Quantum algorithms II: Shor

Quantum computing

G. Chênevert

Feb. 3, 2023

Quantum algorithms II: Shor

Quantum circuits

In the end, a quantum circuit is just a way to describe a big unitary matrix.

n qubits: $2^n \times 2^n$ unitary matrix

Things we can implement using unitary matrices:

- reflections
- rotations
- ...
- **Fourier transforms**

Recall: Discrete Fourier Transform

N-point Fourier transform of a sequence $x[0], \ldots, x[N-1]$:

$$y[k] = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{-\frac{2\pi ijk}{N}} x[j]$$

Matrix formulation:

$$\mathbf{y} = \mathcal{F}\,\mathbf{x} \qquad \text{with} \qquad \mathcal{F} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \zeta & \zeta^2 & \ldots & \zeta^{N-1} \\ \vdots & \vdots & & & \vdots \\ 1 & \zeta^{N-1} & \zeta^{2(N-1)} & \ldots & \zeta^{(N-1)(N-1)} \end{bmatrix}$$

where $\zeta = e^{-\frac{2\pi i}{N}}$ is *a* primitive N-th root of unity

Inverse Fourier Transform

If

$$y[k] = \frac{1}{\sqrt{N}} \sum_{j<N} \zeta^{jk} x[j]$$

then

$$x[j] = \frac{1}{\sqrt{N}} \sum_{k<N} \zeta^{-jk} y[k]$$

$$= \frac{1}{\sqrt{N}} \sum_{k<N} (\zeta^{*})^{jk} y[k]$$

with matrix $\mathcal{F}^{-1} = \mathcal{F}^{*} = \mathcal{F}^{\dagger}$

Fourier transforms are unitary

Quantum Fourier Transform

Suppose we have a quantum state $|\psi\rangle \in \mathcal{V}_N$:

$$|\psi\rangle = \sum_{x<N} \alpha_x |x\rangle$$

Its **Fourier transform** is the state

$$\mathcal{F}|\psi\rangle = \sum_{y<N} \beta_y |y\rangle$$

defined by

$$\beta_y = \frac{1}{\sqrt{N}} \sum_{x<N} \zeta^{xy} \alpha_x.$$

Quantum Fourier Transform

In other words: from a theoretical point of view

$$\text{QFT of a state} = \text{DFT of the probability amplitudes}$$

Often written in the equivalent form:

$$\mathcal{F}|x\rangle = \frac{1}{\sqrt{N}} \sum_{y<N} \zeta^{xy} |y\rangle$$

Naive classical algorithm: $\mathcal{O}(N^2)$ operations

Cooley-Tukey (1965): *Fast Fourier Transform* $\mathcal{O}(N \log N)$ operations

Quantum Fourier Transform

Theorem

There exists a quantum circuit with $\mathcal{O}((\log N)^2)$ gates that computes the QFT.

For $N = 2^n$, we can build such a circuit with $\mathcal{O}(n^2)$

- Hadamard gates $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
- controlled phase shifts $R_m = P(\frac{2\pi}{2^m}) = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^m}} \end{bmatrix}$
- swaps.

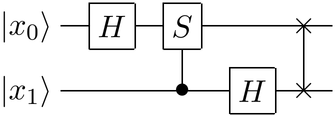
Small values of n

$n = 0$: $\mathcal{F} = I$ ✓

$n = 1$: $\mathcal{F} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = H$ ✓

$n = 2$: with $S = R_2 = P(\frac{\pi}{2}) = \sqrt{Z}$

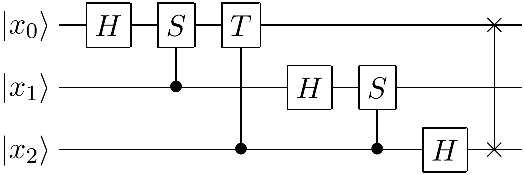
$$\mathcal{F} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix} =$$



