Quantum algorithms

Lecture 3: Quantum algorithms (2)

Benoît Vermersch

October 17, 2022

LPMMC Grenoble & IQOQI Innsbruck

Outline

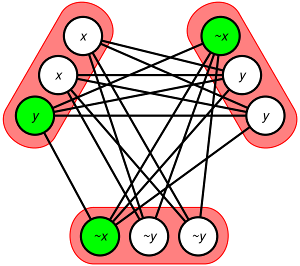
Exponential speedup: Shor's algorithm

- Reduction to order finding
- Order finding quantum subroutine

Grover's algorithm: final remark

- The quadratic speedup $\sqrt{2^n}$ of Grover's algorithm is optimal for any quantum algorithm for unstructured search (see eg Nielsen and Chuang).
- This is sad!!!: With an exponential speedup, some *NP*-complete problems could have been solved in polynomial time $\mathrm{poly}(n)$

1. Consider a *NP*-complete problem in a Boolean problem f.
2. Implement the corresponding Grover oracle.
3. Run Grover's algorithm



wikipedia

- Thus *any NP* problem could have been solved in polynomial time. . . .

Outline

Exponential speedup: Shor's algorithm

- Reduction to order finding
- Order finding quantum subroutine

Shor's algorithm

- Problem: Given N, find non-trivial factors $N = pq$.
- Complexity: Best known algorithm is sub-exponential in the number of digits $\log(N)$.
- Shor's algorithm with polynomial complexity in $\log(N)$ offers an exponential speedup.

Shor's algorithm: Number theory

- For a given $1 < a < N$, we introduce the order r, as the smallest integer such that

$$a^r = 1 \bmod(N)$$

- Theorem: If r is even, let us define $b = a^{r/2}$. If, in addition, $b \neq -1 \bmod(N)$, then

$$p = gcd(b-1, N) \text{ and } q = gcd(b+1, N) \text{ are non-trivial factors of } N$$

Shor's algorithm: Number theory

- Proof: For, e.g, $p = gcd(b-1, N)$:
  - If $p = N$, N divides $b-1$, therefore $a^{r/2} = 1 \mod(N)$, which contradicts the fact that r is the order of a.
  - If $p = 1$, there are integers (u, v) such that (Bézout's theorem)

$$(b-1)u + Nv = 1 \implies (b^2-1)u + N(b+1)v = b+1 \tag{1}$$

  - This implies N divides $b+1$, which implies $b = -1 \mod(N)$, another contradiction.

Shor's algorithm

The algorithm (1994)

1. Pick $1 < a < N$ random
2. Find order r via quantum subroutine
3. If r is even, let us define $b = a^{r/2}$. If, in addition, $b \neq -1 \mod(N)$, then $p = gcd(b-1, N)$ and $q = gcd(b+1, N)$ are non-trivial factors of N.
4. Otherwise, go back to step 1.

- Note: The *gcd* operation can be performed efficiently on a classical computer.
- Existence and 'likelihood' conditions of such even r with $b \neq -1 \mod(N)$: Beyond the scope of this course → c.f., J. Preskill's lectures

Shor's algorithm

- Example $N = 21$ (see TD2)
- We pick $a = 2$, we find $r = 6$, as $a^6 = 1 \mod(N)$.
- We have that r is even, we define $b = a^{r/2} = 8 \neq -1 \mod(21)$. We find that
- $gcd(21, 7) = 7$ divides N
- What about $N = 14351$?

Shor's algorithm

```python
from pylab import *
N = 21
for i in range(10):
    a = randint(1,N)
    r = 1
    ### I simulate here the quantum subroutine with an exponentially costly for loop
    while (r<=N):
        if (a**r)%N==1:  #Check if r is the order of (a,N)
            if r%2==0:
                print('r ', r)
                b = a**(r//2)
                print('b ',b)
                if (b+1)%N>0: print(gcd(b-1,N), ' divides ',N)
                else: print('fail')
            else: print('fail')
            print('---')
            break
        else: r+=1
```

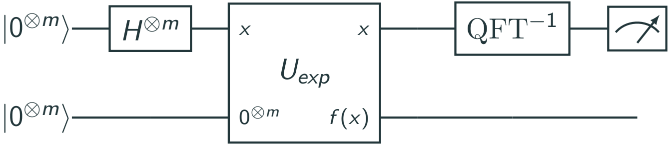
Provided I can find r efficiently. . . I find that 127 divides 14351.

Order finding quantum subroutine

- The order r is the period of the function $f(x) = a^x \mod(N)$, because $f(x + r) = a^x a^r \mod(N) = a^x \mod(N) = f(x)$.
- We can find this period up to excellent approximation using the Quantum Fourier Transformation (QFT) operation.

Order finding quantum subroutine

- Classical input: the function $f(x) = a^x \mod(N)$.
- Classical ouput: the period r.



- To provide enough 'spectral resolution', i.e., represent sufficiently large numbers x, we choose m, such that $M = 2^m > N^2$.

Order finding quantum subroutine

- The first steps, *modular exponentiation*, create the following state with order $O(m^3)$ gates, 'We load the entire function in the Hilbert space via quantum parallelism'

$$|\psi\rangle = \frac{1}{\sqrt{M}} \sum_x |x\rangle \otimes |f(x)\rangle$$

- The second step is the inverse quantum Fourier Transform, realizable with $O(m^2)$ gates see TD2), with unitary circuit

$$\mathrm{QFT}^{-1} |x\rangle = \frac{1}{\sqrt{M}} \sum_y e^{-2i\pi xy/M} |y\rangle$$

- Provided high success probability in measuring the order r, Shor's algorithm factorizes numbers in polynomial time.

Order finding quantum subroutine

- Before the measurement, the quantum state reads

$$|\psi\rangle = \frac{1}{M}\sum_{x,y} e^{-2i\pi xy/M}\,|y, f(x)\rangle \tag{2}$$

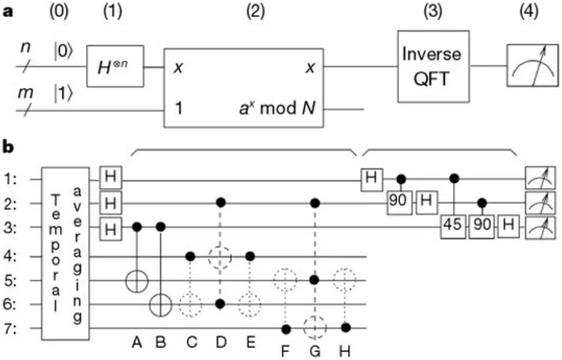
- The probability to measure the bitstring y is

$$P(y) = \sum_x |\langle y, x|\psi\rangle|^2 = \frac{1}{M^2}\sum_{x_1,x_2} e^{-2i\pi y(x_1-x_2)/M}\,\langle f(x_1)|f(x_2)\rangle \tag{3}$$

- We obtain large contributions when $x_1 - x_2 = \alpha r$, α integer. This implies $P(y)$ is maximal when $ry/M \approx p$, p integer, i.e when $y/M \approx p/r$
- The peaks $\tilde{y}$ in $P(y)$ can be used to extract $r \approx p\tilde{y}/M$ with high success probability (for a sufficiently large value of M, using the continous fraction algorithm) [see Exercices 3, for factorizing $N = 21$]

Experimental realization for $N = 15$

- Vandersypen et al, Nature 2001



- Important technological achievement
- What limits the current applications to small numbers?
  $\rightarrow$ Lecture 4: Quantum error correction