Lecture 7: Shor's Factorisation Algorithm

Dr Iain Styles: I.B.Styles@cs.bham.ac.uk

Introduction

Last lecture, we:

- Introduced the Quantum Fourier Transform

In this lecture we will:

- Use the QFT to solve the period-finding problem
- Show how period finding leads to Shor's algorithm for number factoring

The Factoring Problem

- The RSA cryptosystem relies on our inability to find the prime factors of very large numbers
- There are no classical algorithms which can do this efficiently
- The best is the *number field sieve*, which factors a number n in $O(\exp(n^{\frac{1}{3}}\log^{\frac{2}{3}} n))$ operations
- This is exponential in n and it is impossible to factor even modest numbers, making RSA very secure
- It transpires that there is a quantum factoring algorithm due to Shor which is much better
- Shor's algorithm can factor n in $O(n^2 \log n \log\log n)$ operations
- This is exponentially faster than the number field sieve
- Factoring now becomes a tractable problem!

Period Finding

- The key idea underlying Shor's algorithm is that of period finding
- Suppose a function $f(x) = f(x+r)$, e.g. a sine wave
- $f(x)$ is said to have period r
- Classically, finding r is very hard: all we can do is test the function at many different places and find the period in this way
- A Quantum Computer can do very much better than this
- It can compute $f(x)$ for all values of x in a single parallel computation
- This makes finding the period easy
- It turns out that the factoring problem reduces to finding the period of a special function!
- So how do we find the period, r of a function?

Period Finding

- Start with two qubit registers of size n (large enough to hold some number N)
- Prepare these registers both in state $|0\rangle$.
- The composite system then has state $|\psi\rangle = |0\rangle\,|0\rangle$
- First we apply the QFT to the first register. Recall that the QFT is

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \mathrm{e}^{2\pi \mathrm{i} jk/N}$$

- Then the state after QFT is

$$|\psi\rangle = \frac{1}{\sqrt{\omega}} \sum_{j=0}^{\omega-1} |x\rangle\,|0\rangle$$

- where $\omega = 2^n$

Evaluating $f(x)$ in parallel

- Now we operate on $|\psi\rangle$ with some unitary operator which calculates $f(x)$ and puts it in the second register:

$$U_f\,|x\rangle\,|0\rangle = |x\rangle\,|f(x)\rangle$$

- The total state of the system is then

$$|\psi\rangle = \frac{1}{\sqrt{\omega}} \sum_{x=0}^{\omega-1} |x\rangle\,|f(x)\rangle$$

- We've calculated $f(x)$ for all $\omega = 2^n$ values of x in one go!
- For 100 qubits, this is 2^{100} calculations in parallel!
- But we can't see the results of all the calculations due to the way measurement works
- Nonetheless, measuring the system is useful!

Making the Measurement

- Make a measurement of the second register, giving answer $f(x) = u$
- The second register must collapse onto state $|u\rangle$
- The first register must collapse into a superposition of all values of x which give $f(x) = u$:

$$|\psi\rangle = \frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} |x_0 + kr\rangle\,|u\rangle$$

- Contents of first register are periodic: apply QFT to obtain the spectrum

$$\begin{aligned} |\psi\rangle &= \frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} \mathrm{e}^{2\pi \mathrm{i}(x_0+kr)y/2^n} |y\rangle \\ &= \frac{1}{\sqrt{2^n m}} \sum_{y=0}^{2^n-1} \mathrm{e}^{2\pi \mathrm{i} x_0 y/2^n} \sum_{k=0}^{m-1} \mathrm{e}^{2\pi \mathrm{i} kry/2^n} |y\rangle \end{aligned}$$

Making the Measurement

- It follows that

$$p(y) = \frac{1}{\sqrt{2^n m}} \left| \sum_{k=0}^{m-1} e^{2\pi \mathrm{i} kry/2^n} \right|^2$$

- Maxima when $y \approx j2^n/r$ (j an integer)
- Measurement likely to give $j2^n/r$
- Easy to deduce r as long as j and r have no common factors
- If they do, the algorithm fails and we must run it again and hope for better luck when we measure!
- So we can use a QC to find the period of a function
- A careful choice of f(x) can allow us to solve the number factoring problem
- Choose $f(x) = a^x \mod N$, where N is the number to be factored and $a < N$ is a random number, then the period r can be used to find the prime factors of N

Shor's Algorithm

The stages of Shor's Algorithm are as follows:

1. If N is even, return the factor 2
2. Determine whether $N = y^b$ for integers $y \geq 1$, $b \geq 2$. If so, return y
3. Choose $1 \leq a \leq N-1$. If $\gcd(a, N) > 1$, return $\gcd(a, N)$
4. Compute the period r of $f(x) = a^x \mod N$
5. If r is even, and $a^{r/2} \mod N \neq -1 \mod N$, then compute $\gcd(a^{r/2} \pm 1, N)$. For most a, $a^{r/2} \pm 1$ share a common factor with n. Otherwise the algorithm has failed and we must run it again.