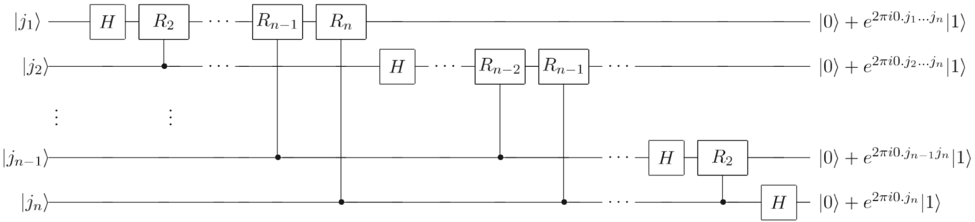
✓

Small values of n

$n = 3$: with $T = R_3 = P(\frac{\pi}{4}) = \sqrt{S}$



General QFT circuit



- n Hadamard gates
- $1 + 2 + \cdots + (n-1) = \binom{n}{2}$ controlled phase shifts
- $\leq \binom{n}{2}$ swaps

Quantum algorithms II: Shor

Application: Period detection

Suppose $f : \mathbb{Z}_N \to \mathbb{Z}_N$ is r-periodic:

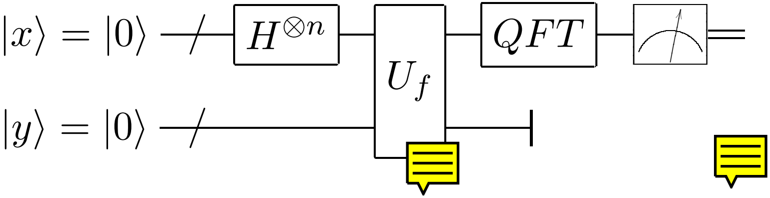
$$f(x + r) = f(x) \qquad \text{for all } x.$$

Problem: find (smallest positive such) r.

A special case of the **hidden subgroup problem**.

Idea: we can detect the period using a Fourier transform.

Quantum period detection



Theorem

If f is r-periodic, then a multiple of $\frac{N}{r}$ is measured.

Remark: for the moment we are assuming that $r \mid N$ (else replace r by $\mathrm{GCD}(r, N)$)

Proof in the $r \mid N$ case

Write $N = rs$.

Evolution of the quantum state:

$$|0\rangle \otimes |0\rangle \quad \overset{H^{\otimes n}}{\longmapsto} \quad \frac{1}{\sqrt{N}} \sum_{x<N} |x\rangle \otimes |0\rangle \quad \overset{U_f}{\longmapsto} \quad \frac{1}{\sqrt{N}} \sum_{x<N} |x\rangle \otimes |f(x)\rangle$$
$$\overset{\text{QFT}}{\longmapsto} \quad \frac{1}{N} \sum_{x<N} \sum_{y<N} \zeta^{xy} |y\rangle \otimes |f(x)\rangle$$

Now write $x = j + kr$, so that $f(x) = f(j)$.

Proof in the $r \mid N$ case

$$|\psi\rangle = \frac{1}{N}\sum_{j<r}\sum_{k<s}\sum_{y<N}\zeta^{(j+kr)y}|y\rangle\otimes|f(j)\rangle = \frac{1}{N}\sum_{y<N}\sum_{j<r}\zeta^{jy}\underbrace{\sum_{k<s}(\zeta^{ry})^k}_{0\text{ unless }\zeta^{ry}=1}|y\rangle\otimes|f(j)\rangle$$

since

$$\sum_{k<s}(\zeta^{ry})^k = \begin{cases} \dfrac{1-(\zeta^{ry})^s}{1-\zeta^{ry}} = \dfrac{1-1}{1-\zeta^{ry}} = 0 & \text{if } \zeta^{ry}\neq 1 \\ \displaystyle\sum_{k<s} 1 = s & \text{if } \zeta^{ry} = 1 \text{ i.e. } s \mid y \end{cases}$$

In the end we have

$$|\psi\rangle = \frac{1}{r}\sum_{t<r}\sum_{j<r}(\zeta^s)^{jt}\,|ts\rangle\otimes|f(j)\rangle$$

