Concrete Analysis of Quantum Lattice Enumeration

Shi Bai[1] **Maya-Iggy van Hoof**[2] Floyd B. Johnson[1]
Tanja Lange[3] **Tran Ngo**[1]

[1]Florida Atlantic University, United States.

[2]Ruhr University Bochum, Germany.

[3]Eindhoven University of Technology, the Netherlands.

6 December 2023

Motivation

- Lattices are a popular choice for post-quantum cryptography.
- 2 popular generic attacks:
  - Sieving (faster, but exponential space).
  - **Enumeration** (slower, but polynomial space).
- Most literature is about asymptotics.
- Our work: concrete quantum look.

History of Work

[Bel13]	Belovs's quantum walk
[AK17, Mon18]	Quantum tree backtracking algorithm
[ADPS16, ABB+ 17, PLP16]	Quantum Enumeration is mentioned
[ANS18]	Presenting quantum lattice enumeration and its variations are discussed at a high level without going into detail about quantum circuits and resource estimates

Our work

Proposed a concrete implementation of Montanaro's algorithm for lattice enumeration, and its circuit resource estimates.

Quantum Algorithms

- Paper: gate level.
- This talk: overview.
- Quantum computers use quantum bits, **qubits**.

- We interact with qubits using quantum **gates**.

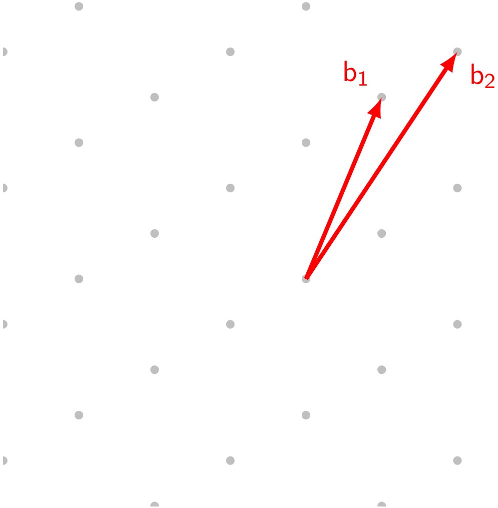
Quantum Algorithms

- Paper: gate level.
- This talk: overview.
- Quantum computers use quantum bits, **qubits**.
- More qubits $\rightarrow$ bigger quantum computer required.
- We interact with qubits using quantum **gates**.

Quantum Algorithms

- Paper: gate level.
- This talk: overview.
- Quantum computers use quantum bits, **qubits**.
- More qubits $\rightarrow$ bigger quantum computer required.
- We interact with qubits using quantum **gates**.
- Quantum efficiency: T-gates or T-depth.

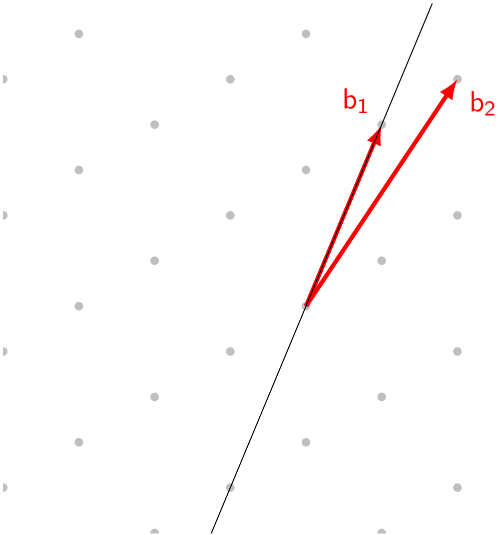
Enumeration



Visualization idea: Thijs Laarhoven.

Enumeration

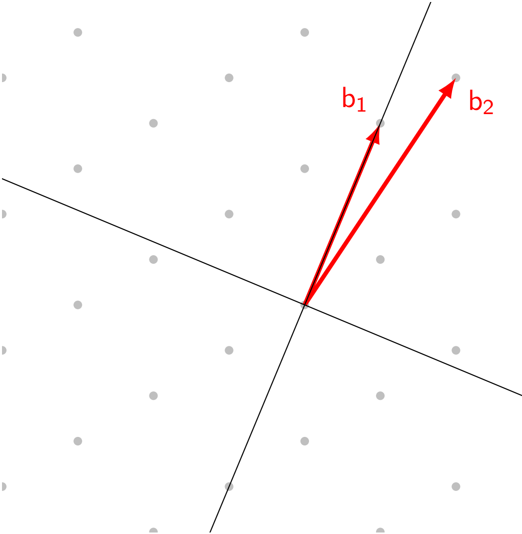
- Pick one direction.



