Introduction to quantum computing

Anthony Leverrier, Inria Paris
&
Mazyar Mirrahimi, Inria Paris

IQUPS course 2018

Outline of the course

courses 1 – 2: basics of quantum computing and standard algorithms (Anthony Leverrier)

- May 29 (9:15 - 10:45): basics of quantum computing: qubits, measurements, circuit model, query complexity model, Simon's algorithm
- *June 5 (11:00 – 12:30): quantum Fourier transform, Shor's algorithm, Grover's algorithm*

courses 3 – 4: quantum error correction and quantum fault tolerance (Mazyar Mirrahimi)

- June 18: basics of quantum error correction (discretization of errors, Shor an Steane codes) and fault-tolerance
- June 25: towards experimental implementation: surface codes and continuous-variable codes

Last week

- several equivalent models for quantum computing: circuit, adiabatic, measurement-based ...
- 2 models of quantum complexity
  - standard model: input is a classical string, quantum circuit and measurement in the computational basis, *what is the number of gates?*
  - query complexity model: input given as a black box (ex: function), *how many queries are made to the black box?*
- Simon's algorithm: exponential speedup compared to classical randomized algorithms in the quantum query complexity model

Outline of the course

- Simon's algorithm
- quantum Fourier transform: exponential speedup, if input and output encoded in a quantum state
- Shor's algorithm for factoring
- Grover's search algorithm

Simon's algorithm

Exponential speedup for query complexity (we count queries, not ordinary operations)

hidden period for 2-to-1 function

Input: $f : \{0,1\}^n \to \{0,1\}^n$ with the property that $\exists s \neq 0 \in \{0,1\}^n$ such that

$$f(x) = f(y) \iff (x = y \quad \text{or} \quad x = y \oplus s).$$

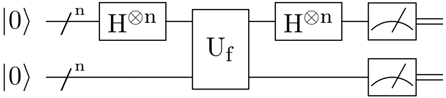
Find s.

complexity

- randomized classical algorithm in $O(\sqrt{2^n})$ queries with birthday paradox
- this is essentially optimal for classical algorithms
- quantum (Simon's algorithm): $O(n)$ queries

$\implies$ exponential separation *quantum* vs *randomized classical*

Simon's algorithm



$$|0^n\rangle|0^n\rangle \longrightarrow \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle|0^n\rangle \longrightarrow \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle|f(x)\rangle$$

Measure 2nd n-bit register: yields $f(x) \in \{0,1\}^n$, collapses the first register to superposition of 2 indices compatible with $f(x)$

$$\frac{1}{\sqrt{2}}(|x\rangle + |x \oplus s\rangle)|f(x)\rangle$$

Hadamard to first n qubits:

$$\frac{1}{\sqrt{2^{n+1}}}\left(\sum_{j\in\{0,1\}^n}(-1)^{x\cdot j}|j\rangle + \sum_{j\in\{0,1\}^n}(-1)^{(x\oplus s)\cdot j}|j\rangle\right) = \frac{1}{\sqrt{2^{n+1}}}\sum_{j\in\{0,1\}^n}(-1)^{x\cdot j}(1+(-1)^{s\cdot j})|j\rangle$$

Simon's algorithm

Measure state

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{j \in \{0,1\}^n} (-1)^{x \cdot j} (1 + (-1)^{s \cdot j}) |j\rangle$$

- $|j\rangle$ has nonzero amplitude iff $s \cdot j = 0 \mod 2$.
- The measurement outcome is uniformly drawn from $\{j \,|\, s \cdot j = 0 \mod 2\}$.
- $\implies$ linear equation giving information about s
- repeat until we get $n-1$ independent linear equations
- solutions are 0 and s via Gaussian elimination (classical circuit of size $O(n^3)$)

$\implies$ exponential speedup in the query complexity model! Can we get it in the standard model as well?

