

Quantum algorithms

Lecture 2: Quantum algorithms

Benoît Vermersch

September 26, 2022

LPMMC Grenoble & IQOQI Innsbruck



Our first quantum algorithm: Deutsch's algorithm

Quadratic speedup: Grover's algorithm

Implementation details

- Groups/Schedule on Moodle
- One week after the class, Send me (Julien Renard) a commented jupyter notebook showing/explaining your results
- Grade: Jupyter notebooks+Oral exam

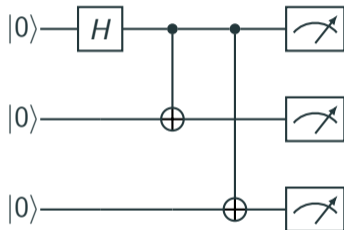
Our first quantum algorithm: Deutsch's algorithm

Quadratic speedup: Grover's algorithm

Implementation details

Reminder: Structure of a quantum circuit

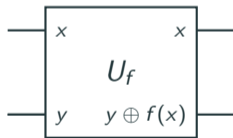
Quantum circuit: single qubit/two-qubit gates and measurements:



Algorithm: a quantum circuit to retrieve the solution of a problem in the measurement data with high probability.

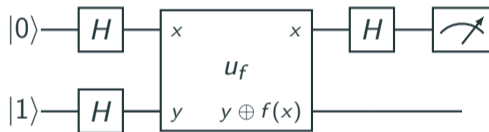
Deutsch's algorithm

- **Problem:** Given a single bit Boolean function $f(x)$, is f constant i.e $f(0) = f(1)$, or balanced, i.e $f(0) \neq f(1)$?
- To solve a classical problem with a quantum computer, we need to introduce an object called an *Oracle*, aka quantum black box.
- An oracle evaluates the classical function f on quantum states



- Complexity will refer to the number of oracles evaluation.
- **Note:** a quantum algorithm will be of practical use if the oracle can be implemented easily (see [Lecture 6](#) for a practical implementation of an oracle)

Deutsch's algorithm



One measurement gives me the solution, I would need two function evaluations in the classical case: **quantum speedup**

Deutsch's algorithm

After the first Hadamards

$$|\psi\rangle = \frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$$

After the oracle

$$|\psi\rangle' = \frac{1}{2}(|0, 0 \oplus f(0)\rangle - |0, 1 \oplus f(0)\rangle + |1, 0 \oplus f(1)\rangle - |1, 1 \oplus f(1)\rangle)$$

If $f(0) = f(1)$, let $0 \oplus f(0) = 0 \oplus f(1) = a$, $1 \oplus f(0) = 1 \oplus f(1) = b = 1 - a$

$$|\psi\rangle = \frac{1}{2}(|0\rangle + |1\rangle)(|a\rangle - |b\rangle)$$

After the last Hadamard,

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle(|a\rangle - |b\rangle)$$

I measure $|0\rangle$ with probability 1

Deutsch's algorithm

If $f(0) \neq f(1)$, let $0 \oplus f(0) = 1 \oplus f(1) = a$, $1 \oplus f(0) = 0 \oplus f(1) = b$

$$|\psi\rangle = \frac{1}{2}(|0\rangle - |1\rangle)(|a\rangle - |b\rangle)$$

After the last Hadamard,

$$|\psi\rangle = \frac{1}{\sqrt{2}}|1\rangle(|a\rangle - |b\rangle)$$

I measure $|1\rangle$ with probability 1

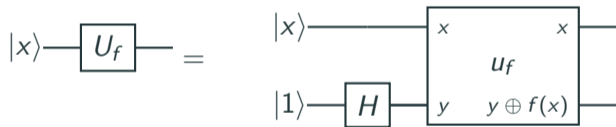
Our first quantum algorithm: Deutsch's algorithm

Quadratic speedup: Grover's algorithm

Implementation details

Grover's algorithm

- **Unstructured search problem:** Given a n -bit Boolean function $f(x)$, such that there exists a unique w such that $f(w) = 1$, find w .
- **Application:** Subroutine in various classical algorithms (example minimization problem, or machine learning)
- **Input:** A n -bit phase oracle



For any input x , we can mark to the solution

$$U_f |x\rangle = (-1)^{f(x)} |x\rangle$$

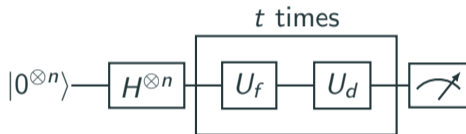
The ancilla qubit has been 'uncomputed'.

Grover's algorithm

- **Classical algorithm:** $O(2^n)$ evaluations (Just test in a loop...)
- **Grover's quantum algorithm** $O(\sqrt{2^n})$ oracle evaluations: quadratic speedup
- Possible applications: solving *NP*-complete problems that allow for oracle implementations (eg Lecture 6 on the 3-SAT problem), brute-force attacks on cryptographic keys ...

