The hidden subgroup problem

Isaac Holt

Motivation

Goals in quantum algorithms:

- Find classical problems that can be solved exponentially faster with a quantum computer.
- Understand what makes these problems amenable to such a speedup.
- Understand the key ingredient unique to quantum computing that gives such a speedup.

Motivation

Goals in quantum algorithms:

- Find classical problems that can be solved exponentially faster with a quantum computer.
- Understand what makes these problems amenable to such a speedup.
- Understand the key ingredient unique to quantum computing that gives such a speedup.

The *abelian hidden subgroup problem* unifies almost all "useful" problems which have a (possible) quantum exponential speedup.

Two simple problems

Two simple problems

Bernstein-Vazirani problem: given (an oracle for) a function $F : \mathbb{F}_2^n \to \mathbb{F}_2, F(x) = x.s$ for some unknown $s \in \mathbb{F}_2^n$, find s.

Two simple problems

Bernstein-Vazirani problem: given (an oracle for) a function $F : \mathbb{F}_2^n \to \mathbb{F}_2, F(\boldsymbol{x}) = \boldsymbol{x}.\boldsymbol{s}$ for some unknown $\boldsymbol{s} \in \mathbb{F}_2^n$, find $\boldsymbol{s}$.

Simon's problem: given (an oracle for) a function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$, $f(\boldsymbol{x}) = f(\boldsymbol{y}) \iff \boldsymbol{y} = \boldsymbol{x} + \boldsymbol{a}$ for some unknown $\boldsymbol{a} \in \mathbb{F}_2^n$, find $\boldsymbol{a}$.

Bernstein-Vazirani problem: given (an oracle for) a function $F : \mathbb{F}_2^n \to \mathbb{F}_2, F(\boldsymbol{x}) = \boldsymbol{x}.\boldsymbol{s}$ for some unknown $\boldsymbol{s} \in \mathbb{F}_2^n$, find $\boldsymbol{s}$.

Simon's problem: given (an oracle for) a function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$, $f(\boldsymbol{x}) = f(\boldsymbol{y}) \iff \boldsymbol{y} = \boldsymbol{x} + \boldsymbol{a}$ for some unknown $\boldsymbol{a} \in \mathbb{F}_2^n$, find $\boldsymbol{a}$.

Bernstein-Vazirani algorithm (1992) gives quantum speedup from n *queries* of f to 1.

Two simple problems

Bernstein-Vazirani problem: given (an oracle for) a function $F : \mathbb{F}_2^n \to \mathbb{F}_2, F(\boldsymbol{x}) = \boldsymbol{x}.\boldsymbol{s}$ for some unknown $\boldsymbol{s} \in \mathbb{F}_2^n$, find $\boldsymbol{s}$.

Simon's problem: given (an oracle for) a function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$, $f(\boldsymbol{x}) = f(\boldsymbol{y}) \iff \boldsymbol{y} = \boldsymbol{x} + \boldsymbol{a}$ for some unknown $\boldsymbol{a} \in \mathbb{F}_2^n$, find $\boldsymbol{a}$.

Bernstein-Vazirani algorithm (1992) gives quantum speedup from n *queries* of f to 1.

Simon's algorithm (1993) gives quantum speedup from $2^{\Theta(n/2)}$ to $\Theta(n)$ (note this doesn't separate **BQP** and **BPP** since the input is an oracle so can't be encoded as a bit string).

Two more problems

Period finding problem: given (an oracle for) a function $f : \mathbb{Z}_N \to X$, with $f(y) = f(x) \iff y = x + a$ for some fixed $a \in \mathbb{Z}_N$, find a.

Two more problems

Period finding problem: given (an oracle for) a function $f : \mathbb{Z}_N \to X$, with $f(y) = f(x) \iff y = x + a$ for some fixed $a \in \mathbb{Z}_N$, find a.

Discrete logarithm problem: given finite cyclic group G' of size M, a generator g of G', and an element $x \in G'$, find $\ell \in \mathbb{Z}_M$ such that $g^\ell = x$. Equivalently, find invertible $a, b \in \mathbb{Z}_M$ such that $g^a x^{-b} = e_{G'}$ (then $\ell = ab^{-1}$).

Two more problems

Period finding problem: given (an oracle for) a function $f : \mathbb{Z}_N \to X$, with $f(y) = f(x) \iff y = x + a$ for some fixed $a \in \mathbb{Z}_N$, find a.