Visualization idea: Thijs Laarhoven.

Enumeration

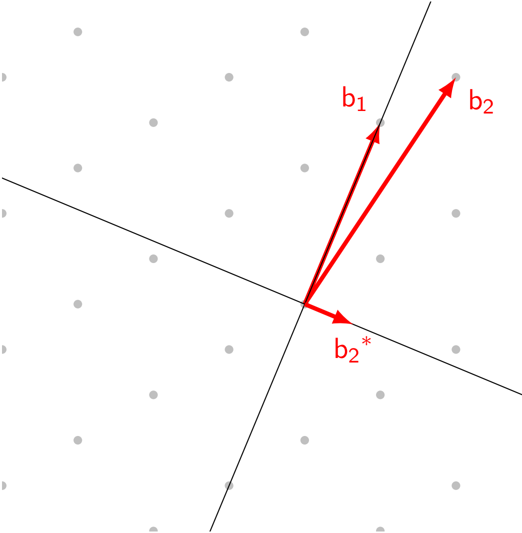
- Pick one direction.
- Consider directions orthogonal to it.



Visualization idea: Thijs Laarhoven.

Enumeration

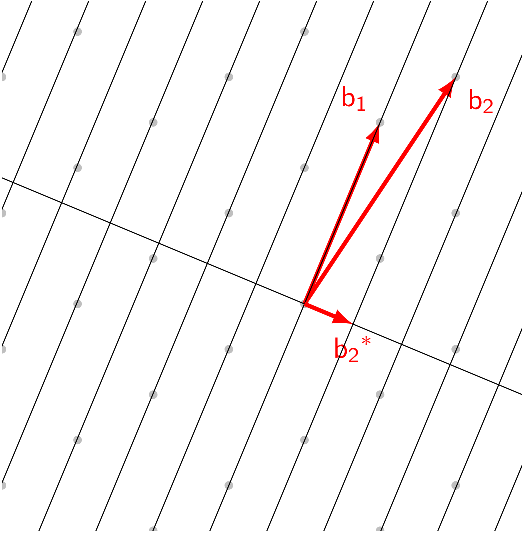
- Pick one direction.
- Consider directions orthogonal to it.
- Project other vector(s).



Visualization idea: Thijs Laarhoven.

Enumeration

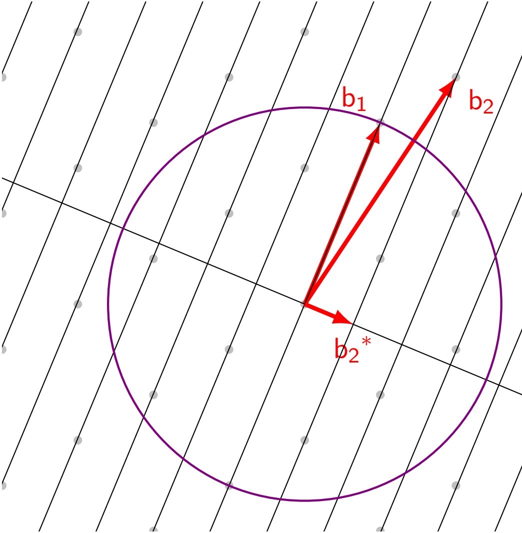
- Pick one direction.
- Consider directions orthogonal to it.
- Project other vector(s).
- Make a grid parallel to b_1 spaced by the length of ${b_2}^*$.



Visualization idea: Thijs Laarhoven.

Enumeration

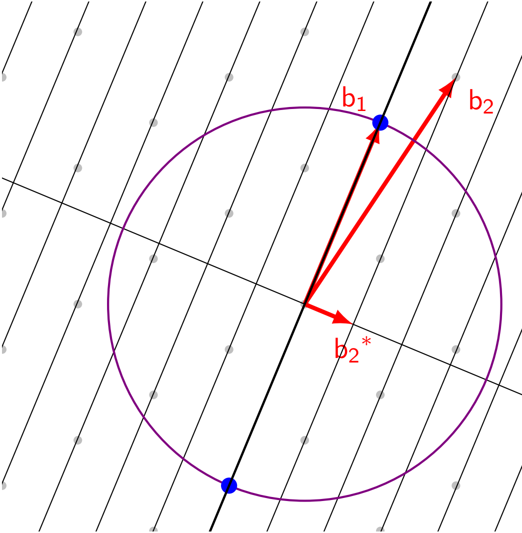
- Pick one direction.
- Consider directions orthogonal to it.
- Project other vector(s).
- Make a grid parallel to b_1 spaced by the length of $b_2{}^*$.
- Consider points within the sphere of radius $||b_1||$.



