28 Jan 09

Hidden Subgroup Problem

Simon's problem and Period finding are two black-box problems for which quantum computers provide exponential speedups. What else can quantum computers do? These two problems have a similar structure, and it is useful to recognize this common ground, because it suggests further generalizations.

More specifically, Simon's problem and Period finding are both special cases of a problem that is naturally formulated in group-theoretic language: the Hidden Subgroup Problem (HSP). This is a black-box problem where we may regard the input to the function f to be an element of a group G, which is mapped into a set X, which we may take to be the set of m-bit strings

$$f: G \rightarrow X = \{0,1\}^m$$

The group G may be either finite or infinite, but we ordinarily assume it is finitely generated. That is, each element of G can be expressed as a product of a finite set of generating elements, where these generating elements may be used any number of times in the product, and in any order. The set X is finite.

We are promised that the function f is constant and distinct on the cosets of a subgroup $H \leq G$. This means that

$$f(g_1) = f(g_2) \text{ iff } g_2^{-1} g_1 \in H$$

(that is, $g_1 = g_2 h$ for some $h \in H$). The problem is to "find H" — to list a set of elements of G that generate H.

We may take the input size for the HSP to be an upper bound on $\log(|G/H|)$, the log of number of cosets (which is finite because X is finite).

The promise may restrict the hidden subgroup further by specifying additional properties of H. For example, in the case of Simon's problem,

$$G = Z_2^n \qquad H = Z_2 = \{0, a\}$$

The group Z_2 is the set $\{0, 1\}$, where the group operation is addition modulo 2. A product group $G_1 \times G_2$ is defined as the group of pairs of elements

$$G_1 \times G_2 = \{(g_1, g_2) \mid g_1 \in G_1,\ g_2 \in G_2\}$$

where the group operations are performed "in parallel"

$$(g_1, g_2) \circ (g_1', g_2') = (g_1 g_1', g_2 g_2')$$

Thus, the elements of Z_2^n (the product of n Z_2's) are n-bit strings of bits, where the group operation is bitwise XOR

$$(x_{n-1}, \ldots, x_0) \circ (y_{n-1}, \ldots, y_0)$$
$$= (x_{n-1} \oplus y_{n-1}, \ldots, x_0 \oplus y_0)$$

Each element is its own inverse (i.e., is order 2)

The promise in Simon's problem is:

$$f(x) = f(y) \text{ iff } x \oplus y \in \{0, a\} = H = \mathbb{Z}_2$$

Here $\{0, a\}$ is isomorphic to $\mathbb{Z}_2$. The problem is to determine how this $\mathbb{Z}_2$ is embedded in $G = \mathbb{Z}_2^n$ – i.e., to find its generator a. The number of possible embeddings is exponential in n. The number of cosets (and so the number of possible outputs in the set X) is 2^{n-1}, and its log is the input size.

Another example is period finding, for which

$$G = \mathbb{Z} \text{ and } H = r\mathbb{Z} = \{rK, K \in \mathbb{Z}\}$$

The group operation is addition, and we are promised that

$$f(x) = f(y) \text{ iff } x - y = r \cdot \text{integer} \in H$$

The problem is to find the generator of H, namely, the period r. The number of cosets of H is $|G/H| = r$, and an upper bound on its log is the input size.

Classically, the HSP has query complexity $\Omega(\sqrt{G/H})$; we need to query this many times in order to get the same output in response to two different queries w/ reasonable probability. This is exponential in the input size – the problem is hard classically.

But for <u>any</u> finitely generated abelian group, the problem is easy quantumly! It can be solved (with high success probability) using $O(\text{poly} \log |G/H|)$ queries and $O(\text{poly} \log |G/H|)$ additional computational steps.

Before we explain the algorithm, let's discuss another application:

Discrete Log Problem

Recall that if q is prime, then the group $\mathbb{Z}_q^*$ (multiplication mod q with elements $\{1, 2, \dots, q-1\}$) is cyclic. This means that $\mathbb{Z}_q^*$ is generated by a single element a; thus

$$\mathbb{Z}_q^* = \{a, a^2, a^3, \dots a^{q-1} = e\}.$$

Therefore any element $x \in \mathbb{Z}_q^*$ can be expressed in a unique way as the modular exponential

$$x = a^y \pmod q) \quad \text{where} \quad y \in \{0, 1, 2, \dots, q-2\}$$

The discrete log mod q with base a is the inverse of this function

$$x = a^y \pmod q \iff y = \text{dlog}_{q,a}(x)$$

A discrete log can be defined this way for any cyclic group and any generating element of the group.