Discrete logarithm problem: given finite cyclic group G' of size M, a generator g of G', and an element $x \in G'$, find $\ell \in \mathbb{Z}_M$ such that $g^\ell = x$. Equivalently, find invertible $a, b \in \mathbb{Z}_M$ such that $g^a x^{-b} = e_{G'}$ (then $\ell = ab^{-1}$).

Both problems are in BQP (Shor 1994), and no *known* polynomial time classical algorithm exists for them.

(Side note: Shor's algorithm reduces to period finding (although over $\mathbb{Z}$ rather than $\mathbb{Z}_N$).)

Unifying the problems

The four problems above all involve some global structure, so natural to consider groups. Additionally, the input function is fixed on a subset of inputs, so natural to consider subgroups.

In Simon's problem and period finding, there is (strict) periodicity: the function agrees on inputs if and only if the inputs lie in the same period.

We can generally periodicity in $\mathbb{F}_2^n$ and in $\mathbb{Z}_N$ to general subgroup periodicity.

Subgroup periodicity

Let G be a group, H be a subgroup of G, X be a finite set, $f: G \to X$ be a function. If any of the following holds, f is **H-periodic**:

- f is constant on (left) cosets of H: $f(gh) = f(g)$ for all $g \in G, h \in H$.
- The function $\overline{f}: G/H \to X, \overline{f}(gH) = f(g)$, is well-defined.

Subgroup periodicity

Let G be a group, H be a subgroup of G, X be a finite set, $f : G \to X$ be a function. If any of the following holds, f is **H-periodic**:

- f is constant on (left) cosets of H: $f(gh) = f(g)$ for all $g \in G, h \in H$.
- The function $\overline{f} : G/H \to X, \overline{f}(gH) = f(g)$, is well-defined.

Additionally, f is **strictly H-periodic** if any of the following holds:

- f takes distinct values on distinct (left) cosets of H.
- $\overline{f}$ is injective.
- H is the largest (unique) subgroup K such that f is K-periodic.

Easy exercise: check the equivalences.

Input $f : G \to X, G$ a group, X a finite set.

Promise f is strictly H-periodic for some unknown subgroup H of G.

Output (a set of generators of) H.

Our first four problems are all abelian HSPs:

Examples

Our first four problems are all abelian HSPs:

Simon's problem: $G = \mathbb{F}_2^n, H = \langle \boldsymbol{a} \rangle = \{\mathbf{0}, \boldsymbol{a}\}$.

Examples

Our first four problems are all abelian HSPs:

Simon's problem: $G = \mathbb{F}_2^n, H = \langle \boldsymbol{a} \rangle = \{\mathbf{0}, \boldsymbol{a}\}$.

Bernstein-Vazirani problem: $G = \mathbb{F}_2^n, H = \langle s \rangle^\perp = \{\boldsymbol{x} \in \mathbb{F}_2^n : \boldsymbol{x}.\boldsymbol{s} = 0\}$.

Examples

Our first four problems are all abelian HSPs:

Simon's problem: $G = \mathbb{F}_2^n, H = \langle \boldsymbol{a} \rangle = \{\mathbf{0}, \boldsymbol{a}\}$.

Bernstein-Vazirani problem: $G = \mathbb{F}_2^n, H = \langle s \rangle^\perp = \{\boldsymbol{x} \in \mathbb{F}_2^n : \boldsymbol{x}.\boldsymbol{s} = 0\}$.

Period finding problem: $G = \mathbb{Z}_N, H = \langle a \rangle = \{0, a, 2a, ..., (K-1)a\}$ where $K = N/a$.

Examples

Our first four problems are all abelian HSPs:

Simon's problem: $G = \mathbb{F}_2^n, H = \langle \boldsymbol{a} \rangle = \{\boldsymbol{0}, \boldsymbol{a}\}$.

Bernstein-Vazirani problem: $G = \mathbb{F}_2^n, H = \langle s \rangle^{\perp} = \{\boldsymbol{x} \in \mathbb{F}_2^n : \boldsymbol{x}.\boldsymbol{s} = 0\}$.

Period finding problem: $G = \mathbb{Z}_N, H = \langle a \rangle = \{0, a, 2a, ..., (K-1)a\}$ where $K = N/a$.