Visualization idea: Thijs Laarhoven.

Enumeration

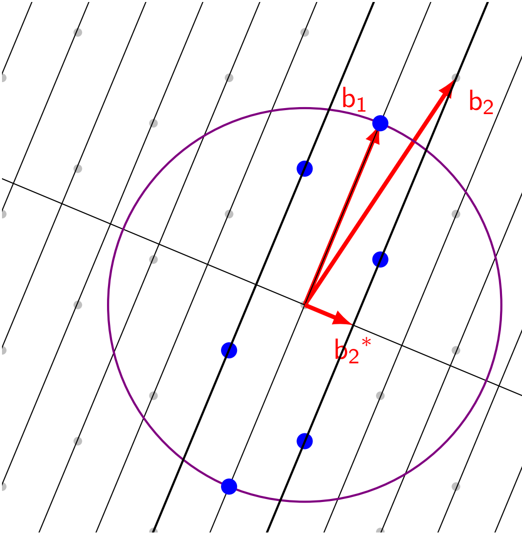
- Pick one direction.
- Consider directions orthogonal to it.
- Project other vector(s).
- Make a grid parallel to b_1 spaced by the length of $b_2{}^*$.
- Consider points within the sphere of radius $||b_1||$.
- For each multiple of $||b_2{}^*||$ find all lattice points on that line.



Visualization idea: Thijs Laarhoven.

Enumeration

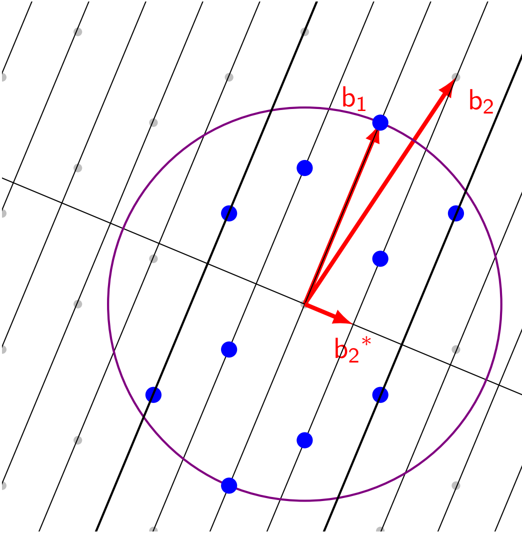
- Pick one direction.
- Consider directions orthogonal to it.
- Project other vector(s).
- Make a grid parallel to b_1 spaced by the length of $b_2{}^*$.
- Consider points within the sphere of radius $||b_1||$.
- For each multiple of $||b_2{}^*||$ find all lattice points on that line.



Visualization idea: Thijs Laarhoven.

Enumeration

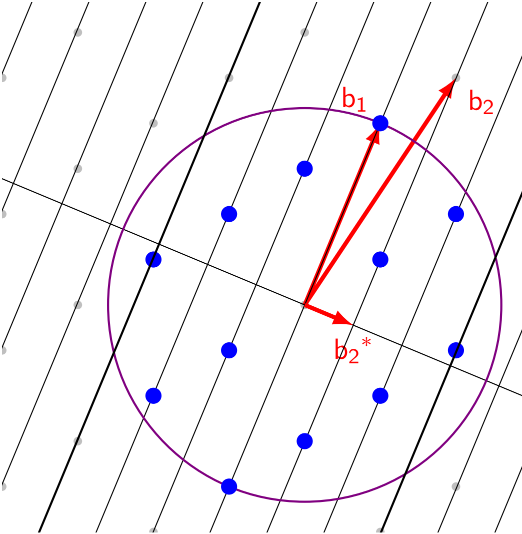
- Pick one direction.
- Consider directions orthogonal to it.
- Project other vector(s).
- Make a grid parallel to b_1 spaced by the length of b_2^*.
- Consider points within the sphere of radius $||b_1||$.
- For each multiple of $||b_2^*||$ find all lattice points on that line.