Example: $q = 7$, $Z_q^* = \{1, 2, \ldots, 6\}$,
$a = 5$ (or $a = 3$) is generator

$y =$	0,	1,	2,	3,	4,	5
$x = 5^y \pmod 7 =$	1	5	4	6	2	3

Inverse function is

$x =$	1	2	3	4	5	6
$y = d\log_{7,5}(x) =$	0	4	5	2	1	3

The modular exponential is easy to compute classically (by repeated squaring), but the discrete log seems to be hard to compute - the modular exponential is a candidate one-way function. It is hard to invert because a^x jumps about in Z_q^* "pseudorandomly" as x varies (for at least some values of q)

There are applications of this one-way function in cryptography; for example,

Diffie-Hellman Key exchange

This protocol's security rests on the presumed hardness of computing the discrete logarithm. The objective is for Alice and Bob to generate a shared secret key that is not known by their adversary Eve.

1. A prime number q and a generating element $a \in Z_q^*$ are publicly anounced.

2. Alice generates a random element $x \in Z_q^*$ and keeps it secret. Bob generates random $y \in Z_q^*$ and keeps it secret.

3. Alice computes and announces $a^x \pmod q$
   Bob computes and announces $a^y \pmod q$

4. Alice computes $(a^y)^x = a^{xy} \pmod q$
   Bob computes $(a^x)^y = a^{xy} \pmod q$
   This is their final shared key

Alice and Bob can both compute the key because the modular exponential can be evaluated efficiently. The protocol is expected to be secure because even when a^x and a^y (but not x or y) are known, it is hard to compute a^{xy}. Of course, if Eve can compute the discrete log, she could break the protocol. E.g., knowing a^x, a^y she could find x and then compute $(a^y)^x$.

And a quantum computer can evaluate a discrete log by solving a HSP! Here is how. We would like to find

$$r = \mathrm{dlog}_{q,a}(x) \text{ — the value of } r \text{ such that } x = a^r \pmod q$$

We consider the function

$$f: Z \times Z \to Z_q^* \qquad f(y_1, y_2) = a^{y_1} x^{-y_2} \pmod q$$

When does f map two different inputs to the same output?

$$f(y_1, y_2) = a^{y_1 - ry_2} \pmod q$$
$$= f(z_1, z_2) = a^{z_1 - rz_2} \pmod q$$
$$\text{iff} \quad y_1 - ry_2 = z_1 - rz_2 \pmod{q-1}$$
$$\text{i.e.} \quad (y_1 - z_1) - r(y_2 - z_2) = 0 \pmod{q-1}$$

This means that we may think of the input to f as an element of the additive group

$$G = \mathbb{Z} \times \mathbb{Z}$$

where f is constant and distinct on the cosets of $H = \{(y_1, y_2) \mid y_1 = ry_2 \pmod{q-1}\}$

H is generated by the elements $(r, 1)$, $(q-1, 0)$. So if we find generators, we determine r.

For a HSP problem with finitely generated $\tilde{G}$, we may consider without loss of generality a corresponding problem with $G = \mathbb{Z}^n$, since there is a homomorphism mapping G onto $\tilde{G}$.

For example, in our original formulation of Simon's problem we considered $\tilde{G} = \mathbb{Z}_2^n$ where $\tilde{f}(x) = \tilde{f}(y)$ iff $x \oplus y \in \{0, a\}$. But instead we could consider

$G = \mathbb{Z}^n$ where $f(x) = f(y)$ iff $x - y \pmod 2 \in \{0, a\}$

This means that the hidden subgroup is

$$H = (2\mathbb{Z})^{n-1} \times (a\mathbb{Z})$$

That is the elements of H are
$\{(m_1 a_1, 2m_2 + m_1 a_2, 2m_3 + m_1 a_3, \ldots, 2m_n + m_1 a_n)\}$
where $m_1, m_2, \ldots, m_n \in \mathbb{Z}$, and we assume WLOG that $a_1 = 1$.

