem

Input: Lists L_1, L_2, L_3, configuration C'
Output: All the tuples $(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3) \in L_1 \times L_2 \times L_3$ satisfying configuration C'

$(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3)$ satisfies C'

$$\Leftrightarrow \begin{cases} \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_2 \rangle & \leq C'_{12} \\ \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_3 \rangle & \leq C'_{13} \\ \langle \vec{\mathbf{y}}_2 | \vec{\mathbf{y}}_3 \rangle & \leq C'_{23} \end{cases}$$


L_1
L_2
L_3
L_{12}
$(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2)$
$\langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_2 \rangle \leq C'_{12}$

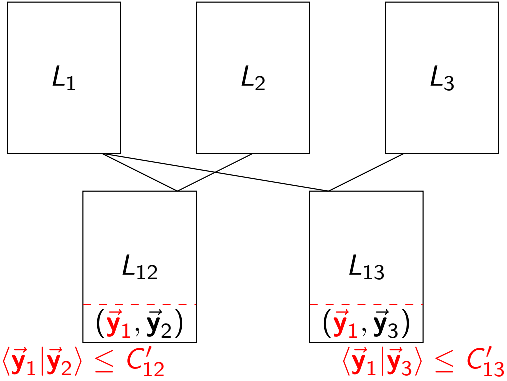
Classic 3-sieve — Subroutine

Configuration problem

Input: Lists L_1, L_2, L_3, configuration C'
Output: All the tuples $(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3) \in L_1 \times L_2 \times L_3$ satisfying configuration C'

$(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3)$ satisfies C'

$$\Leftrightarrow \begin{cases} \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_2 \rangle & \leq C'_{12} \\ \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_3 \rangle & \leq C'_{13} \\ \langle \vec{\mathbf{y}}_2 | \vec{\mathbf{y}}_3 \rangle & \leq C'_{23} \end{cases}$$


L_1
L_2
L_3
L_{12}
L_{13}
$(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2)$
$(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_3)$
$\langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_2 \rangle \leq C'_{12}$
$\langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_3 \rangle \leq C'_{13}$

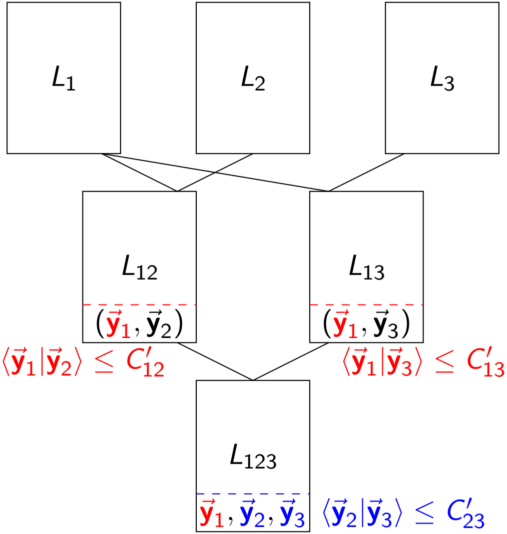
Classic 3-sieve — Subroutine

Configuration problem

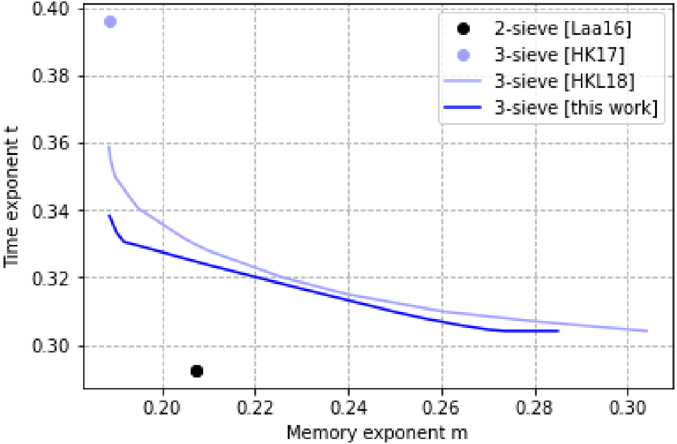
Input: Lists L_1, L_2, L_3, configuration C'
Output: All the tuples $(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3) \in L_1 \times L_2 \times L_3$ satisfying configuration C'