Quantum Fourier Transform

Classical discrete Fourier transform

For N, define $\omega_N = e^{2\pi i/N}$ the N-th root of identity, and the $N \times N$ matrix:

$$F_N = \frac{1}{\sqrt{N}} \begin{pmatrix} & \vdots & \\ \cdots & \omega_N^{jk} & \cdots \\ & \vdots & \end{pmatrix}$$

We'll be mostly interested in the case $N = 2^n$.

For $v \in \mathbb{R}^N$, the Fourier transform of v is

$$\hat{v} = F_N v$$

$$\text{for} \quad j \in \{0, N-1\}, \quad \hat{v}_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{jk} v_k$$

Complexity of discrete Fourier transform

Naïve classical algorithm

matrix multiplication: $O(N)$ additions/multiplications per entry

$$\implies O(N^2) \quad \text{steps}$$

Fast Fourier Transform

Recursive procedure: compute 2 FT for $N/2$ and combine

$$\implies O(N \log N) \quad \text{steps}$$

Quantum Fourier Transform

F_N is a unitary matrix: can be interpreted as a quantum operation on $n = \log_2 N$ qubits.
If input and output are encoded as $|v\rangle = \sum_{i=0}^{N-1} v_i |i\rangle$ and $|\hat{v}\rangle = \sum_{i=0}^{N-1} \hat{v}_i |i\rangle$

$$\implies O(\log^2 N) \quad \text{steps} \quad \implies \quad \textit{exponential speedup!}$$

Efficient quantum circuit for the n*-qubit QFT* $(N = 2^n)$

linearity: sufficient to implement QFT on basis states $|x\rangle = |x_1 x_2 \cdots x_n\rangle$ *with* $x_i \in \{0, 1\}$

QFT: $|x\rangle \mapsto F_N|x\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega_N^{jk} |j\rangle$

Insight: $F_N|x\rangle$ *is a product state!*

integer in binary notation: $x = x_1 x_2 \cdots x_n$ (x_1 = most significant bit)

$$F_N|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{N-1} e^{2\pi i jx/2^n} |j\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{N-1} e^{2\pi i (\sum_{\ell=1}^n j_\ell 2^{-\ell})x} |j_1 \cdots j_n\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{j=0}^{N-1} \prod_{\ell=1}^{n} e^{2\pi i j_\ell x/2^\ell} |j_1 \cdots j_n\rangle$$

$$= \bigotimes_{\ell=1}^{n} \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i x/2^\ell} |1\rangle \right)$$

$\implies$ sufficient to prepare qubits of the form $\frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i [0.x_{n-\ell+1} x_{x-\ell+2} \cdots x_n]} |1\rangle \right)$

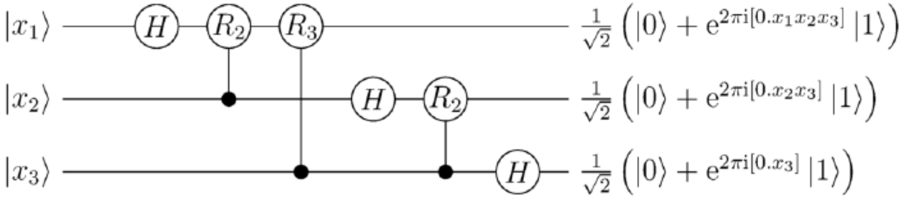
Efficient quantum circuit for the n*-qubit QFT*

Allowed gates

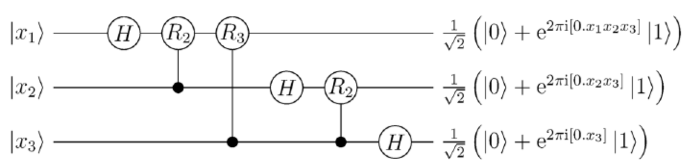
- Hadamard gate: $|0\rangle \leftrightarrow |+\rangle, \quad |1\rangle \leftrightarrow |-\rangle \qquad \mathrm{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
- phase-flip gate $\mathrm{R_s}$: $|0\rangle \mapsto |0\rangle, \quad |1\rangle \mapsto e^{2\pi i/2^s}|1\rangle \qquad \mathrm{R}_\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^s} \end{pmatrix}$

example:

$$\mathrm{F_N}|x_1x_2x_3\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_3]}|1\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_2x_3]}|1\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_1x_2x_3]}|1\rangle\right)$$