- Note that efficient classical algorithms exist for stages 1,2, and for finding the gcd
- We are in desparate need of an example to show how this works!
- Let's try to factor $N = 39$

Factoring 39

- First, let's make sure we can't do this efficiently classically
- 39 is not even, and it cannot be written as y^b for integers y and b
- Therefore we cannot factor 39 efficiently classically
- Choose $a = 7$, so that $\gcd(a, N) = 1$
- Choose $n = 10$ so that the register is big enough. Note that the bigger n we choose, the greater the accuracy of our calculations
- Initially, we start with $|\psi\rangle = |0\rangle\,|0\rangle$. Applying the QFT to the first register gives us

$$|\psi\rangle = \frac{1}{2^5} \sum_{x=0}^{2^{10}-1} |x\rangle\,|0\rangle$$

- Now compute $f(x) = a^x \mod N$ and put the result in the second register

$$|\psi\rangle = \frac{1}{2^5} \sum_{x=0}^{2^{10}-1} |x\rangle\,|f(x)\rangle$$

Factoring 39

- The values of $f(x)$ are:

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$f(x)$	1	7	10	31	22	37	25	19	16	34	4	28	1	7	10	31

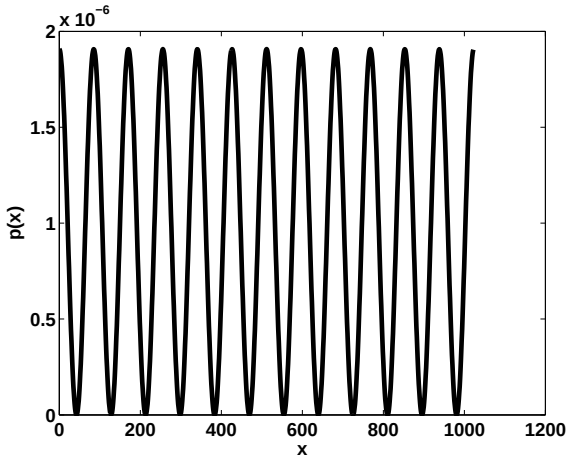
- In fact, it's quite easy to see that since we are looking for r, where $f(x) = f(x + r)$, that in this case $r = 12$
- Nonetheless, let us continue to show how the algorithm works
- We measure $f(x)$ and find (say), that $f(x) = 22$. Then:

$$|\psi\rangle = \frac{1}{\sqrt{m}} \left[|4\rangle + |16\rangle + |28\rangle + |40\rangle + \ldots\right] |22\rangle$$

- We then apply the Inverse QFT.
- This leads to a probability distribution of all possible measurements on the first qubit

Factoring 39

- The probability distribution of all possible measurements of the first register is now



- A highly likely measurement is $y = 85$

Factoring 39

- Now remember that measuring the first register will give some $y = j2^n/r$
- So $j/r \approx 85/2^{10} \approx 1/12$, giving us period $r = 12$
- The proper way to do this is to use continued fractions - see Appendix K of Mermin's book.
- So we've found the period of $f(x)$. Now let's find the factors of N
- Referring to step 5 of Shor's algorithm, we note that r is even. We compute $a^{r/2} \mod N = 7^6 \mod 39 = 25$, which is not equal to $-1 \mod N = 38$
- Finally, we compute $\gcd(a^{r/2} \pm 1, N)$, giving $\gcd(a^{r/2} - 1, N) = 3$, and $\gcd(a^{r/2} + 1, N) = 13$
- So $39 = 3 \times 13$

Some Additional Notes

- Shor's algorithm works! Most of the time.
- There are several causes of potential error
- Firstly, when you measure the first register at the end, there is a finite probability of getting a bad result (i.e. not one of the peaks)
- Secondly, j and r might have a common factor, making it impossible to find r alone
- Both of these problems can be solved with some additional complexity: see Nielsen & Chuang for more details
- In practice, you use continued fractions to find r from the final measurement. This is a little tricky and time consuming, but is what you'd do on a computer.
- On a classical computer, this would be very inefficient: it is the parallel computation of $f(x)$ which makes this algorithm fast on a quantum computer

Conclusions

In this lecture we have:

- Shown how to use a Quantum Computer to find the period of a function
- Shown how period finding can be used to compute the factors of a (large) number in an efficient manner
- Proved that $39 = 3 \times 13$!

Next lecture we will

- Show how Quantum Computers can perform efficient searches.