$(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3)$ satisfies C'

$$\Leftrightarrow \begin{cases} \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_2 \rangle & \leq C'_{12} \\ \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_3 \rangle & \leq C'_{13} \\ \langle \vec{\mathbf{y}}_2 | \vec{\mathbf{y}}_3 \rangle & \leq C'_{23} \end{cases}$$


L_1
L_2
L_3
L_{12}
$(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2)$
$\langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_2 \rangle \leq C'_{12}$
L_{13}
$(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_3)$
$\langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_3 \rangle \leq C'_{13}$
L_{123}
$\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3$
$\langle \vec{\mathbf{y}}_2 | \vec{\mathbf{y}}_3 \rangle \leq C'_{23}$

Classic 3-sieve

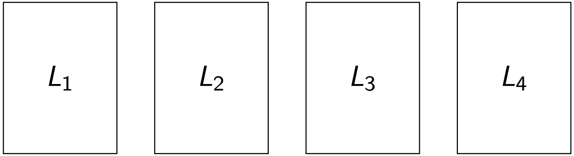


Classic 4-sieve — Subroutine

Configuration problem

Input: Lists L_1, L_2, L_3, L_4, configuration C'

Output: All the tuples $(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3, \vec{\mathbf{y}}_4) \in L_1 \times L_2 \times L_3 \times L_4$ satisfying configuration C'



L_1

L_2

L_3

L_4

$(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3, \vec{\mathbf{y}}_4)$ satisfies C'

$$\Leftrightarrow \begin{cases} \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_2 \rangle & \leq C'_{12} \\ \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_3 \rangle & \leq C'_{13} \\ \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_4 \rangle & \leq C'_{14} \\ \langle \vec{\mathbf{y}}_2 | \vec{\mathbf{y}}_3 \rangle & \leq C'_{23} \\ \langle \vec{\mathbf{y}}_2 | \vec{\mathbf{y}}_4 \rangle & \leq C'_{24} \\ \langle \vec{\mathbf{y}}_3 | \vec{\mathbf{y}}_4 \rangle & \leq C'_{34} \end{cases}$$

Classic 4-sieve — Subroutine

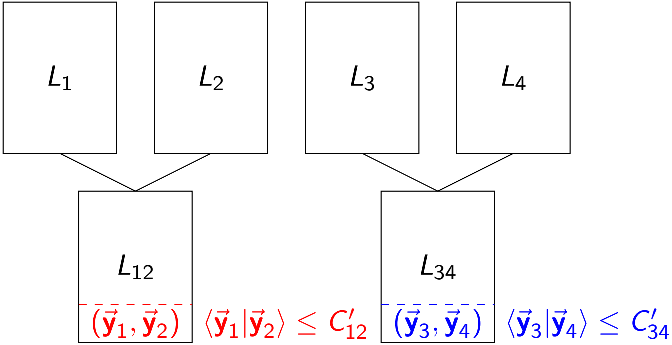
Configuration problem

Input: Lists L_1, L_2, L_3, L_4, configuration C'

Output: All the tuples $(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3, \vec{\mathbf{y}}_4) \in L_1 \times L_2 \times L_3 \times L_4$ satisfying configuration C'

