Quantum Computation
Amplitude amplification

Luís Soares Barbosa & Renato Neves

Universidade do Minho

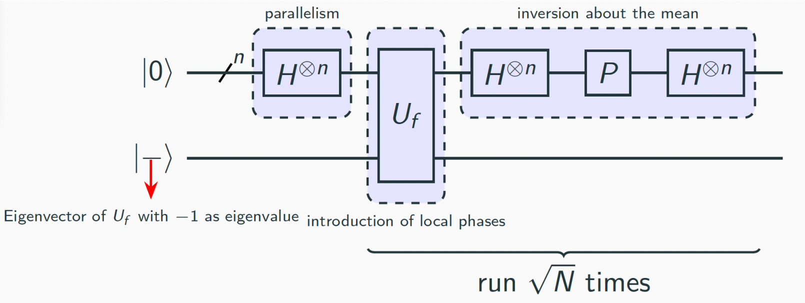
HASLab HIGH-ASSURANCE SOFTWARE LABORATORY

INTERNATIONAL IBERIAN NANOTECHNOLOGY LABORATORY

MSc Physics Engineering

Universidade do Minho, 2023-24

Recall Grover's iterator $G = WU_f$



Question

Can Grover's algorithm be generalised to search in contexts with multiple solutions?

Multiple solutions

$\boxed{\text{Assume there are } M \text{ (out of } 2^n = N\text{) input strings evaluating to 0 by } f}$

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle = \underbrace{\sqrt{\frac{M}{N}}|s\rangle}_{\text{solution}} + \underbrace{\sqrt{\frac{N-M}{N}}|r\rangle}_{\text{the rest}}$$

where

$$|s\rangle = \frac{1}{\sqrt{M}} \sum_{x \text{ solution}} |x\rangle \quad \text{and} \quad |r\rangle = \frac{1}{\sqrt{N-M}} \sum_{x \text{ no solution}} |x\rangle$$

Multiple solutions

$$t = \left| \frac{\frac{\pi}{2} - \arcsin\sqrt{\frac{M}{N}}}{2\theta} \right|$$

which, for N large, $M \ll N$ (thus $\theta \approx \sin\theta$), yields

$$t \approx \frac{\pi}{4}\sqrt{\frac{N}{M}}$$

The probability to retrieve a correct solution is

$$|\langle s|G^t|\psi\rangle|^2 \geq \cos^2\theta = 1 - \sin^2\theta = \frac{N-M}{N}$$

which, for $M = \frac{N}{2}$ yields $\frac{1}{2}$, but for $M \ll N$, is again close to 1.

Multiple solutions

Computing the effect of G: 2θ

$$2\theta = \arcsin\left(2\frac{\sqrt{M(N-M)}}{N}\right)$$

WHY?

M (out of 100)	2θ	*arcosin* 2θ
0		
1		
20		
30		
40		
50		
60		
70		
80		
99		
M		

Multiple solutions

Computing the effect of G: 2θ

$$2\theta = \arcsin\left(2\frac{\sqrt{M(N-M)}}{N}\right)$$

M (out of 100)	2θ	*arcosin* 2θ
0	0	0
1	0.198	0.199
20	0.8	0.927
30	0.916	1.158
40	0.979	1.365
50	1	1.571
60	0.979	1.365
70	0.916	1.158
80	0.8	0.927
99	0.198	0.199
M	0	0

Multiple solutions

Surprisingly, the rotation in each iteration decreases from $M = \frac{N}{2}$ to N, and the number of iterations consequently increases, although one would expect to be easier to find a correct solution if their number increases!

Solution: resort to draft paper!

To double the number of elements in the search space, by adding N extra elements, none of which being a solution.

The technique: Amplitude amplification

Grover's algorithm made use of

$$H^{\otimes n}|0\rangle$$

to prepare a uniform superposition of potential solutions.

In general, one may resort to any program K to map the solution space to any superposition of guesses, plus some extra qubits to be used as draft paper:

$$K|0\rangle = \sum_x \alpha_x |x\rangle\,|\mathrm{draft}(x)\rangle$$

The technique: Amplitude amplification

$$|\psi\rangle = \sum_{x \text{ solution}} \alpha_x |x\rangle \, |\text{draft}(x)\rangle + \sum_{x \text{ no solution}} \alpha_x |x\rangle \, |\text{draft}(x)\rangle$$

yielding the following probabilities:

$$p_s = \sum_{x \text{ solution}} |\alpha_x|^2 \quad \text{and} \quad p_{ns} = \sum_{x \text{ no solution}} |\alpha_x|^2 = 1 - p_s$$

Of course, amplification has no use if $p_s \in \{0, 1\}$.