In general, it is useful that we can give geometrical interpretations to G and H. G is the n-dimensional hypercubic lattice, containing all order n-tuples of integers. The subgroup H can be regarded as a sublattice of $\mathbb{Z}^n$. This sublattice is spanned by a set of n linearly independent vectors $\{v_1, v_2, \ldots v_n\}$, each an element of $\mathbb{Z}^n$ (i.e., with integer entries). A general element of H is a linear combination of the generating vectors

$$x = \sum_{a=1}^{n} \alpha_a v_a$$

We may construct an $n \times n$ generator matrix for the lattice H whose rows are the generating vectors:

$$M = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \quad \text{and then} \quad H = \{x = \alpha M, \ \alpha \in \mathbb{Z}^n\}$$

where α is the row vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$. For fixed H, the generator matrix is not unique. We may make the replacement

$$\mathcal{M} \mapsto R\mathcal{M}$$

where R is an invertible integral matrix with $\det R = \pm 1$ (so that R^{-1} is also integral). Both M and RM are generators of the same lattice.

The quotient space G/H may be called the "unit cell" of the lattice. It contains all of the distinct ways to "shift" the lattice H by an element of G. We may say that $|G/H|$ is the volume of the unit cell, the number of points it contains. Note that

$$|G/H| = \det \mathcal{M}$$

(The linear transformation $\mathcal{M}$ inflates the cube $\{0,1\}^n$ to a region of volume $\det \mathcal{M}$).

Corresponding to the integral lattice H is its "dual lattice," denote $H^\perp$. The elements of $H^\perp$ are points in $\mathbb{R}^n$ that are orthogonal to all the vectors in H, modulo integers:

$$H^\perp = \{ K \in \mathbb{R}^n \text{ s.t. } K \cdot X \in \mathbb{Z} \text{ for all } X \in H \}$$

Equivalently, $\exp(2\pi i K \cdot X) = 1$ for $K \in H^\perp$ and $X \in H$.

$H^\perp$ is also a lattice (i.e. its elements are the span of a set of generating vectors with integer coefficients), but the components are not necessarily integer (although they are rational numbers). If $H^\perp$ is generated by vectors $u_1, u_2, \ldots, u_n$, then its generating matrix is

$$M^\perp = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} \quad \text{and} \quad H^\perp = \{K = \beta M^\perp,\ \beta \in \mathbb{Z}^n\}$$

We can choose the basis for the dual lattice such that $u_a \cdot v_b = \delta_{ab}$, in which case

$M^\perp M^T = I$. That means that, once we have found $M^\perp$, an easy computation determines M (matrix inversion of transpose of $M^\perp$). In the quantum algorithm for the abelian HSP, the quantum computation determines the generators of $H^\perp$ (i.e. the matrix $M^\perp$) and then finding the generators of H is easy (by matrix inversion)

For example, in the case of period finding, we have $G = \mathbb{Z}$, $H = r\mathbb{Z} = \{X = r\alpha, \alpha \in \mathbb{Z}\}$

and $H^\perp = \{K = \frac{\beta}{r}, \beta \in \mathbb{Z}\}$. In the quantum algorithm, we are promised that $r \le R$; thus rather than $\mathbb{Z}$ we used the quantum Fourier transform for $\mathbb{Z}_N$ where $N \geq R^2$. Fourier sampling then provided sufficient accuracy to determine an element β/r of $H^\perp$ with high success prob. After a few samples we could determine $\frac{1}{r}$, the generator of $H^\perp$, and hence r, the generator of H. We want to extend this idea from subgroups of $\mathbb{Z}$ to subgroups of $\mathbb{Z}^n$.

So, instead of $\mathbb{Z}^n$ suppose we consider $\mathbb{Z}_N^n$ for some sufficiently larger N. And to keep the discussion simple at first, suppose that H is actually a subgroup of $\mathbb{Z}_N^n$ rather than of $\mathbb{Z}^n$.