$(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3, \vec{\mathbf{y}}_4)$ satisfies C'

$$\Leftrightarrow \begin{cases} \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_2 \rangle & \leq C'_{12} \\ \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_3 \rangle & \leq C'_{13} \\ \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_4 \rangle & \leq C'_{14} \\ \langle \vec{\mathbf{y}}_2 | \vec{\mathbf{y}}_3 \rangle & \leq C'_{23} \\ \langle \vec{\mathbf{y}}_2 | \vec{\mathbf{y}}_4 \rangle & \leq C'_{24} \\ \langle \vec{\mathbf{y}}_3 | \vec{\mathbf{y}}_4 \rangle & \leq C'_{34} \end{cases}$$


L_1
L_2
L_3
L_4
L_{12}
L_{34}
$(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2)$
$\langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_2 \rangle \leq C'_{12}$
$(\vec{\mathbf{y}}_3, \vec{\mathbf{y}}_4)$
$\langle \vec{\mathbf{y}}_3 | \vec{\mathbf{y}}_4 \rangle \leq C'_{34}$

Classic 4-sieve — Subroutine

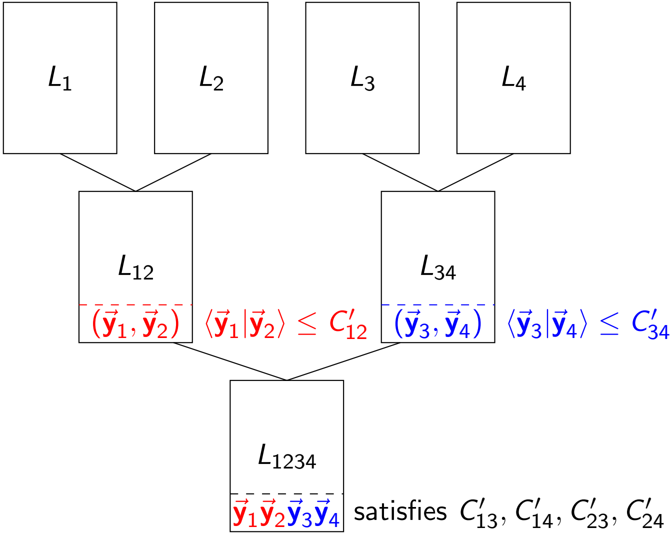
Configuration problem

Input: Lists L_1, L_2, L_3, L_4, configuration C'

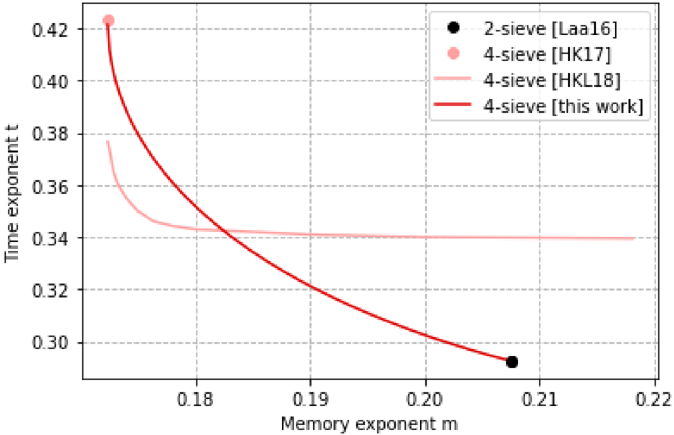
Output: All the tuples $(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3, \vec{\mathbf{y}}_4) \in L_1 \times L_2 \times L_3 \times L_4$ satisfying configuration C'

$(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3, \vec{\mathbf{y}}_4)$ satisfies C'