Grover's algorithm

So simple...



- with the diffuser $U_d = 2|\psi\rangle\langle\psi| - 1$, with $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ the superposition on all $N = 2^n$ bitstrings $x = x_1, \dots, x_n$.

Grover's algorithm

After the first Hadamards ($N = 2^n$), the state is

$$H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{N}}(|0\rangle + |1\rangle)^{\otimes n} = \frac{1}{\sqrt{N}} \sum_x |x\rangle = |\psi\rangle$$

Introducing, $|\alpha\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq w} |x\rangle$, we can write

$$|\psi\rangle = \sin(\theta/2) |w\rangle + \cos(\theta/2) |\alpha\rangle,$$

with $\sin(\theta/2) = 1/\sqrt{N}$.

Combined application of oracle and diffuser will lead to a rotation of the state $|\psi\rangle$ towards the solution.

$$U_f |\psi\rangle = -\sin(\theta/2) |w\rangle + \cos(\theta/2) |\alpha\rangle,$$

Grover's algorithm

$$U_d |\alpha\rangle = \cos(\theta) |\alpha\rangle + \sin(\theta) |w\rangle$$

$$U_d |w\rangle = -\cos(\theta) |w\rangle + \sin(\theta) |\alpha\rangle$$

After one iteration,

$$|\psi_1\rangle = \sin(3\theta/2) |w\rangle + \cos(3\theta/2) |\alpha\rangle$$

After t iterations,

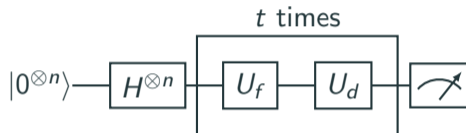
$$|\psi_t\rangle = \sin((2t + 1)\theta/2) |w\rangle + \cos((2t + 1)\theta/2) |\alpha\rangle$$

Success probability

$$p_t = |\langle w | \psi_t \rangle|^2 = \sin^2((2t + 1)\theta/2) \approx_{t \gg 1} \sin^2(\theta t/2)$$

which becomes of order one for $\theta t \approx \pi/2$, i.e $t \approx \pi\sqrt{2^n}/4$

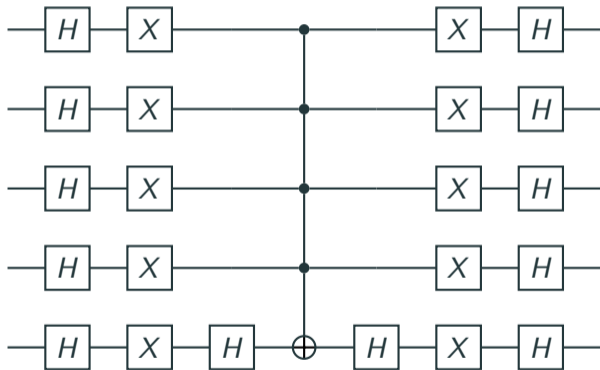
Implementation details



- Implementation of the oracle U_f depending on the function f : Careful Boolean logic to 'mark' solution without knowing the solution, eg test Boolean assertions using *CNOT*s and ancillas (Lecture 6).

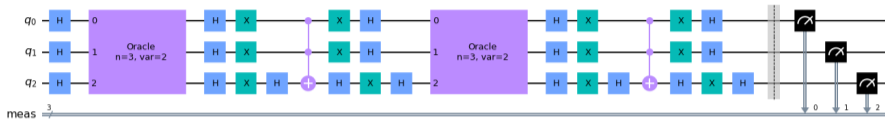
Implementation details

- Implementation of the diffuser $U_d = 2|\psi\rangle\langle\psi| - 1$: This can be done with a few gates, including a N -qubit Toffoli gate (see TD2)

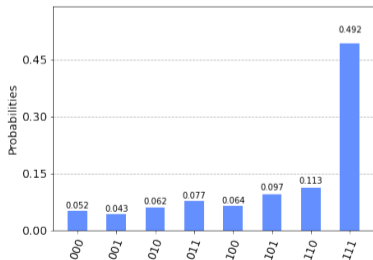


- In practice, the Toffoli gate must be decomposed in elementary CNOT gates, in an optimal way that is platform dependent

Illustration with an IBM quantum computer (c.f., Quantum Practical 2)



- The measurement gives you the solution (if errors are not too large)

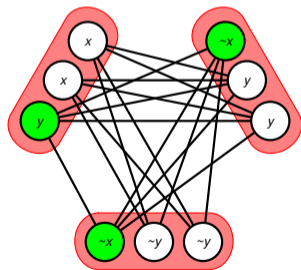


- Take-Home Message: The required number of oracle evaluations $\sim \sqrt{N}$ is smaller than the number of entries N of the database!

Grover's algorithm: final remarks

- The quadratic speedup \sqrt{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, thus *any NP* problem could have been solved in polynomial time. . . .

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



wikipedia