As in the period finding algorithm we query the black box with

$$\frac{1}{\sqrt{|G|}}\sum_{x\in G}|x\rangle \text{ and so obtain } \frac{1}{\sqrt{|G|}}\sum_{x\in G}|x\rangle\otimes|f(x)\rangle$$

where f is constant and distinct on the cosets of $H\subseteq G$. Were we to measure the output register, obtaining outcome $f(x_0)$, we would prepare, in the input register the uniform superposition of elements in the same coset as x_0, which is

$$|H,x_0\rangle = \frac{1}{\sqrt{|H|}}\sum_{x\in H}|x+x_0\rangle$$

This state has an important property: it is H-invariant. We may consider the unitary transformation U_y associated with an element $y\in G$ whose action is

$$U_y: |x\rangle \mapsto |x+y\rangle.$$

We note that, for $y\in H$, the "coset state" $|H,x_0\rangle$ is invariant under U_y, because

$$U_y|H,x_0\rangle = \frac{1}{\sqrt{H}}\sum_{x\in H}|y+x+x_0\rangle$$

and we may reparametrize the sum over x by replacing $x\mapsto x'-y$, thus obtaining

$$\frac{1}{\sqrt{H}}\sum_{x'\in H}|x'+x_0\rangle = |H,x_0\rangle$$

To appreciate the significance of H-invariance, note that if $|\psi\rangle$ obeys $U|\psi\rangle=|\psi\rangle$, then

$\tilde{U}|\tilde{\psi}\rangle = (VUV^{-1})|\tilde{\psi}\rangle = |\tilde{\psi}\rangle$, where $|\tilde{\psi}\rangle = V|\psi\rangle$

Now apply this identity to $U = U_y$ and $V = FT$, that is

$$V: |x\rangle \mapsto \frac{1}{\sqrt{|G|}} \sum_{K \in G^\perp} e^{2\pi i K\cdot x/N} |K\rangle$$

$$V^{-1}: |K\rangle \mapsto \frac{1}{\sqrt{|G|}} \sum_{x \in G} e^{-2\pi i K\cdot x/N} |x\rangle$$

Then $\tilde{U}_y |K\rangle = e^{2\pi i K\cdot y/N} |K\rangle$

i.e. $|K\rangle \xmapsto{V^{-1}} \frac{1}{\sqrt{|G|}} \sum_{x\in G} e^{-2\pi i K\cdot x/N} |x\rangle$

$$\xmapsto{U_y} \frac{1}{\sqrt{|G|}} \sum_{x\in G} e^{-2\pi i K\cdot x/N} |x+y\rangle$$

$$= e^{2\pi i K\cdot y/N} \frac{1}{\sqrt{|G|}} \sum_{x'\in G} e^{-2\pi i K\cdot x'/N} |x'\rangle$$

$$\xmapsto{V} e^{2\pi i K\cdot y/N} |K\rangle$$

Therefore, the state $|K\rangle$ is invariant under $\tilde{U}_y$ iff $\frac{1}{N} K\cdot y =$ integer — or for $y \in H$ iff $\frac{K}{N} \in H^\perp$. Thus, if a state is H-invariant, then in the Fourier basis its expansion contains the state $|K\rangle$ with a nonzero coefficient only if $\frac{1}{N} K \in H^\perp$

More explicitly, we compute

$$|H, x_0\rangle = \frac{1}{\sqrt{|H|}} \sum_{x\in H} |x_0 + x\rangle$$

$$\mapsto \frac{1}{\sqrt{|H|\cdot|G|}} \sum_{x\in H} \sum_{K\in G^\perp} e^{2\pi i K\cdot(x_0+x)/N} |K\rangle$$

Because of H-invariance, only $\frac{K}{N} \in H^\perp$ survives in the sum, and, for such K, $e^{2\pi i (K\cdot x)/N} = 1$, and we obtain

$$\frac{1}{\sqrt{|H^\perp|}} \sum_{k \in H^\perp} e^{2\pi i k \cdot x_0} |k\rangle$$

Therefore, if we "Fourier sample" — i.e. Fourier transform and then measure, the prob. distribution that governs the outcome is the uniform distribution on $H^\perp$. Once we have sampled from $H^\perp$ enough times, with high probability a generating set for $H^\perp$ will be found.

How many samples are enough (assuming now that G is finite — e.g. $G = \mathbb{Z}_N^n$ — and $H \leq G$)?