Proof in the $r \mid N$ case

$$|\psi\rangle = \frac{1}{r}\sum_{t<r}\sum_{j<r}(\zeta^s)^{jt}\,|ts\rangle \otimes |f(j)\rangle$$

Only values of y of the form ts have a non-zero probability of being measured:

$$\mathbb{P}[\,y = ts\,] = \frac{1}{r^2}\left\|\sum_{j<r}(\zeta^s)^{jt}\,|f(j)\rangle\right\|^2$$

In particular, in the special case where all values $f(j)$ are distinct:

$$\mathbb{P}[\,y = ts\,] = \frac{1}{r^2}\sum_{j<r}|(\zeta^s)^{jt}|^2 = \frac{1}{r^2}\sum_{j<r}1 = \frac{1}{r}$$

Approximate period detection

Now suppose $f : [\![0, N[\![\to [\![0, N[\![$ is **almost** r**-periodic**:

$$f(x + r) = f(x) \qquad \text{for all } 0 \leq x < N - r.$$

Theorem

If f is almost r-periodic, then an approximate multiple of $\frac{N}{r}$ is probably measured.

And then we can (probably) recover r with classical post-processing.

Analysis in the general case

Write $N = rs + a$ with $0 \leq a < r$. Everything is the same until

$$|\psi\rangle = \frac{1}{N} \sum_{y<N} \sum_{j<r} \zeta^{jy} \sum_{k<s_j} (\zeta^{ry})^k |y\rangle \otimes |f(j)\rangle$$

$$\text{with } s_j = \begin{cases} s+1 & \text{if } 0 \leq j < a \\ s & \text{if } a \leq j < r \end{cases}$$

$$\sum_{k<s_j} (\zeta^{ry})^k = \frac{1-(\zeta^{ry})^{s_j}}{1-\zeta^{ry}} = \zeta^{\frac{ry(s_j-1)}{2}} \sigma_{s_j}\left(y\frac{r}{N}\right) \quad \text{where} \quad \sigma_\alpha(x) = \begin{cases} \dfrac{\sin(\alpha\pi x)}{\sin(\pi x)} & x \notin \mathbb{Z} \\ \alpha & x \in \mathbb{Z} \end{cases}$$

Analysis in the general case

$$|\psi\rangle = \frac{1}{N} \sum_{y<N} \sum_{j<r} \zeta^{jy + \frac{ry(s_j - 1)}{2}} \sigma_{s_j}\left(y \tfrac{r}{N}\right) |y\rangle \otimes |f(j)\rangle$$

To simplify: under the assumption that all the p values $f(j)$ are distinct:

$$\mathbb{P}[\, y \,] = \frac{1}{N^2} \left(a \, \sigma_{s+1}\left(y \tfrac{r}{N}\right)^2 + (r - a) \, \sigma_s\left(y \tfrac{r}{N}\right)^2 \right)$$

cf. Sage visualization

Analysis in the general case

Proposition

The probability that $\lfloor t\frac{N}{r} \rceil$ or $\lceil t\frac{N}{r} \rceil$ is measured is asymptotically $\geq \frac{4}{\pi^2} \approx 40\%$

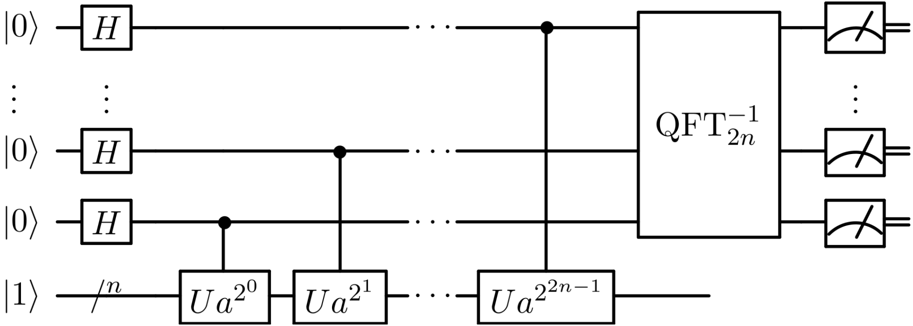
with t uniformly distributed among $[\![0, r[\![$.