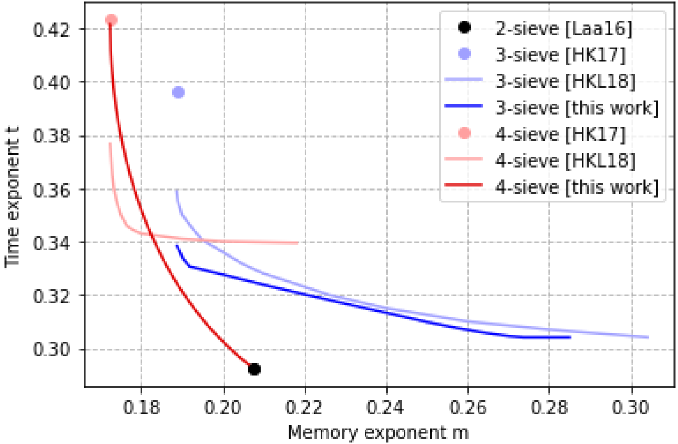
$$\Leftrightarrow \begin{cases} \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_2 \rangle & \leq C'_{12} \\ \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_3 \rangle & \leq C'_{13} \\ \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_4 \rangle & \leq C'_{14} \\ \langle \vec{\mathbf{y}}_2 | \vec{\mathbf{y}}_3 \rangle & \leq C'_{23} \\ \langle \vec{\mathbf{y}}_2 | \vec{\mathbf{y}}_4 \rangle & \leq C'_{24} \\ \langle \vec{\mathbf{y}}_3 | \vec{\mathbf{y}}_4 \rangle & \leq C'_{34} \end{cases}$$


L_1
L_2
L_3
L_4
L_{12}
L_{34}
$(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2)$
$\langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_2 \rangle \leq C'_{12}$
$(\vec{\mathbf{y}}_3, \vec{\mathbf{y}}_4)$
$\langle \vec{\mathbf{y}}_3 | \vec{\mathbf{y}}_4 \rangle \leq C'_{34}$
L_{1234}
$\vec{\mathbf{y}}_1 \vec{\mathbf{y}}_2 \vec{\mathbf{y}}_3 \vec{\mathbf{y}}_4$
satisfies $C'_{13}, C'_{14}, C'_{23}, C'_{24}$

Classic 4-sieve



Classic k-sieves



Quantum 3-sieve — Subroutine

$|\psi_{L_1}\rangle \qquad |\psi_{L_2}\rangle \qquad |\psi_{L_3}\rangle$

Quantum 3-sieve — Subroutine

$$|\psi_{L_1}\rangle \qquad |\psi_{L_2}\rangle \qquad |\psi_{L_3}\rangle$$

$$||$$

$$\frac{1}{\sqrt{|L_1|}} \sum_{\vec{\mathbf{y}}_1 \in L_1} |\mathbf{i}_{\vec{\mathbf{y}}_1}\rangle |\vec{\mathbf{y}}_1\rangle$$

Quantum 3-sieve — Subroutine

$$\begin{tikzcd}
{|\psi_{L_1}\rangle} \arrow[d, phantom, "\parallel"] & {|\psi_{L_2}\rangle} \arrow[d, "\text{Grover}"] & {|\psi_{L_3}\rangle} & \\
{\frac{1}{\sqrt{|L_1|}}\sum_{\vec{\mathbf{y}}_1 \in L_1} |\mathbf{i}_{\vec{\mathbf{y}}_1}\rangle |\vec{\mathbf{y}}_1\rangle} & {|\psi_{L_2(\vec{\mathbf{y}}_1)}\rangle} & & {\langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_2 \rangle \leq C'_{12}}
\end{tikzcd}$$

Quantum 3-sieve — Subroutine

$$\begin{tikzcd}
|\psi_{L_1}\rangle \arrow[d, equal] & |\psi_{L_2}\rangle \arrow[d, "\text{Grover}"] & |\psi_{L_3}\rangle \arrow[d, "\text{Grover}"] \\
\frac{1}{\sqrt{|L_1|}} \sum_{\vec{\mathbf{y}}_1 \in L_1} |\mathbf{i}_{\vec{\mathbf{y}}_1}\rangle |\vec{\mathbf{y}}_1\rangle & |\psi_{L_2(\vec{\mathbf{y}}_1)}\rangle & |\psi_{L_3(\vec{\mathbf{y}}_1)}\rangle
\end{tikzcd}$$

$$\langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_2 \rangle \leq C'_{12}$$

$$\langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_3 \rangle \leq C'_{13}$$