Efficient quantum circuit for the n-qubit QFT



Complexity

- n qubits
- at most n gates applied to each qubit
- total number of gates $\leq n^2 = (\log_2 N)^2$
- the phase gates are almost equal to the identity for $s \gg \log n$, so the corresponding gates can be omitted without causing much error
- complexity $\approx n \log n$

Note that the inverse Fourier transform is obtained by reversing the circuit and taking R_{-s} instead of R_s

Shor's algorithm

Factoring

Given a composite number N, find a factor of N.

- Best (known) classical algorithm: complexity $2^{(\log N)^{1/3}}$
- Shor's algorithm: complexity $(\log N)^2$ steps

Reduction to period finding

efficient algorithm for period finding $\implies$ efficient algorithm for factoring
choose random integer $x \in \{2, \cdots, N-1\}$ coprime to N and define

$$f(a) = x^a \mod N$$

$$f(0) = 1 \mod N, \quad f(1) = x \mod N, \quad f(2) = x^2 \mod N \cdots$$

This sequence is cyclic with period $r \implies$ find r!

Reduction to period finding

$$f(a) = x^a \mod N$$

Lemma

With probability $\geq 1/2$, the period r is even and $x^{r/2}+1$ and $x^{r/2}-1$ are not multiples of N.

Then,

$$\begin{aligned} x^r \equiv 1 \mod N &\iff (x^{r/2})^2 \equiv 1 \mod N \\ &\iff (x^{r/2}+1)(x^{r/2}-1) \equiv 0 \mod N \\ &\iff (x^{r/2}+1)(x^{r/2}-1) = kN \quad \text{for some} \quad k > 0 \end{aligned}$$

Then $x^{r/2}+1$ or $x^{r/2}-1$ shares a factor with N.

With Euclid algorithm, one can recover $\gcd(x^{r/2} \pm 1, N)$ efficiently, which gives non-trivial factors of N.

f can be computed efficiently

$$f(a) = x^a \mod N$$

idea: repeated squaring

- compute $x^2 \mod N, x^4 \mod N, x^8 \mod N, \ldots$
- write a in binary: $a = \sum_{i \geq 0} a_i 2^i$
- $x^a = \prod_{i\,:\,a_i=1} x^{2^i}$

Complexity

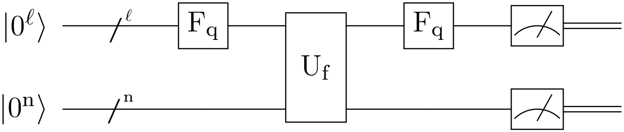
$O((\log N)^2 \log \log N \log \log \log N)$ steps

$\implies$ a quantum circuit for $U_f : |a\rangle|0^n\rangle \mapsto |a\rangle|f(a)\rangle$ has the same complexity

$\implies$ we don't need to work in the oracle model since we can implement the function quantumly

Quantum circuit for factoring

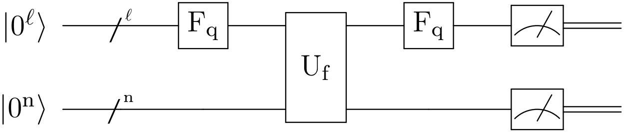
same circuit as Simon's algorithm, with Hadamard $\leftrightarrow$ QFT