Discrete log problem: $G = \mathbb{Z}_n, H = \{(a, \ell a) : a \in \mathbb{Z}_n\}$.

Before the HSP was defined, Simon's problem already had a poly time algorithm: (quantumly) obtain $\Theta(n)$ vectors x such that $x.a = 0 \in \mathbb{Z}_2$ i.e. $\frac{a_1 x_1}{2} + \cdots + \frac{a_n x_n}{2} \in \mathbb{Z}$, then find a w.h.p. by Gaussian elimination.

Before the HSP was defined, Simon's problem already had a poly time algorithm: (quantumly) obtain $\Theta(n)$ vectors x such that $x.a = 0 \in \mathbb{Z}_2$ i.e. $\frac{a_1 x_1}{2} + \cdots + \frac{a_n x_n}{2} \in \mathbb{Z}$, then find a w.h.p. by Gaussian elimination.

Classification of finite abelian groups tells us that $G \cong \mathbb{Z}_{N_1} \times \cdots \times \mathbb{Z}_{N_k}$ for some $k, N_1, \ldots, N_k \in \mathbb{N}$.

Natural generalisation to finite abelian groups: find vectors x such that $\frac{h_1 x_1}{N_1} + \cdots + \frac{h_k x_k}{N_k} \in \mathbb{Z}$ for all $h \in H$, i.e.

$$h_1 x_1 N_{(1)} + \cdots + h_k x_k N_{(k)} = 0 \in \mathbb{Z}_{N_1 \cdots N_k},$$

where $N_{(i)} = \prod_{j \neq i} N_j$. Then again can solve by Gaussian elimination.

If we had $|H\rangle$, then could just make measurements to obtain elements of H. Recall $\mathrm{Graph}(f) = \{(g, f(g)) : g \in G\}$. We can prepare

$$|\mathrm{Graph}(f)\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle |f(g)\rangle \in \mathbb{C}^{|G|} \otimes \mathbb{C}^{|X|}$$

efficiently (by preparing $|G\rangle|0\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle|0\rangle$, then applying the quantum oracle $U_f : |x\rangle|y\rangle \mapsto |x\rangle|y + f(x)\rangle$).

If we had $|H\rangle$, then could just make measurements to obtain elements of H. Recall $\mathrm{Graph}(f) = \{(g, f(g)) : g \in G\}$. We can prepare

$$|\mathrm{Graph}(f)\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle |f(g)\rangle \in \mathbb{C}^{|G|} \otimes \mathbb{C}^{|X|}$$

efficiently (by preparing $|G\rangle|0\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle|0\rangle$, then applying the quantum oracle $U_f : |x\rangle|y\rangle \mapsto |x\rangle|y + f(x)\rangle$).

Strict H-periodicity of f means that measuring then discarding the second register produces a **coset state** $|c + H\rangle = \frac{1}{\sqrt{|H|}} \sum_{h \in H} |c + h\rangle$, for a *uniformly random* and *unknown* shift $c \in G$.

We would like $|H\rangle$, but shift c is *unknown* so can't obtain from $|c+H\rangle$.

Idea: $|c+H\rangle = U_c|H\rangle$, where U_c is the unitary mapping $|g\rangle \mapsto |c+g\rangle$.

Want to find a basis $\{|\chi_g\rangle\}$ that is invariant under the action of U_c for all c (up to a global phase), i.e. $U_c|\chi_g\rangle = e^{i\theta_g}|\chi_g\rangle$ for some θ_g.

Then performing basis change from this "shift-invariant" $\{|\chi_g\rangle\}$ basis to the computational basis $\{|g\rangle\}$ will map $|c+H\rangle$ and $|H\rangle$ to states with the amplitudes (up to phases), which means measuring in the computational basis yields the same output probability distribution.

We would like $|H\rangle$, but shift c is *unknown* so can't obtain from $|c+H\rangle$.

Idea: $|c+H\rangle = U_c|H\rangle$, where U_c is the unitary mapping $|g\rangle \mapsto |c+g\rangle$.

Want to find a basis $\{|\chi_g\rangle\}$ that is invariant under the action of U_c for all c (up to a global phase), i.e. $U_c|\chi_g\rangle = e^{i\theta_g}|\chi_g\rangle$ for some θ_g.