Quantum 3-sieve — Subroutine

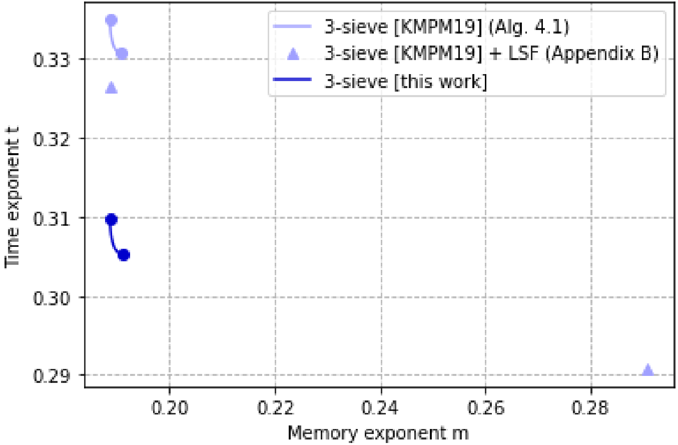
$$\begin{tikzcd}
\left|\psi_{L_1}\right\rangle \arrow[d, equal] & \left|\psi_{L_2}\right\rangle \arrow[d, "\text{Grover}"] & \left|\psi_{L_3}\right\rangle \arrow[d, "\text{Grover}"] & \\
\frac{1}{\sqrt{|L_1|}} \sum_{\vec{\mathbf{y}}_1 \in L_1} \left|\mathrm{i}_{\vec{\mathbf{y}}_1}\right\rangle \left|\vec{\mathbf{y}}_1\right\rangle & \left|\psi_{L_2(\vec{\mathbf{y}}_1)}\right\rangle \arrow[d, equal] & \left|\psi_{L_3(\vec{\mathbf{y}}_1)}\right\rangle \arrow[d, "\text{Grover}"] & \begin{array}{c} \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_2 \rangle \leq C'_{12} \\ \langle \vec{\mathbf{y}}_1 | \vec{\mathbf{y}}_3 \rangle \leq C'_{13} \end{array} \\
& \frac{1}{\sqrt{|L_2(\vec{\mathbf{y}}_1)|}} \sum_{\vec{\mathbf{y}}_2 \in L_2(\vec{\mathbf{y}}_1)} \left|\mathrm{i}_{\vec{\mathbf{y}}_2}\right\rangle \left|\vec{\mathbf{y}}_2\right\rangle & \left|\psi_{L_3(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2)}\right\rangle & \langle \vec{\mathbf{y}}_2 | \vec{\mathbf{y}}_3 \rangle \leq C'_{23}
\end{tikzcd}$$

Quantum 3-sieve — Subroutine

$$\left|\psi_{L_1}\right\rangle\left|\psi_{L_2(\vec{\mathbf{y}}_1)}\right\rangle\left|\psi_{L_3(\vec{\mathbf{y}}_1,\vec{\mathbf{y}}_2)}\right\rangle$$

- Apply amplitude amplification
- Measure and get a solution $(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3)$
- Repeat to find all the solutions in $L_1 \times L_2 \times L_3$