Visualization idea: Thijs Laarhoven.

Enumeration

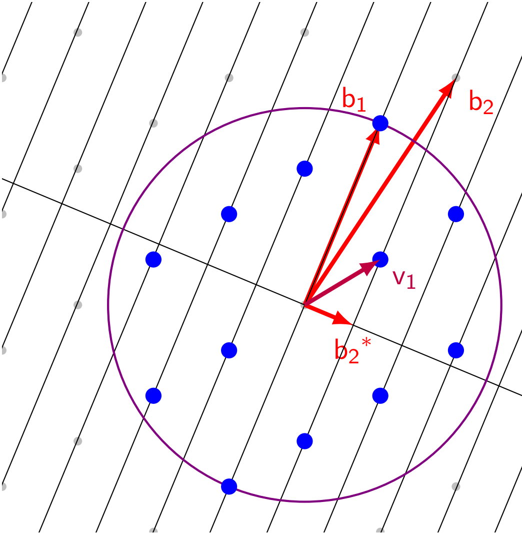
- Pick one direction.
- Consider directions orthogonal to it.
- Project other vector(s).
- Make a grid parallel to b_1 spaced by the length of b_2^*.
- Consider points within the sphere of radius $||b_1||$.
- For each multiple of $||b_2^*||$ find all lattice points on that line.
- Output the shortest vector in the sphere.



Visualization idea: Thijs Laarhoven.

Enumeration

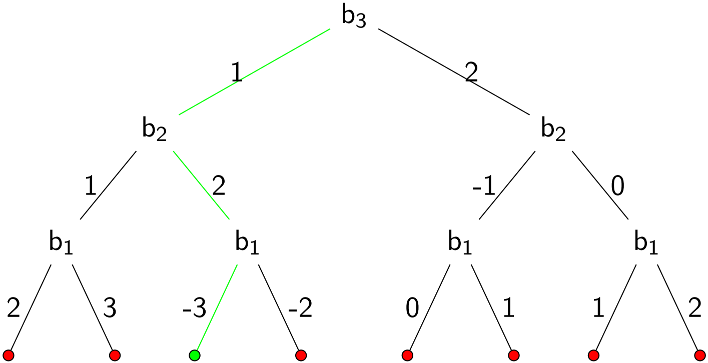
- Pick one direction.
- Consider directions orthogonal to it.
- Project other vector(s).
- Make a grid parallel to b_1 spaced by the length of b_2^*.
- Consider points within the sphere of radius $||b_1||$.
- For each multiple of $||b_2^*||$ find all lattice points on that line.
- Output the shortest vector in the sphere.



Visualization idea: Thijs Laarhoven.

Naive tree search

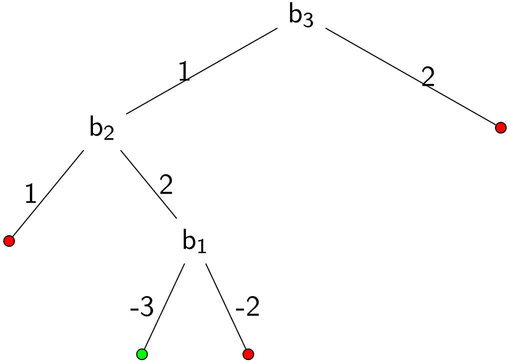
Our problem can be modelled as a tree.



Where the green node corresponds to the shortest vector $-3\mathbf{b}_1 + 2\mathbf{b}_2 + \mathbf{b}_3$.

Backtracking: idea

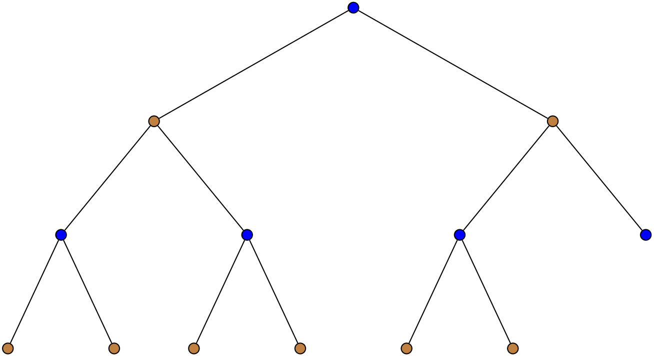
We need to find some way to reduce the number of nodes in the tree.