Thus probability $.4\left(1 - \frac{1}{r}\right)$ of getting a "good" result y.

Fact: If $N > 2r^2$, the period r can then be efficiently recovered since $\frac{t}{r}$ appears in lowest terms in the continued fraction expansion of $\frac{y}{N}$ and if r is large, t and r will most likely be coprime (probability $\approx \frac{1}{\log \log r}$ of failure).

Quantum algorithms II: Shor

Shor's algorithm (1994)



Shor's algorithm

(Probably) factors an integer N with quantum complexity

$$\mathcal{O}\big((\log N)^2(\log\log N)(\log\log\log N)\big)$$

This is much better than the best currently known classical algorithm

that has complexity

$$\mathcal{O}\Big(e^{1.9(\log N)^{\frac{1}{3}}(\log\log N)^{\frac{2}{3}}}\Big)$$

A note on complexity

If N is an integer that can be written on n bits:

$$N < 2^n \qquad \text{so} \qquad \log_2(N) < n.$$

This is the natural parametrer to measure the size of an integer.

Factorization on a classical computer: $\mathcal{O}\left(e^{1.9 n^{\frac{1}{3}} (\log n)^{\frac{2}{3}}}\right)$

Factorization on a quantum computer: $\mathcal{O}\left(n^2 (\log n)(\log \log n)\right)$ or even $\mathcal{O}\left(n^2 (\log n)\right)$

quasiexponential speedup

Factoring vs period finding

Suppose $N = pq$ with p and q two distinct prime numbers.

Theorem (Euler)

For any integer a coprime with N, we have $a^r \equiv 1 \bmod N$ for

$$r = \varphi(N) = (p-1)(q-1).$$

In other words: r is a period for the function $f : x \mapsto a^x \bmod N$.

Reduction from factoring to period finding

To factor $N = pq$:

- pick a random integer $a \in [\![0, N[\![$
- with high probability $\mathrm{GCD}(a, N) = 1$
- if $x \mapsto a^x \bmod N$ has *odd* period, pick another a
- so now we have an integer a of even multiplicative order $2r$: $a^{2r} \equiv 1 \bmod N$

$$N \mid a^{2r} - 1 = (a^r - 1)(a^r + 1)$$

- there is a 50% chance that $\mathrm{GCD}(N, a^r \pm 1)$ are non-trivial divisors of N

Small example: $N = 21$

Try $a = 4$:

$$4 \quad \to \quad 4^2 = 16 \quad \to \quad 4^3 \equiv 1 \qquad ✗$$

Try $a = 5$:

$$5 \quad \to \quad 5^2 \equiv 4 \quad \to \quad 5^3 \equiv 20 \quad \to \quad 5^4 \equiv 16 \quad \to \quad 5^5 \equiv 17 \quad \to \quad 5^6 \equiv 1$$

but $\mathrm{GCD}(5^3 - 1, 21) = 1$, $\mathrm{GCD}(5^3 + 1, 21) = 21$ ✗

Try $a = 2$:

$$2 \quad \to \quad 2^2 = 4 \quad \to \quad 2^3 = 8 \quad \to \quad 2^4 = 16 \quad \to \quad 2^5 \equiv 11 \quad \to \quad 2^6 \equiv 1$$

and $\mathrm{GCD}(2^3 - 1, 21) = 7$, $\mathrm{GCD}(2^3 + 1, 21) = 3$ ✓

Implementation

So we need a quantum implementation U_f of the function

$$f(x) = a^x \bmod N.$$

Shor picks $Q = 2^n > N^2$ in order to be able to apply postprocessing and considers the function $f(x)$ for $x \in [\![0, Q[\![$.