The technique: Amplitude amplification

Otherwise ($0 < p_s < 1$), the amplitudes of solution inputs should be amplified.
First, express

$$|\psi\rangle \;=\; \sqrt{p_s}|\psi_s\rangle \;+\; \sqrt{p_{ns}}|\psi_{ns}\rangle$$

for the normalised components

$$\begin{aligned}
|\psi_s\rangle \;&=\; \sum_{x \text{ solution}} \frac{\alpha_x}{\sqrt{p_s}}|x\rangle\,|\text{draft}(x)\rangle \\
|\psi_{ns}\rangle \;&=\; \sum_{x \text{ solution}} \frac{\alpha_x}{\sqrt{p_{ns}}}|x\rangle\,|\text{draft}(x)\rangle
\end{aligned}$$

which rewrites to

$$|\psi\rangle \;=\; \sin\theta|\psi_s\rangle \;+\; \cos\theta|\psi_{ns}\rangle$$

for $\theta \in [0, \frac{\pi}{2}]$ such that $\sin^2\theta = p_s$.

The technique: Amplitude amplification

A generic search iterator is built as

$$S = KPK^{-1}V = W_K V$$

where

$$\begin{aligned} W_K|\psi\rangle &= |\psi\rangle \\ W_K|\phi\rangle &= -|\phi\rangle \quad \text{for all states orthogonal to } |\psi\rangle \end{aligned}$$

The sets $\{|\psi_s\rangle, |\psi_{ns}\rangle\}$ and $\{|\psi\rangle, |\overline{\psi}\rangle\}$ are bases for the relevant 2-dimensional subspace.

The technique: Amplitude amplification

As expected, starting in $|\psi\rangle$, the oracle produces

$$-\sin\theta|\psi_s\rangle \;+\; \cos\theta|\psi_{ns}\rangle \;=\; \cos(2\theta)|\psi\rangle - \sin(2\theta)|\overline{\psi}\rangle$$

which, followed by the amplifier, yields

$$\cos(2\theta)|\psi\rangle + \sin(2\theta)|\overline{\psi}\rangle$$

i.e. the effect of iterator S is

$$S|\psi\rangle \;=\; \cos(2\theta)|\psi\rangle + \sin(2\theta)|\overline{\psi}\rangle$$

The technique: Amplitude amplification

Exercise

Show that

$$S|\psi\rangle \;=\; \cos(2\theta)|\psi\rangle + \sin(2\theta)|\overline{\psi}\rangle$$

can be expressed in the basis $\{|\psi_s\rangle, |\psi_{ns}\rangle\}$ as

$$S|\psi\rangle \;=\; \sin(3\theta)|\psi_s\rangle + \cos(3\theta)|\psi_{ns}\rangle$$

The technique: Amplitude amplification

$$
\begin{aligned}
S|\psi\rangle &= \cos(2\theta)|\psi\rangle + \sin(2\theta)|\overline{\psi}\rangle \\
&= \cos(2\theta)(\sin\theta|\psi_s\rangle + \cos\theta|\psi_{ns}\rangle) + \sin(2\theta)(\cos\theta|\psi_s\rangle - \sin\theta|\psi_{ns}\rangle) \\
&= \cos(2\theta)\sin\theta|\psi_s\rangle + \cos(2\theta)\cos\theta|\psi_{ns}\rangle + \sin(2\theta)\cos\theta|\psi_s\rangle - \sin(2\theta)\sin\theta|\psi_{ns}\rangle \\
&= (\cos(2\theta)\sin\theta + \sin(2\theta)\cos\theta)|\psi_s\rangle + (\cos(2\theta)\cos\theta - \sin(2\theta)\sin\theta)|\psi_{ns}\rangle \\
&= ((\cos^2\theta - \sin^2\theta)\sin\theta + \sin(2\theta)\cos\theta)|\psi_s\rangle + ((\cos^2\theta - \sin^2\theta)\cos\theta - \sin(2\theta)\sin\theta)|\psi_{ns}\rangle \\
&= (\cos^2\theta\sin\theta - \sin^2\theta\sin\theta + \sin(2\theta)\cos\theta)|\psi_s\rangle + (\cos^2\theta\cos\theta - \sin^2\theta\cos\theta - \sin(2\theta)\sin\theta)|\psi_{ns}\rangle \\
&= (\cos^2\theta\sin\theta - \sin^3\theta + 2\sin\theta\cos^2\theta)|\psi_s\rangle + (\cos^3\theta - \sin^2\theta\cos\theta - 2\sin^2\theta\cos\theta)|\psi_{ns}\rangle \\
&= (\cos^2\theta\sin\theta - \sin^3\theta + 2\sin\theta\cos^2\theta)|\psi_s\rangle + (\cos^3\theta - \sin^2\theta\cos\theta - 2\sin^2\theta\cos\theta)|\psi_{ns}\rangle \\
&= (3\cos^2\theta\sin\theta - \sin^3\theta)|\psi_s\rangle + (\cos^3\theta - 3\sin^2\theta\cos\theta)|\psi_{ns}\rangle \\
&= (3(1-\sin^2\theta)\sin\theta - \sin^3\theta)|\psi_s\rangle + (\cos^3\theta - 3(1-\cos^2\theta)\cos\theta)|\psi_{ns}\rangle \\
&= (3\sin\theta - 3\sin^3\theta - \sin^3\theta)|\psi_s\rangle + (\cos^3\theta - 3\cos\theta + 3\cos^2\theta)|\psi_{ns}\rangle \\
&= (3\sin\theta - 4\sin^3\theta)|\psi_s\rangle + (4\cos^3\theta - 3\cos\theta)|\psi_{ns}\rangle \\
&= \sin(3\theta)|\psi_s\rangle + \cos(3\theta)|\psi_{ns}\rangle
\end{aligned}
$$