At each level call an oracle/predicate, this is called backtracking.

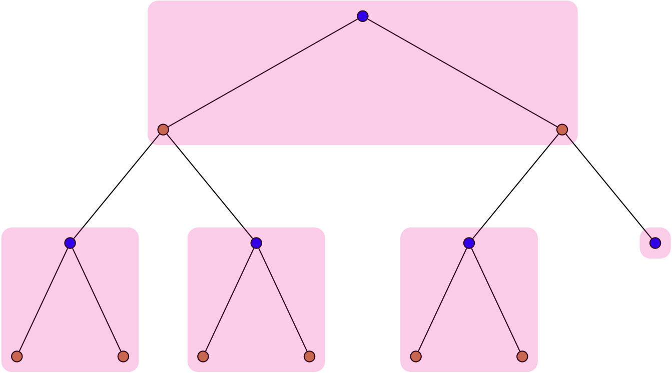
Quantum backtracking

Montanaro '18: split the tree into odd and even distance from the root.



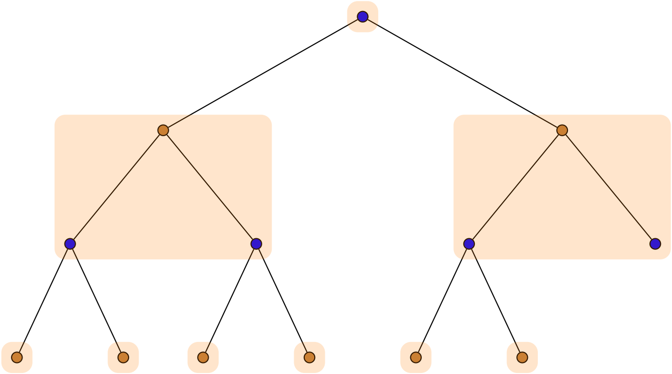
Quantum backtracking

Montanaro '18: split the tree into odd and even distance from the root.
Each step consists of going over the even distance nodes and their children.



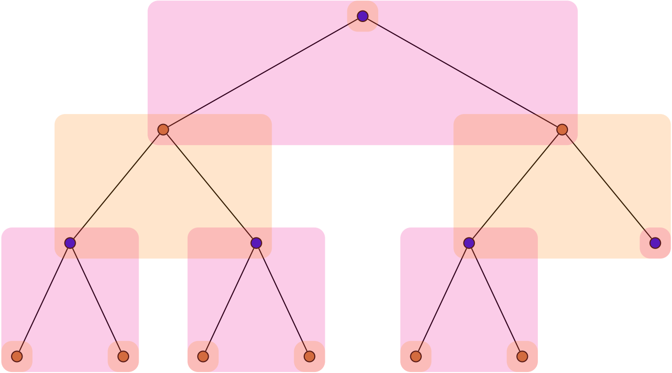
Quantum backtracking

Montanaro '18: split the tree into odd and even distance from the root.
Each step consists of going over the even distance nodes and their children.
And going over the odd distance nodes and their children (and the root).



Quantum backtracking

Montanaro '18: split the tree into odd and even distance from the root.
Each step consists of going over the even distance nodes and their children.
And going over the odd distance nodes and their children (and the root).
Part of our paper: circuit level implementation of this step, called $R_B R_A$.



Overall Circuit

Montanaro's algorithm uses the phase estimation of the operator $U = R_B R_A$ on the root of the substrees to detect a marked solution

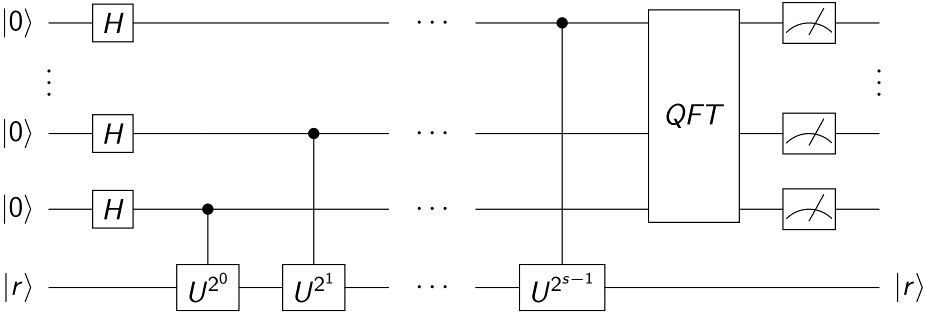


Figure: General circuit for phase estimation where $U := R_B R_A$.