Then performing basis change from this "shift-invariant" $\{|\chi_g\rangle\}$ basis to the computational basis $\{|g\rangle\}$ will map $|c+H\rangle$ and $|H\rangle$ to states with the amplitudes (up to phases), which means measuring in the computational basis yields the same output probability distribution.

Exercise: show that $|\chi_g\rangle = \frac{1}{\sqrt{|G|}}\sum_{x\in G}\overline{\chi_g(x)}|x\rangle$, $g \in G$, form a shift-invariant orthonormal basis (here, $\chi_g(x) = e^{2\pi i\left(\frac{g_1 x_1}{N_1}+\cdots+\frac{g_k x_k}{N_k}\right)}$).

The QFT is the unitary which maps the *shift-invariant basis* $\{|\chi_g\rangle : g \in G\}$ to the *computational basis* $\{|g\rangle : g \in G\}$.

Exercise: show that $\mathrm{QFT}|x\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} \chi_g(x) |g\rangle$.

The QFT is the unitary which maps the *shift-invariant basis* $\{|\chi_g\rangle : g \in G\}$ to the *computational basis* $\{|g\rangle : g \in G\}$.

Exercise: show that $\mathrm{QFT}|x\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} \chi_g(x)|g\rangle$.

Can implement QFT over $\mathbb{Z}_N$ exactly, with N a power of two, using $O(\log^2 N)$ gates. Harder exercise: write out a quantum circuit for this (hint: use controlled phase gates of the form $\begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^\ell} \end{bmatrix}$).

The QFT is the unitary which maps the *shift-invariant basis* $\{|\chi_g\rangle : g \in G\}$ to the *computational basis* $\{|g\rangle : g \in G\}$.

Exercise: show that $\mathrm{QFT}|x\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} \chi_g(x)|g\rangle$.

Can implement QFT over $\mathbb{Z}_N$ exactly, with N a power of two, using $O(\log^2 N)$ gates. Harder exercise: write out a quantum circuit for this (hint: use controlled phase gates of the form $\begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^\ell} \end{bmatrix}$).

Can also implement QFT over $\mathbb{Z}_N$ exactly using $O(\log^2 N)$ gates.

The QFT is the unitary which maps the *shift-invariant basis* $\{|\chi_g\rangle : g \in G\}$ to the *computational basis* $\{|g\rangle : g \in G\}$.

Exercise: show that $\mathrm{QFT}|x\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} \chi_g(x)|g\rangle$.

Can implement QFT over $\mathbb{Z}_N$ exactly, with N a power of two, using $O(\log^2 N)$ gates. Harder exercise: write out a quantum circuit for this (hint: use controlled phase gates of the form $\begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^\ell} \end{bmatrix}$).

Can also implement QFT over $\mathbb{Z}_N$ exactly using $O(\log^2 N)$ gates.

Exercise: show the QFT over $G \times G'$ is $\mathrm{QFT}_G \otimes \mathrm{QFT}_{G'}$.

The QFT is the unitary which maps the *shift-invariant basis* $\{|\chi_g\rangle : g \in G\}$ to the *computational basis* $\{|g\rangle : g \in G\}$.

Exercise: show that $\mathrm{QFT}|x\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} \chi_g(x)|g\rangle$.

Can implement QFT over $\mathbb{Z}_N$ exactly, with N a power of two, using $O(\log^2 N)$ gates. Harder exercise: write out a quantum circuit for this (hint: use controlled phase gates of the form $\begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^\ell} \end{bmatrix}$).

Can also implement QFT over $\mathbb{Z}_N$ exactly using $O(\log^2 N)$ gates.

Exercise: show the QFT over $G \times G'$ is $\mathrm{QFT}_G \otimes \mathrm{QFT}_{G'}$.

So QFT over all abelian groups implementable exactly and efficiently.

Abelian HSP algorithm

1. Prepare a random coset state $|c + H\rangle$ via coset sampling.
2. Apply the quantum Fourier transform to $|c + H\rangle$.
3. Measure in the computational basis (this is Fourier sampling).

1. Prepare a random coset state $|c+H\rangle$ via coset sampling.
2. Apply the quantum Fourier transform to $|c+H\rangle$.
3. Measure in the computational basis (this is Fourier sampling).

Output distribution is same for $|c+H\rangle$ as it is for $|H\rangle$, so now can assume WLOG that $c=0$.