Suppose K is a group (either abelian or not), and m elements of K are chosen uniformly at random. If these m elements do not generate K, then they must be contained in some maximal proper subgroup $S \leq K$. (Proper means S is smaller than K, and "maximal" means we cannot add another element of K to S without generating all of K.) Any proper subgroup has order $|S| \leq |K|/2$, because the order of the subgroup must divide the order of K, ∴ and the probability that all m elements are in S is

$$\text{Prob (all } m \text{ in } S) = \left(\frac{|S|}{|K|}\right)^m$$

and therefore the prob that the m elements generate K is

$$\text{Prob } (m \text{ elements generate } K) \geq 1 - \sum_{S \in \max} \left(\frac{|S|}{|K|}\right)^m$$

where the sum is over maximal proper S

$$\geq 1 - (\#\max)\, 2^{-m}$$

that can be expressed as $\frac{\text{integer}}{\det M}$, where $\det M = |G/H|$, the number of cosets. In the formulation of the HSP, we are provided with an upper bound $|G/H| \leq R$, and N needs to be large enough to point to a unique rational number with denominator $\leq R$, with reasonable success probability.

In our discussion of period finding ($H^\perp = \frac{1}{r}\mathbb{Z}$) we noted that Fourier sampling yields a rational number y/N close to integer$/r$ with high prob:

$$\sum_k \text{Prob}\left(\left|\frac{y}{N} - \frac{k}{r}\right| \leq \frac{1}{2N}\right) \geq \frac{4}{\pi^2}$$

(so that choosing $N \geq R^2$ was good enough to determine a rational number with denom $< R$. If we fix the desired accuracy δ, then the part of the distribution lying outside the peaks decreases as N increases (an exercise):

$$\text{Prob}\left(\forall k\ \left|\frac{y}{N} - \frac{k}{r}\right| > \delta\right) \leq \frac{1}{N\delta}$$

The peak of the Fourier transform sharpens with increasing N, so that the prob of lying outside all peaks with half width δ scales like $1/N$.

When we sample $H^\perp$, we find an n-component vector, where each component should be determined to accuracy $1/R^2$ (where $|G/H| < R$). The probability of success in finding all n components to this accuracy is where (# max) denotes the total # of maximal proper subgroups.

If K is abelian, we can count the maximal proper subgroups. S is a sublattice of K, and if S is a maximal proper subgroup, then its dual lattice $S^\perp$ contains a vector not in $K^\perp$. There is only one such (linearly independent) vector if S is maximal, for if there were two then we could remove one, obtaining a smaller $S^\perp$ and such a larger proper subgroup. Any nontrivial vector not in $K^\perp$ determines such a subgroup, so there are $|G/K^\perp| - 1$ choices (where e.g. $G = Z_N^n$), and

$$\text{Prob}(m \text{ elements generate } S) \geq 1 - 2^{-m}|G/K^\perp|$$

So in the case of the hidden subgroup problem where $H \leq G = Z_N^n$, we are sampling $K = H^\perp$ and $|G/K^\perp|$ becomes $|G/H|$, the number of cosets. To have constant success probability, then we choose m such that e.g.

$$2^{-m}|G/H| < \frac{1}{2}$$

or $m - 1 > \log|G/H|$ (compare the conclusion for Simon's problem)

and $|G/H| < N^n$, so it suffices if $m = O(n \log N)$

How large should N be? For period finding with $r \leq R$, choosing $N \geq R^2$ provided adequate precision for finding r. For an integral lattice with generator matrix M, its inverse matrix (transpose of M^{-1}) has entries

$$P(\text{success}) \geq \left(1 - \frac{1}{N\delta}\right)^n$$

and the prob of being successful in each of m consecutive samplings is

$$P(\text{success m Time}) \geq \left(1 - \frac{1}{N\delta}\right)^{nm}$$

For $\delta = 1/R^2$, the success probability is a constant for

$$\frac{mnR^2}{N} < \text{constant} \quad \text{or} \quad N = O(mnR^2)$$

Since $m = O(n \log N)$ samples are sufficient to find generators of $H^\perp$, we conclude it suffices to choose N to be

$$N = O(n^2R^2 \log N) = O(n^2R^2 \log(nR))$$

This is good enough to determine $H^\perp$ in $m = O(n \log N) = O(n \log(nR))$ queries, and

the generators of H are found by inverting the matrix $M^\perp$ that generates $H^\perp$.

The algorithm is efficient: both the number of queries and the number of steps in the quantum Fourier Transform are poly log in the (upper bound on) the number of cosets $|G/H|$