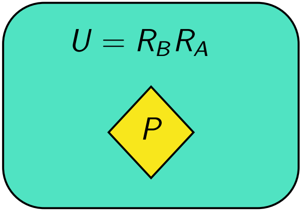
Our circuit design and its resource estimation

Overall Picture

The circuit design contains two main components

Overall Picture

The circuit design contains two main components

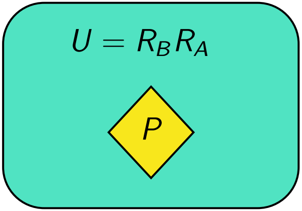


U: Quantum walk operator

P: Ensuring if the walk is correct

Overall Picture

The circuit design contains two main components



U: Quantum walk operator
P: Ensuring if the walk is correct

The predicate P will ensure the coefficients v_i satisfy

$$\sum_{j \geq n+1-l} \Big(\sum_{i \geq j} \mu_{i,j} \cdot v_i\Big)^2 \cdot \|\mathbf{b}_j^*\|^2 \leq R^2,$$

where $R, \mu_{ij}, \|\mathbf{b}_j^*\|$ are precomputed classically.

Predicate Design in a nutshell

$$\underbrace{\sum_{j \geq n+1-\ell}}_{(IV) Add_{FF}} \Big(\underbrace{\sum_{i \geq j}}_{(II) A_k} \underbrace{\mu_{i,j} \cdot v_i}_{(I)} \Big)^2 \underbrace{\cdot \|\mathbf{b}_j^*\|^2}_{(III) Mul_{FF}} \leq R^2,$$

Predicate Design in a nutshell

$$\underbrace{\sum_{j \geq n+1-\ell}}_{(IV) Add_{FF}} \Big(\underbrace{\sum_{i \geq j}}_{(II) A_k} \underbrace{\mu_{i,j} \cdot v_i}_{(I)} \Big)^2 \underbrace{\cdot \|\mathbf{b}_j^*\|^2}_{(III) Mul_{FF}} \leq R^2,$$

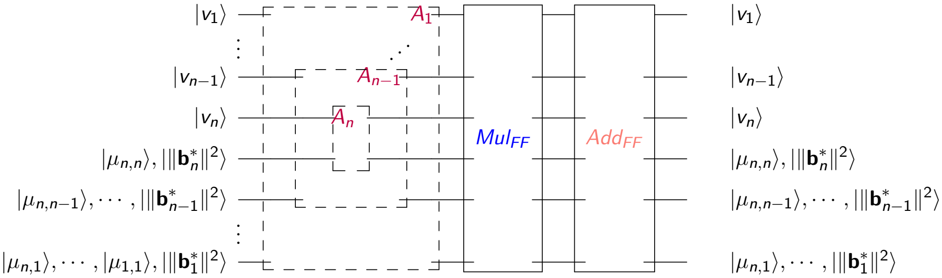


Figure: Components in the predicate circuit

Overall Circuit Design - Requirements

Overall Circuit Design - Requirements

n: $\dim(\mathcal{L})$ d: maximum degree of the tree B: bound of coefficients

Overall Circuit Design - Requirements

n: $\dim(\mathcal{L})$ d: maximum degree of the tree B: bound of coefficients

- Floating point precision for the enumeration:

Overall Circuit Design - Requirements

n: $\dim(\mathcal{L})$ d: maximum degree of the tree B: bound of coefficients

- Floating point precision for the enumeration:
  - We assume that $\mu_{i,j}, \|\mathbf{b}_i\|$ have similar magnitudes

Overall Circuit Design - Requirements

n: $\dim(\mathcal{L})$ d: maximum degree of the tree B: bound of coefficients

- Floating point precision for the enumeration:
  - We assume that $\mu_{i,j}, \|\mathbf{b}_i\|$ have similar magnitudes
  - The precision for enumeration is $p' \approx 0.3n$. The overall precision $p \approx 4p' + 4\log B + \log^2 n + 7\log n$