Quantum 3-sieve

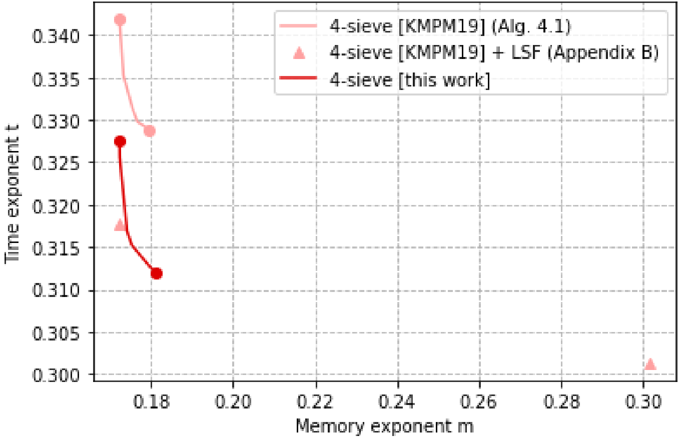


Quantum 4-sieve — Subroutine

$$\left|\psi_{L_1}\right\rangle\left|\psi_{L_2(\vec{\mathbf{y}}_1)}\right\rangle\left|\psi_{L_3(\vec{\mathbf{y}}_1,\vec{\mathbf{y}}_2)}\right\rangle\left|\psi_{L_4(\vec{\mathbf{y}}_1,\vec{\mathbf{y}}_2)}\right\rangle$$

- Apply amplitude amplification
- Measure and get a solution $(\vec{\mathbf{y}}_1, \vec{\mathbf{y}}_2, \vec{\mathbf{y}}_3, \vec{\mathbf{y}}_4)$
- Repeat to find all the solutions in $L_1 \times L_2 \times L_3 \times L_4$

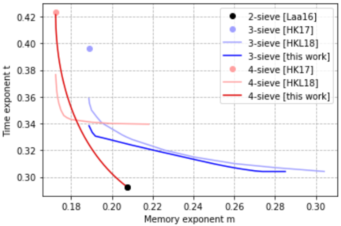
Quantum 4-sieve

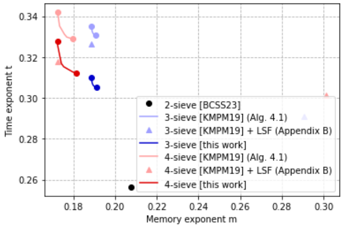


Conclusion

This work:

- Improves the 3-sieve trade-offs
- New trade-offs for the 4-sieves





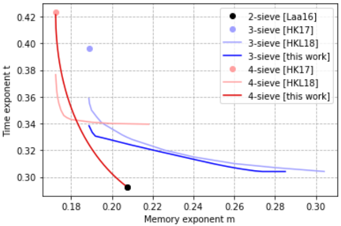
Further research:

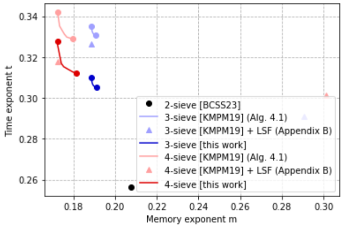
- k-sieve for $k > 4$
- Mix our prefiltering with inner filtering as in [HKL18, KMPM19]
- **Classical**: Find the optimal merging tree
- **Quantum**: k-sieve via quantum walks.

Conclusion

This work:

- Improves the 3-sieve trade-offs
- New trade-offs for the 4-sieves





Further research:

- k-sieve for $k > 4$
- Mix our prefiltering with inner filtering as in [HKL18, KMPM19]
- **Classical**: Find the optimal merging tree
- **Quantum**: k-sieve via quantum walks.

Thank you for listening! Any questions?

References

[NV08] P.Q. Nguyen and T. Vidick
Sieve algorithms for the shortest vector problem are practical
doi.org/10.1515/JMC.2008.009

[BDGL16] A. Becker, L. Ducas, N. Gama and T. Laarhoven
New directions in nearest neighbor searching with applications to lattice sieving
ePrint 2015/1128

[HK17] G. Herold and E. Kirshanova
Improved algorithms for the approximate *k*-list problem in Euclidean norm
ePrint 2017/017

[HKL18] G. Herold, E. Kirshanova and T. Laarhoven
Speed-ups and time–memory trade-offs for tuple lattice sieving
ePrint 2017/1228

[KMPM19] E. Kirshanova, E. Mårtensson, E.W. Postlethwaite and S.R. Moulik
Quantum algorithms for the approximate *k*-list problem and their application to lattice sieving
ePrint 2019/1016