The technique: Amplitude amplification

By an inductive argument, the repeated application of S a total of k times rotates the initial state $|\psi\rangle$ to

$$S^k|\psi\rangle = \sin((2k+1)\theta)|\psi_s\rangle + \cos((2k+1)\theta)|\psi_{ns}\rangle$$

For the correct number of iterations, this procedure reaches a state such that a measurement will return an element of the subspace spanned by $|\psi_s\rangle$ with a probability close to 1.

The technique: Amplitude amplification

As before, to get that high probability, the smallest value for k one can choose is such that

$$(2k+1)\theta \approx \frac{\pi}{2}$$

For a small θ, as

$$\sin\theta = \sqrt{p_s} \approx \theta$$

the magnitude of the right number of iterations is

$$\mathcal{O}\left(\sqrt{\frac{1}{\theta}}\right)$$

because

$$(2k+1)\sqrt{p_s} = \theta \Leftrightarrow k = \frac{\pi}{4\sqrt{p_s}} - \frac{1}{2}$$

To follow

The algorithm requires that one knows in advance how many times iterator S is to be applied:

- For $K = H$ (uniform sampling the input) this boils down to know the number of solutions of the search problem.
- For a generic K this amounts to know the probability with which K guesses a solution to the problem, i.e. $\sin(\theta)$.

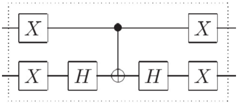
To see ...

- blind search
- estimate the amplitude with which K maps $|0\rangle$ to the subspace of solutions

PROBLEM 1

Consider a search space $N = 4$, with $M = 2$.

- How many iterations are required to find the correct solution with high probability? Why? Which is the angle of the rotation in each iteration?
- How many queries to the oracle would be necessary under a classical computer?
- Discuss whether the circuit below performs the phase shift operation $2|0\rangle\langle 0| - I$, up to an irrelevant global phase factor.



- Prove the Grover iterator G is, as expected, unitary.

PROBLEM 2

SAT (= Boolean satisfiability) problems

Determining values for Boolean variables so that a given Boolean expression evaluates to true

- NP-complete
- Many problems, like scheduling, can be converted into a SAT
- Can be seen as a search problem whose goal is to find a precise combination of Boolean values that yields true

PROBLEM 2

Mini project

Implement Grover's Algorithm in Qiskit to find a satisfying assignment containing one true literal per clause.

- INPUT: SAT formula in conjunctive normal form, i.e. a conjunction of disjunctive clauses $\bigvee_{k=1..m} \phi_k$ over n Boolean variables with 3 literals per clause.
- OUTPUT: Is there an assignment to the n Boolean variables such that every clause has exactly one true literal?

PROBLEM 2

Note: Creating a uniform superposition of all basis states does not allow to satisfactorily solve NP-complete problems

Let U_f encode a SAT formula on n Boolean variables:

$$U_f(|i\rangle \otimes |0\rangle) = |i\rangle \otimes |f(i)\rangle$$

Applying U_f to a superposition obtained via $H^{\otimes n}|0\rangle$, which evaluates the truth assignment of all possible binary strings, will return a binary string that satisfies the formula iff the last qubit has value 1 after the measurement, and this happens with a probability that depends on the number of binary assignments that satisfy the formula
(e.g. $\frac{\tau}{2^n}$, for τ such assignments).

Second thoughts

Although, in general, solving NP-hard problems in polynomial time with quantum computers is probably not possible (cf $P = NP$?), there is a recipe to produce faster equivalent quantum algorithms:

- Create a uniform superposition of basis states
- Make the basis states interact with each other so that the modulus of the coefficients for some (desirable) basis states increase, which implies that the other coefficients decrease.
- How to do it ... depends on the problem