Overall Circuit Design - Requirements

n: $\dim(\mathcal{L})$ d: maximum degree of the tree B: bound of coefficients

- Floating point precision for the enumeration:
  - We assume that $\mu_{i,j}, \|\mathbf{b}_i\|$ have similar magnitudes
  - The precision for enumeration is $p' \approx 0.3n$. The overall precision $p \approx 4p' + 4\log B + \log^2 n + 7\log n$
- Bounds:

Overall Circuit Design - Requirements

n: $\dim(\mathcal{L})$ d: maximum degree of the tree B: bound of coefficients

- Floating point precision for the enumeration:
  - We assume that $\mu_{i,j}, \|\mathbf{b}_i\|$ have similar magnitudes
  - The precision for enumeration is $p' \approx 0.3n$. The overall precision $p \approx 4p' + 4\log B + \log^2 n + 7\log n$
- Bounds:
  - We know $\mathcal{T} \leq n^{n/(2e+o(n))}$

Overall Circuit Design - Requirements

n: $\dim(\mathcal{L})$ d: maximum degree of the tree B: bound of coefficients

- Floating point precision for the enumeration:
  - We assume that $\mu_{i,j}, \|\mathbf{b}_i\|$ have similar magnitudes
  - The precision for enumeration is $p' \approx 0.3n$. The overall precision $p \approx 4p' + 4\log B + \log^2 n + 7\log n$
- Bounds:
  - We know $\mathcal{T} \leq n^{n/(2e+o(n))}$
  - Better when the basis is preprocessed, e.g., in HKZ-reduced basis $d \approx n^{(\ln n)/4}$

Our results

n: $\dim(\mathcal{L})$ d: maximum degree of the tree B: bound of coefficients

Our results

n: $\dim(\mathcal{L})$ d: maximum degree of the tree B: bound of coefficients

Overall Resource Estimates

— T-depth —

$$32\sqrt{\mathcal{T}n} \cdot \left[16np(\log B + 2\log n + p^{0.158}) + O(n\log B) + 8d^2\log(d\sqrt{\mathcal{T}n}) + 4d^2\log d + O(d^2)\right]$$

— T-size —

$$32\sqrt{\mathcal{T}n} \cdot \left[8(d+1)(14pn^2(B+1) + O(n^2B)) + 8d^2\log(d\sqrt{\mathcal{T}n}) + 16d^2\log d + O(d^2)\right]$$

Our results

n: $\dim(\mathcal{L})$ d: maximum degree of the tree B: bound of coefficients

Our results

n: $\dim(\mathcal{L})$ d: maximum degree of the tree B: bound of coefficients

Practical Parameters

For cryptographic size n (e.g., $n \gtrapprox 400$), it's reasonable to model
$d \approx n, B \approx n^2, p \leq 3n$, $\log(\mathcal{T}) \approx cn \log n$

Our results

n: $\dim(\mathcal{L})$ d: maximum degree of the tree B: bound of coefficients

Practical Parameters

For cryptographic size n (e.g., $n \gtrapprox 400$), it's reasonable to model
$d \approx n, B \approx n^2, p \leq 3n$, $\log(\mathcal{T}) \approx cn \log n$

T-depth: $(128cn^3 \log n + O(n^{2.158}))\sqrt{\mathcal{T}n}$

Our results

n: $\dim(\mathcal{L})$ d: maximum degree of the tree B: bound of coefficients

Practical Parameters

For cryptographic size n (e.g., $n \gtrapprox 400$), it's reasonable to model $d \approx n, B \approx n^2, p \leq 3n$, $\log(\mathcal{T}) \approx cn \log n$

T-depth: $(128cn^3 \log n + O(n^{2.158}))\sqrt{\mathcal{T} n}$

T-size: $(10752n^6 + O(n^5))\sqrt{\mathcal{T} n}$

Heuristic 1: $d \approx n, B \approx n^2$

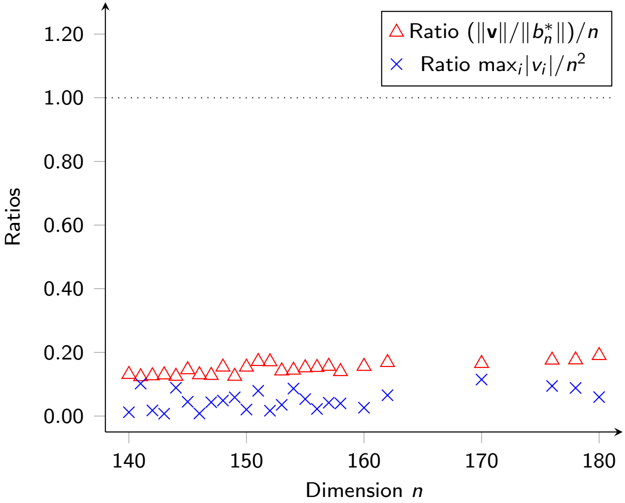


Figure: Bounds d and B, based on solved SVP Challenges.