If x is written in binary:

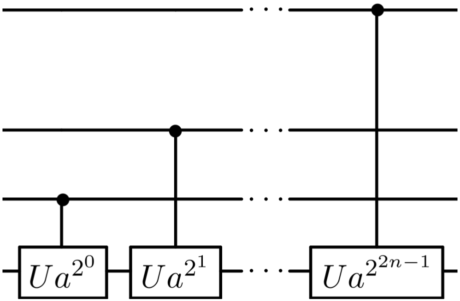
$$x = 2^{n-1} b_{n-1} + \cdots + 2^1 b_1 + 2^0 b_0$$

then

$$a^x = (a^{2^{n-1}})^{b_{n-1}} \cdots (a^{2^1})^{b_1} \cdot (a^{2^0})^{b_0}$$

Implementation

Thus we only need to implement "multiplication by a^{2^k} mod N"



This is actually the difficult part! One approach is to translate *reversible* classical arithmetical circuits into quantum circuits + classical pre-processing.

Quantum factoring: summary

To find a nontrivial factor of N:

- Pick $Q = 2^n$ large enough
- Choose a coprime with N randomly
- Implement modular exponentiation $f(x) = a^x \bmod N$ as a quantum circuit
- Apply QFT + classical post-processing to recover period R of f
- Repeat until $R = 2r$ is even and $\mathrm{GCD}(a^r - 1, N) \neq 1, N$
- Output $\mathrm{GCD}(a^r - 1, N)$

Towards Shor

Period detection

Shor's algorithm

Shor IRL

Recent history of factoring

- 1977: RSA public-key cryptosystem, based on the difficulty of factoring large quasiprime integers $N = pq$

  Scientific American Challenge: factor RSA-129 $\approx 2^{426}$

- 1981: Quadratic sieve

- 1994: RSA-129 factored (1600 computers)

- 1996: General number field sieve, RSA-130 $\approx 2^{430}$ factored

  $\vdots$

- Feb. 28, 2020: RSA-250 $\approx 2^{829}$ factored (P. Zimmerman, INRIA)

  2048-bit RSA moduli are considered out of reach for the next 25 years

Quantum computers might change that

Computing

How a quantum computer could break 2048-bit RSA encryption in 8 hours

A new study shows that quantum technology will catch up with today's encryption standards much sooner than expected. That should worry anybody who needs to store data securely for 25 years or so.

by **Emerging Technology from the arXiv** May 30, 2019

Quantum computers might change that

IBM unveils its first commercial quantum computer

Frederic Lardinois @fredericl / 5:29 pm CET • January 8, 2019

Comment

At CES, **IBM** today announced its first commercial quantum computer for use outside of the lab. The 20-qubit system combines into a single package the quantum and classical computing parts it takes to use a machine like this for research and business applications. That package, the IBM Q system, is still huge, of course, but it includes everything a company would need to get started with its quantum computing experiments, including all

How to factor 2048-bit RSA integers in 8 hours

Quantum Physics

How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits

Craig Gidney, Martin Ekerå

(Submitted on 23 May 2019 (v1), last revised 5 Dec 2019 (this version, v2))

We significantly reduce the cost of factoring integers and computing discrete logarithms in finite fields on a quantum computer by combining techniques from Shor 1994, Griffiths-Niu 1996, Zalka 2006, Fowler 2012, Ekerå-Håstad 2017, Ekerå 2017, Ekerå 2018, Gidney-Fowler 2019, Gidney 2019. We estimate the approximate cost of our construction using plausible physical assumptions for large-scale superconducting qubit platforms: a planar grid of qubits with nearest-neighbor connectivity, a characteristic physical gate error rate of 10^{-3}, a surface code cycle time of 1 microsecond, and a reaction time of 10 microseconds. We account for factors that are normally ignored such as noise, the need to make repeated attempts, and the spacetime layout of the computation. When factoring 2048 bit RSA integers, our construction's spacetime volume is a hundredfold less than comparable estimates from earlier works (Fowler et al. 2012, Gheorghiu et al. 2019). In the abstract circuit model (which ignores overheads from distillation, routing, and error correction) our construction uses $3n + 0.002n \lg n$ logical qubits, $0.3n^3 + 0.0005n^3 \lg n$ Toffolis, and $500n^2 + n^2 \lg n$ measurement depth to factor n-bit RSA integers. We quantify the cryptographic implications of our work, both for RSA and for schemes based on the DLP in finite fields.