- $q = 2^\ell$ such that $N^2 < q \leq 2N^2$
- Quantum Fourier Transform F_q requires $O(\log^2 N)$ gates
- black-box $U_f : |a\rangle|0^n\rangle \mapsto |a\rangle|f(a)\rangle$
  requires $O((\log N)^2 \log\log N \log\log\log N)$ steps

  $\Longrightarrow$ *this is the costly part of the algorithm!*
- $n = \lceil \log N \rceil$ qubits

Quantum circuit for factoring



$$|0^\ell\rangle|0^n\rangle \to \frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a\rangle|0^n\rangle$$
$$\to \frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a\rangle|f(a)\rangle$$

Measure second register and get $f(s)$ for $s < r$

$\implies$ first register collapses to

$$|s\rangle + |r+s\rangle + |2r+s\rangle + |3r+s\rangle + \cdots + |(m-1)r+s\rangle$$

with $m \approx q/r$

Quantum circuit for factoring

QFT applied to $\frac{1}{\sqrt{m}}\sum_{j=0}^{m-1}|jr+s\rangle$ yields

$$\frac{1}{\sqrt{m}}\sum_{j=0}^{m-1}\frac{1}{\sqrt{q}}\sum_{b=0}^{q-1}e^{2\pi i(jr+s)b/q}|b\rangle = \frac{1}{\sqrt{mq}}\sum_{b=0}^{q-1}e^{2\pi isb/q}\left(\sum_{j=0}^{m-1}e^{2\pi ijrb/q}\right)|b\rangle$$

what are the b with large amplitude?

$$\sum_{j=0}^{m-1}e^{2\pi ijrb/q} = \begin{cases} m & \text{if } e^{2\pi i\frac{rb}{q}} = 1 \\ \frac{1-e^{2\pi i\frac{mrb}{q}}}{1-e^{2\pi i\frac{rb}{q}}} & \text{if } e^{2\pi i\frac{rb}{q}} \neq 1 \end{cases}$$

- yields with high probability a value b such that rb/q is close to an integer c
- One can find efficiently (with continued fractions) the value of $\frac{c}{r}$
- c and r will be coprime with probability $\Omega(1/\log\log r)$, which will occur after $O(\log\log N)$ repetitions of the procedure
- in that case, one obtain r as the denominator by writing c/r in lowest terms.

Grover's algorithm

The search problem

The problem

Input: function $f : \{0,1\}^n \rightarrow \{0,1\}$. Find x such that $f(x) = 1$ or output no solution if no such x.

Complexity

- randomized classical algorithm: $\Theta(2^n)$ queries if single correct value
- Grover's algorithm: $O(\sqrt{2^n})$ queries and $O(n\sqrt{2^n})$ other gates

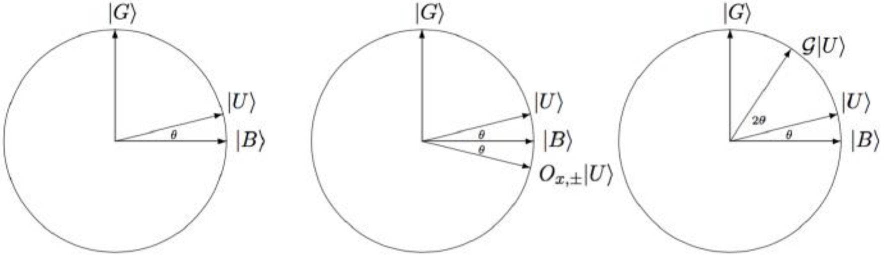
$\Longrightarrow$ quadratic speedup

Idea of the algorithm

Start with uniform superposition (via Hadamard):