Heuristic 2 - Bound $B \approx n^2$

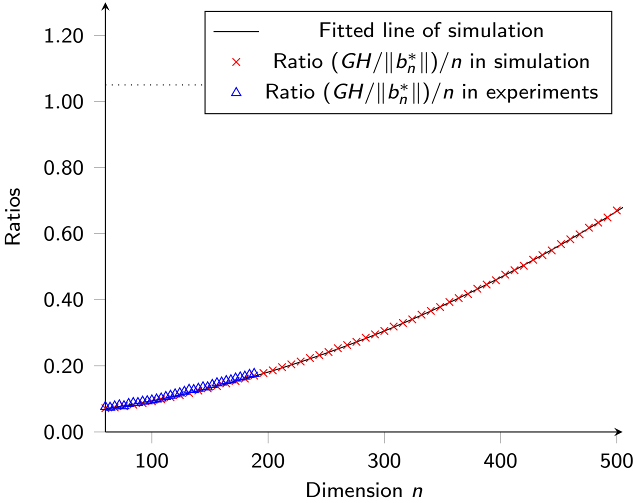


Figure: Bound B, based on BKZ experiments and simulations.

Heuristic 3 - $\mu_{i,j}$ have similar magnitude

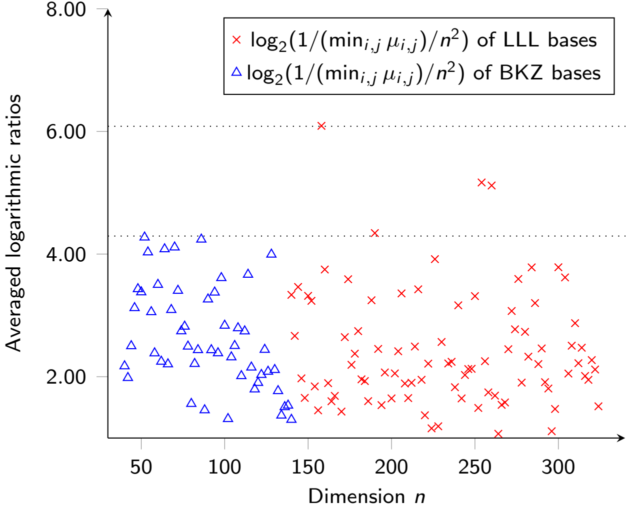


Figure: Experiments on the bound for $\mu_{i,j}$ in LLL/BKZ reduced basis.

Questions

- Can we make the phase estimation process parallel to reduce the depth?
- What are the resource estimates for extreme pruning?

The oracle receives questions now.
Do you have any questions?

References

- [ABB+17] Erdem Alkim, Nina Bindel, Johannes A. Buchmann, Özgür Dagdelen, Edward Eaton, Gus Gutoski, Juliane Krämer, and Filip Pawlega, *Revisiting TESLA in the quantum random oracle model*, 2017
- [ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe, *Post quantum key exchange: A New Hope*, 2016
- [AK17] Andris Ambainis and Martins Kokainis, *Quantum algorithm for tree size estimation, with applications to backtracking and 2-player games*, 2017
- [ANS18] Yoshinori Aono, Phong Q. Nguyen, and Yixin Shen, *Quantum lattice enumeration and tweaking discrete pruning*, ASIACRYPT 2018, Part I (Thomas Peyrin and Steven Galbraith, eds.), 2018
- [Bel13] Aleksandrs Belovs, *Quantum walks and electric networks*, 2013
- [Mon18] Ashley Montanaro, *Quantum-walk speedup of backtracking algorithms*, 2018
- [PLP16] Rafael Pino, Vadim Lyubashevsky, and David Pointcheval, *The whole is less than the sum of its parts: Constructing more efficient lattice-based akes*, 2016