Exercise: check that the measurement yields $(x_1,\ldots,x_k)$ such that $\frac{h_1x_1}{N_1}+\cdots+\frac{h_kx_k}{N_k}\in\mathbb{Z}$ for all $h\in H$.

As for Simon's algorithm, obtaining $\Theta(\log|G|)$ such samples x is sufficient to determine H.

What about non-abelian groups? Where does our algorithm for the abelian HSP break down for non-abelian groups?

What about non-abelian groups? Where does our algorithm for the abelian HSP break down for non-abelian groups?

Can still perform coset sampling to obtain random $|cH\rangle$.

What about non-abelian groups? Where does our algorithm for the abelian HSP break down for non-abelian groups?

Can still perform coset sampling to obtain random $|cH\rangle$.

There is notion of Fourier transform for non-abelian groups, but there is no shift-invariant basis (due to non-commutativity).

What about non-abelian groups? Where does our algorithm for the abelian HSP break down for non-abelian groups?

Can still perform coset sampling to obtain random $|cH\rangle$.

There is notion of Fourier transform for non-abelian groups, but there is no shift-invariant basis (due to non-commutativity).

It suspected that the HSP for non-abelian groups is hard, even on quantum computers.

Closely connected to group representation theory. Representations of non-abelian groups are often much more complicated (e.g. hard to write down irreducible representations of the symmetric group explicitly).

Although the non-abelian HSP is thought to be hard, it can be solved using polynomially many queries to the input function f.

Although the non-abelian HSP is thought to be hard, it can be solved using polynomially many queries to the input function f.

Idea: prepare M copies of the state $|\mathrm{Graph}(f)\rangle$, along with a counter register and an indicator register. The indicator register is measured at the end, this reveals what the hidden subgroup H is.

Although the non-abelian HSP is thought to be hard, it can be solved using polynomially many queries to the input function f.

Idea: prepare M copies of the state $|\mathrm{Graph}(f)\rangle$, along with a counter register and an indicator register. The indicator register is measured at the end, this reveals what the hidden subgroup H is.

For each subgroup, apply a "check" unitary to the full state, which updates the indicator register if that subgroup is the hidden subgroup. If not, then the state only changes very slightly (L^2 distance to the previous state is exponentially small in M).

Since number of subgroups is at most $2^{\log^2|G|}$, M can be $O(\log^2|G|)$.

Dihedral case

$$D_N = \langle r, s \mid r^N = s^2 = e, rs = sr^{-1} \rangle.$$

Dihedral groups are some of the simplest non-abelian groups.

Exercise: DHSP reduces polynomially to the case that the hidden subgroup is a reflection (order two).

Dihedral case

$$D_N = \langle r, s \mid r^N = s^2 = e, rs = sr^{-1} \rangle.$$

Dihedral groups are some of the simplest non-abelian groups.

Exercise: DHSP reduces polynomially to the case that the hidden subgroup is a reflection (order two).

Best known result: Kuperberg's sub-exponential time algorithm runs in time $2^{O(\sqrt{\log N})}$.

Dihedral case

$D_N = \langle r, s \mid r^N = s^2 = e, rs = sr^{-1} \rangle.$

Dihedral groups are some of the simplest non-abelian groups.

Exercise: DHSP reduces polynomially to the case that the hidden subgroup is a reflection (order two).

Best known result: Kuperberg's sub-exponential time algorithm runs in time $2^{O(\sqrt{\log N})}$.

Still believed to be hard for quantum computers: closely connected to problems on lattices (e.g. shortest vector problem). So also connected to cryptography (lattice-based cryptography is main candidate for post-quantum cryptography).

Other cases

If hidden subgroup is known to be normal, then there is a poly time quantum algorithm (similar to the abelian HSP algorithm, but involves group representation theory, more difficult to prove correctness).

If hidden subgroup is known to be normal, then there is a poly time quantum algorithm (similar to the abelian HSP algorithm, but involves group representation theory, more difficult to prove correctness).

Graph isomorphism reduces polynomially to the hidden subgroup problem on the symmetric group, where the subgroup is either of size 2 or is trivial. The symmetric group is much more complicated than dihedral group, and its representations are much less simple to describe.

Even distinguishing a size 2 hidden subgroup from the trivial subgroup of S_n is thought to be hard.