From the shortest vector problem to the dihedral hidden subgroup problem

Curtis Bright

University of Waterloo

December 8, 2011

Reduction

- Roughly, "problem A reduces to problem B" means there is a way of solving A given some way of solving problem B.
- Intuitively, this says that B is at least as hard as A, and possibly harder, so we write

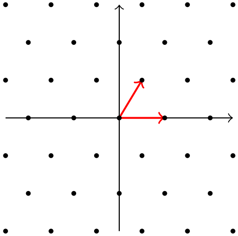
$$A \leq B.$$

Point lattices

- A *point lattice* is a discrete subset of $\mathbb{R}^n$ closed under addition and subtraction.
- A set of linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ generate a lattice L by taking their "integer span"

$$L = \left\{ \sum_{i=1}^{n} x_i \boldsymbol{b}_i : x_i \in \mathbb{Z} \right\}.$$

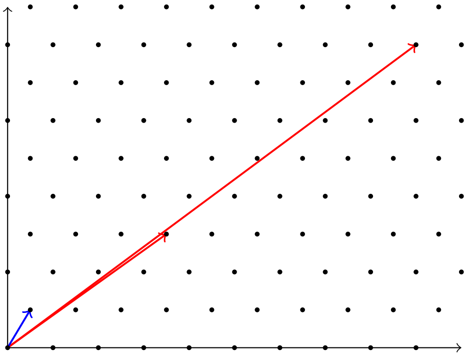
Example lattice in two dimensions:



Shortest vector problem (SVP)

- Given a lattice described by basis vectors, find a shortest nonzero vector in the lattice.

Example: Given the red vectors, find the blue vector:



Hardness of SVP

- In n dimensions, all known algorithms for the shortest vector problem run in exponential time in n.
- Using the max-norm the problem is known to be NP-hard, and is still suspected to be NP-hard using the usual Euclidean norm.

Approximate SVP

- Since it is too hard to solve SVP exactly, we will be content with finding an approximation to the shortest vector.
- Classically, the so-called LLL algorithm approximates the shortest vector to within a factor of $2^{(n-1)/2}$.

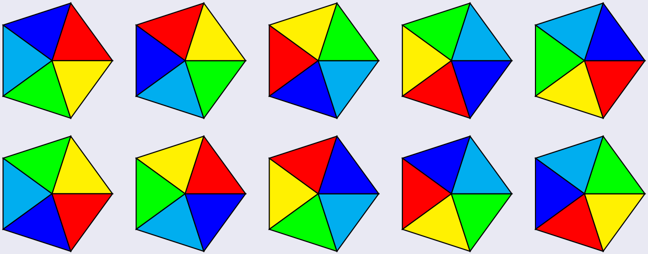
$f(n)$-unique-SVP

- As a special case of approximation, consider solving SVP in lattices which have an especially short shortest vector.
- Say a vector is $f(n)$-unique if it is a factor of $f(n)$ shorter than all nonparallel vectors.

If $f(n)$ is large enough, there is hope...

- The 2^n-unique-SVP is in P.
- The poly(n)-unique-SVP is in NP and coNP, so unlikely to be NP-hard.

The dihedral group of 10 elements—in colourized form



The hidden subgroup problem (HSP)

- An example function f on D_{10}:

$f($$) = 0$	$f($$) = 1$	$f($$) = 2$	$f($$) = 3$	$f($$) = 4$
$f($$) = 3$	$f($$) = 4$	$f($$) = 0$	$f($$) = 1$	$f($$) = 2$

- A function f on a group is said to *hide* a subgroup H if:
  - f is constant (say, 0) on the subgroup H
  - f is constant on the cosets of H and each coset has a distinct value
- The *hidden subgroup problem* is to find H with as few queries to f as possible.

Solving HSP by sampling cosets

- Construction of coset state:
  1. Construct superposition over all elements in group G:
  $$\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle$$
  2. Query f in an extra register:
  $$\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle |f(g)\rangle$$
  3. Measure second register, say with result $f(g)$:
  $$\frac{1}{\sqrt{|H|}} \sum_{h \in H} |gh\rangle$$
- Repeat this procedure to construct other coset samples, and use these to solve HSP somehow.

Dihedral coset problem (DCP)

- Find the constant d, given a collection of states of the form

$$\frac{1}{\sqrt{2}}|0\rangle|x\rangle + \frac{1}{\sqrt{2}}|1\rangle|x+d\rangle \text{ for random } x,$$

  where arithmetic is done mod n.
- These can be thought of as cosets states of order 2 subgroups of D_{2n}.
- If we could solve dihedral HSP by coset sampling, we could use that procedure to solve this problem, so:

$$\text{DCP} \leq \text{dihedral HSP by coset sampling}$$

Two-point problem

- Find the constant $\boldsymbol{d}$, given a collection of states of the form

$$\frac{1}{\sqrt{2}}|0\rangle|\boldsymbol{x}\rangle + \frac{1}{\sqrt{2}}|1\rangle|\boldsymbol{y}\rangle,$$

  for random $\boldsymbol{x}$, $\boldsymbol{y}$ integer vectors with entries in $[0, M)$ and $\boldsymbol{x} - \boldsymbol{y} = \boldsymbol{d}$.
- Consider the function f which views $\boldsymbol{x}$ as a "base $2M$" integer:

$$f(\boldsymbol{x}) := \sum_{i=1}^{n} x_i (2M)^{i-1}$$

Two-point problem continued

- Applying f to the second register,

$$\frac{1}{\sqrt{2}}|0\rangle|f(\boldsymbol{x})\rangle + \frac{1}{\sqrt{2}}|1\rangle|f(\boldsymbol{y})\rangle$$

  is a valid input to the DCP.
- Its output (slightly modified) read in base $2M$ gives us $\boldsymbol{d}$.
- So if we could solve DCP we could solve the two-point problem:

$$\text{two-point problem} \leq \text{DCP}$$

poly(n)-unique-SVP setup

- Create a superposition of points in $\mathbb{Z}^n$,

$$|\boldsymbol{Z}\rangle := \frac{1}{\sqrt{M^n}} \sum_{\boldsymbol{x}} |\boldsymbol{x}\rangle$$

  where $\boldsymbol{x}$ has entries in $[0, M)$ for some large M.

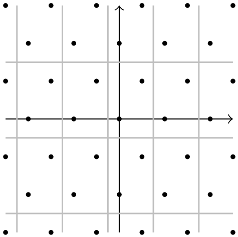
- Create a superposition of lattice points by applying

$$f(\boldsymbol{x}) := \sum_{i=1}^{n} x_i \boldsymbol{b}_i,$$

  where $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ generate the lattice.

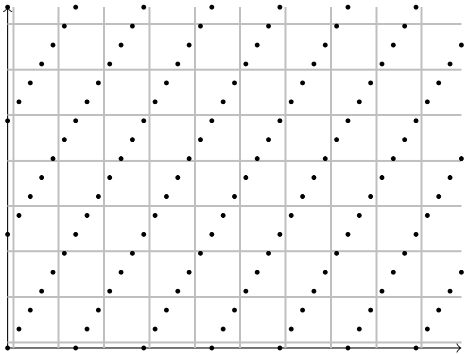
Partitioning space into cubes

- Suppose we could partition $\mathbb{R}^n$ into cubes such that every cube has two points whose difference is the shortest vector.
- Define a unique cube labeling function g and apply g to our lattice superposition.
- After measuring the result, the state collapses to a superposition of two points whose difference is the shortest vector.



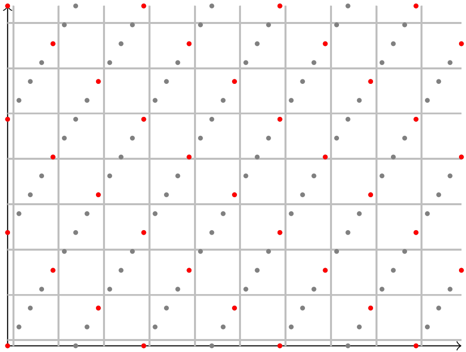
Approximate partitioning

- If the shortest vector is poly(n)-unique, cubes which are roughly the size of the shortest vector will only contain vectors whose difference is a multiple of the shortest vector.



Approximate partitioning continued

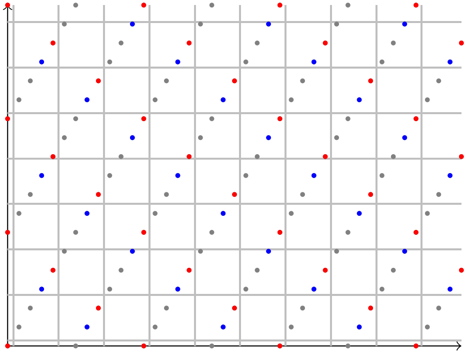
- But we want exactly two points in each cube.
- Idea: Scale up the basis vector $\boldsymbol{b}_1$ by a factor p to form a superposition over a subset of the points.



Approximate partitioning continued

- Shift the points over by $m\boldsymbol{b}_1$ based on a boolean variable t.
- To $\frac{1}{\sqrt{2}}|0\rangle|\boldsymbol{Z}\rangle + \frac{1}{\sqrt{2}}|1\rangle|\boldsymbol{Z}\rangle$ apply

$$f(t, \boldsymbol{x}) := x_1(p\boldsymbol{b}_1) + t(m\boldsymbol{b}_1) + \sum_{i=2}^{n} x_i\boldsymbol{b}_i.$$



Solving poly(n)-unique-SVP

- How to determine the proper cube size and shift amount?
  - Cube size needs to be within a factor of 2 of the shortest vector. This requires $O(n)$ cases to check, since LLL finds an $O(2^n)$ approximation to the shortest vector.
  - Determining the proper shift $m \in [0, p)$ requires $p \approx n^2$ cases to check.
- After partitioning and collapsing, with high probability the state will be a superposition of two vectors whose difference is the shortest vector.
- Then we can use a solution to the two-point problem to find the shortest vector:

$$\text{poly}(n)\text{-unique-SVP} \leq \text{two-point problem}$$

In summary

$$\begin{aligned} \text{poly}(n)\text{-unique-SVP} &\leq \text{two-point problem} \\ &\leq \text{dihedral coset problem} \\ &\leq \text{dihedral HSP by coset sampling} \end{aligned}$$