Comments: 26 pages, 10 figures, 5 tables
Subjects: **Quantum Physics (quant-ph)**
Cite as: arXiv:1905.09749 [quant-ph]

Back of the envelope estimation

Today: ~ 433 noisy qubits

When would we have functional quantum computers with 20×10^6 qubits ?

Multiplicative factor of 46,189 $\approx 2^{15.5}$

15.5 doublings – assuming the continuing validity of Rose's law

$$\implies 15.5 \times 1.5 \approx 23 \text{ years of safety}$$

This is starting to be a bit worrisome for the security of long-term secrets.

The uncertain future of modular arithmetic-based cryptography

Large-scale quantum computers would have a definite asymptotic quantum advantage over classical algorithms. Indeed, for:

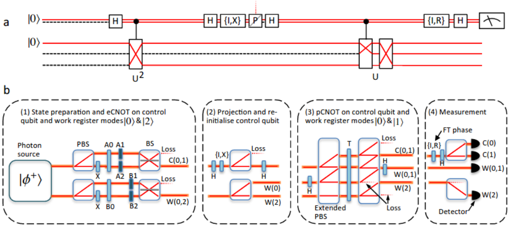
- RSA encryption and signatures (hardness of factoring)
- DSA signatures,
- elliptic curve cryptography (hardess of the *discrete logarithm problem*)

a quantum attacker breaks the system essentially as fast as classical users Alice and Bob users are able to use it with the appropriate private key ...

$\Longrightarrow$ ongoing NIST Post-quantum cryptography standardization process

Quantum factoring records

- 2001: 15 factored (IBM, Shor on 7 qubits)
- 2012: 21 factored ($a = 4$, 1 qubit + 1 qutrit)



"Compiled version of Shor's algorithm"

(see Pretending to factor large numbers on a quantum computer)

More recently: quantum annealing

- 2014: 56153 factored
- 2017: 291311 factored
- 2019: 1099551473989 factored

Numbers having a very specific shape:

$$56153_{\text{dec}} = 1101101101011001_{\text{bin}}$$

$$291311_{\text{dec}} = 1000111000111101111_{\text{bin}}$$

First, we describe the general framework for prime factorization as follows. Suppose that the integer N is the number that needs to be factored, while p and q are the prime factors, *i.e.*, $N = p \times q$. Here, the factors p and q can be denoted in binary form as $\{1p_m p_{m-1} \ldots p_2 p_1 1\}_{\text{bin}}$ for $p = 2^{m+1} + \sum_{i=1}^{m} p_i \times 2^i + 1$ and $\{1q_n q_{n-1} \ldots q_2 q_1 1\}_{\text{bin}}$ for q. In this form, the factorization problem is to find the values of $p_1, \ldots, p_m, q_1, \ldots, q_n$ that meet the restriction $N = p \times q$. Recent work has shown that the $m + n$ variables can be reduced to a significantly smaller number of variables [27]. For example, the factorization problem of $N = 291311$ reduces to the equations [27]:

$$\begin{aligned}
p_1 + q_1 &= 1 \\
p_2 + q_2 &= 1 \\
p_5 + q_5 &= 1 \\
p_1 q_2 + p_2 q_1 &= 1 \\
p_2 q_5 + p_5 q_2 &= 0 \\
p_5 q_1 + p_1 q_5 &= 1,
\end{aligned} \tag{1}$$

where the binary form of the factors are $p = \{1000p_5 01 p_2 p_1 1\}_{\text{bin}}$ and $q = \{1000q_5 01 q_2 q_1 1\}_{\text{bin}}$. Since the first three equations imply that $p_i = 1 - q_i$ for $i = 1, 2, 5$, the equations become:

$$\begin{aligned}
q_1 + q_2 - 2q_1 q_2 &= 1 \\
q_2 + q_5 - 2q_2 q_5 &= 0 \\
q_1 + q_5 - 2q_1 q_5 &= 1,
\end{aligned} \tag{2}$$

which form a 3-variable binary optimization problem.