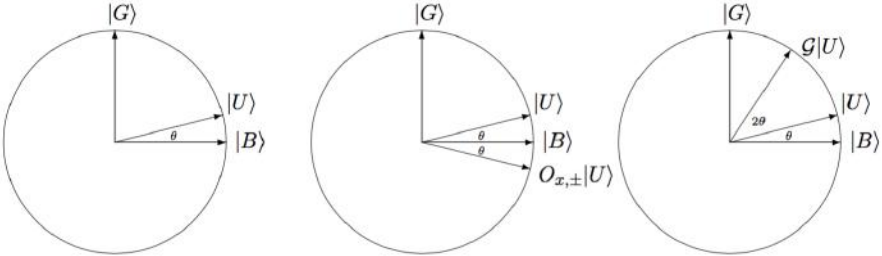
$$|U\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle = \sin\theta|G\rangle + \cos\theta|B\rangle$$

- $\sin\theta = \sqrt{t/2^n}$ and $t = \#\{x \mid f(x) = 1\}$
- good state $|G\rangle = \frac{1}{\sqrt{t}} \sum_{x \text{ s.t. } f(x)=1} |x\rangle$
- bad state $|B\rangle = \frac{1}{\sqrt{2^n - t}} \sum_{x \text{ s.t. } f(x)=0} |x\rangle$



goal: rotate in the $\{|B\rangle, |G\rangle\}$ plane to reach $|G\rangle$

How to implement rotation



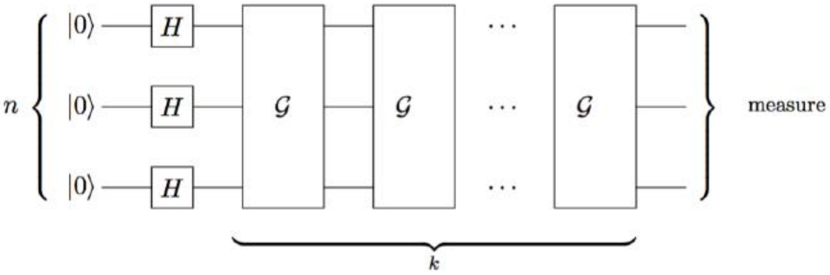
perform two reflections:

- through $|\mathrm{B}\rangle$ by calling the oracle $\mathrm{O}_{\mathrm{f},\pm} : |\mathrm{x}\rangle \mapsto (-1)^{\mathrm{f}(\mathrm{x})}|\mathrm{x}\rangle$
- through $|\mathrm{U}\rangle$ by $\mathrm{H}^{\otimes \mathrm{n}}\mathrm{R}\mathrm{H}^{\otimes \mathrm{n}} = 2|\mathrm{U}\rangle\langle \mathrm{U}| - \mathbb{1}$, where $\mathrm{R} : |\mathrm{x}\rangle \to (-1)^{[\mathrm{x} \neq 0^{\mathrm{n}}]}|\mathrm{x}\rangle$

define $\mathcal{G} = \mathrm{H}^{\otimes \mathrm{n}}\mathrm{R}\mathrm{H}^{\otimes \mathrm{n}}\mathrm{O}_{\mathrm{f},\pm} \implies$ *rotation of angle 2θ*

Grover's algorithm

assuming we know the fraction of solutions $t/2^n = \sin^2\theta \approx \theta^2$



1. start with $|U\rangle = H^{\otimes n}|0\rangle$
2. repeat $k \approx \frac{\pi/2}{2\theta} = O(1/\sqrt{t/2^n})$ times the rotation $\mathcal{G}$ of angle 2θ
3. measure and check that the outcome is a solution

Recap

- quantum Fourier transform: exponential speedup compared to classical: $\log^2 N$ vs $N \log N$
- seems like cheating because input and output are encoded in quantum states, and not classically accessible
- yet, this is the main ingredient for Shor's algorithm
- more recently (2009): HHL algorithm solves linear equations $Ax = b$ in $O(\log n)$ time (exponential speedup) if solution encoded as $|x\rangle \propto \sum_i x_i |i\rangle$
- seems again like cheating, but useful for *quantum machine learning algorithms*
- to be continued ...

next talks

Mazyar Mirrahimi on the challenges to build a quantum computer (error correction and fault-tolerance)