Quantum Computation

Lectures by Richard Jozsa

Contents

1 Review of Shor's Algorithm

This result is powered by the **quantum period finding algorithm**, and will lead us to the **hidden subgroup problem** (henceforce HSP).

1.1 Factoring Problem

Given an integer N, with $n = O(\log N)$ digits, we want to find a non-trivial factor in time complexity $O(\text{poly}(n))$.

The important concept here is that of **polynomial time complexity**: any computation has an input, from which we obtain an input *size* n. Then by polynomial time complexity, we mean that the number of steps/gates (either classical or quantum) grows only polynomially with n (*i.e.* is $O(\text{poly}(n))$).

When we refer to **efficient** computation, we are always referring to polynomial time complexity.

The best known *classical* factoring algorithm has complexity $\mathrm{e}^{O\left(n^{\frac{1}{3}}(\log n)^{\frac{2}{3}}\right)}$. However, the best known quantum algorithm (due to Shor) runs in $O(n^3)$, a considerable improvement.

1.2 Quantum Factoring Algorithm Summary

First, we convert factoring into period determination:

Given N, choose $a < N$ with $(a, N) = 1$ and consider $f : \mathbb{Z} \to \mathbb{Z}_N,\ x \mapsto a^x \mod N$. Euler's Theorem tells us that f is periodic, and the period r is the order of a modulo N, *i.e.*the least $m > 1$ such that $a^m \equiv 1 \mod N$ - this exists if and only if a, N are coprime. Through knowledge of r we are able to compute a factor of N.

While the process of determining r is *mathematically* very simple, it is in fact as difficult to compute from a classical perspective as factoring N itself. Instead we use the **Quantum algorithm for periodicity determination**.

The task: Given an oracle/black box for $f : \mathbb{Z}_M \to \mathbb{Z}_N$ with promises:

- f is periodic, with (unknown) period $r \in \mathbb{Z}_M$, *i.e.*f(x+r) = f(x) for all $x \in \mathbb{Z}_M$.
- f is $1-1$ in each period, *i.e.*$f(x_1) \neq f(x_2)$ for any $0 \leq x_1 < x_2 < r$.

We want to find r in time $O(\text{poly}(m)),\ m = \log M$ (with any prescribed success probability $1-\varepsilon$, $\varepsilon > 0$).

Remark: Queries to the oracle count as 1 step. In the quantum context we assume the oracle is a unitary gate $\mathcal{U}_f$ on $\mathcal{U}_M \otimes \mathcal{U}_N$, where $\mathcal{U}_M$ is the state space with dimension M, basis $\{\langle i|\}_{i \in \mathbb{Z}_M}$. U_f acts on basis states as

$$U_f \underbrace{\langle i|}_{\text{input}} \underbrace{\langle j|}_{\text{output}} = \langle i|\langle j + f(i)|, \qquad i \in \mathbb{Z}_M,\ j \in \mathbb{Z}_M$$

The **Query complexity** of an algorithm is the number of times the oracle is queried, which is also required to be $O(\text{poly}(m))$.

To solve the periodicity problem classically, it can be shown that it is both necessary and sufficient to query the oracle $O(\sqrt{N})$ times, so there is no polynomial algorithm. However, there *is* a quantum algorithm.

1.3 Quantum Algorithm for Periodicity Determination

For further details *c.f.* Part II notes pp.60-64.

Write $A = M/r = \#\text{periods}$. We work in the state space $\mathcal{U}_M \otimes \mathcal{U}_N$ with basis $\{\langle i|\langle k| : i \in \mathbb{Z}_M,\ k \in \mathbb{Z}_N\}$.

Step 1: obtain the state

$$\frac{1}{\sqrt{M}} \sum_{i=0}^{M-1} \langle i|\langle 0|$$

Step 2: apply U_f to obtain

$$\frac{1}{\sqrt{M}} \sum_{i=0}^{M-1} \langle i|\langle f(i)|$$

Step 3: measure the output register, obtaining result y. By the **Born rule**, the input register collapses to all those i such that $f(i) = y$, *i.e.*$i = x_0,\ x_0 + r, \cdots,\ x_0 + (A-1)r$ where $0 \leq x_0 < r$ in the first period has $f(x_0) = y$.

We discard the output reigister to obtain

$$\langle \text{per}| = \frac{1}{\sqrt{A}} \sum_{j=0}^{A-1} \langle x_0 + jr|$$

Note that each $0 \leq x_0 < r$ occurs with probability $1/r$.

If we naively measure $\langle \text{per}|$, the Born rule implies we get $x_0 + jr$ with $j = 0, \cdots, A-1$ chosen uniformly with probability $1/A$, *i.e.*a random element of a random period; this is a uniformly random integer in $\mathbb{Z}_M$. This is useless to us. Instead...

Step 4: apply **Quantum Fourier Transform** (QFT).