Lecture 6: Quantum oracles & Recent experiments with quantum computers

Quantum Algorithms

Benoît Vermersch

November 25, 2022

LPMMC Grenoble & IQOQI Innsbruck

Université Grenoble Alpes

Outline

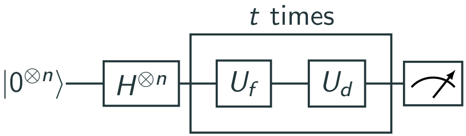
Implementing a quantum oracle

Recent experimental breakthroughs with quantum computers

- Presentation of a superconducting qubit quantum computer
- The quantum supremacy experiment (Arute et al, 2019)

Grover's oracle: Reminder

- Problem given a n-bit Boolean function $f(x)$. Find the solution $x = w$ such that $f(w) = 1$
- The quantum algorithm is given by



Grover's oracle: Reminder

- Grover's algorithm converges to the solution for $t \propto \sqrt{N = 2^n}$.
- The diffuser $U_d = 2\left|\psi\right\rangle\left\langle\psi\right| - 1$ can be implemented with the Toffoli gate
- How can we implement an oracle $U_f\left|x\right\rangle = (-1)^{f(x)}\left|x\right\rangle$? (without knowing the solution $w\ldots$)?
- Let us show that this is sometimes possible using elementary bitstring operations and the technique of 'uncomputation'.

Case study: p-SAT Problem

- p-SAT: Given n bits, we are looking for the bit strings $x = (x_1, \dots, x_n)$ that satisfy the Boolean function

$$f(x) = C_1(x) \wedge \cdots \wedge C_M(x), \tag{1}$$

  where $\wedge$ is a conjunction (AND).
- Each clause $C_m(x)$ is made of a disjunction (OR: $\vee$) of at most p litterals.
- Example with $M = 3, n = 2, p = 2$

$$f(x) = (x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \tag{2}$$

- 3-SAT is NP-Complete.

Implementing a Grover's oracle for 2-SAT

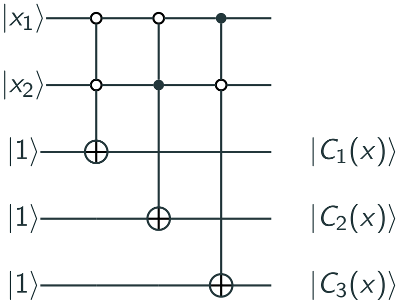
- Oracle $U_f |x\rangle = (-1)^{f(x)} |x\rangle$ with $f(x) = (x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2)$
- Step 1: Use de Morgan's law $A \vee B = \neg(\neg A \wedge \neg B)$

$$f(x) = \neg(\neg x_1 \wedge \neg x_2) \wedge \neg(\neg x_1 \wedge x_2) \wedge \neg(x_1 \wedge \neg x_2) \tag{3}$$

Implementing a Grover's oracle for 2-SAT

- Step 2: Add one ancilla qubit and Tofolli gates to test each clause

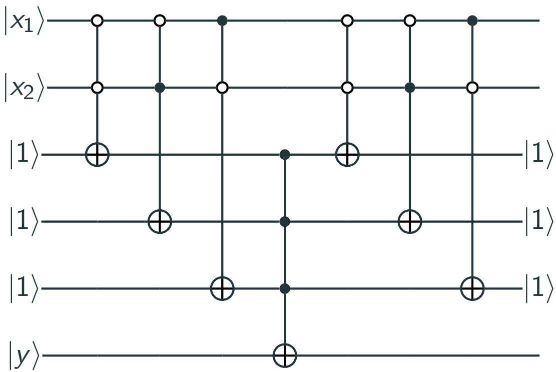
$$|x\rangle \otimes |000\rangle \rightarrow |x\rangle \otimes |C_1(x)\rangle\,|C_2(x)\rangle\,|C_3(x)\rangle \tag{4}$$



Implementing a Grover's oracle for 2-SAT

- Step 3: Add an extra ancilla bit for performing the conjunctions between clauses and 'uncompute' the first ancilla qubits

$$|x\rangle |1\rangle^{\otimes M} |y\rangle \rightarrow |x\rangle |1\rangle^{\otimes M} |y \oplus f(x)\rangle \tag{5}$$



- If the last ancilla is flipped, $f(x) = 1$, i.e., we can mark the solution!

Implementing a Grover's oracle for 2-SAT

- All ancilla qubits except one have been uncomputed, we have effectively realized a *XOR* oracle

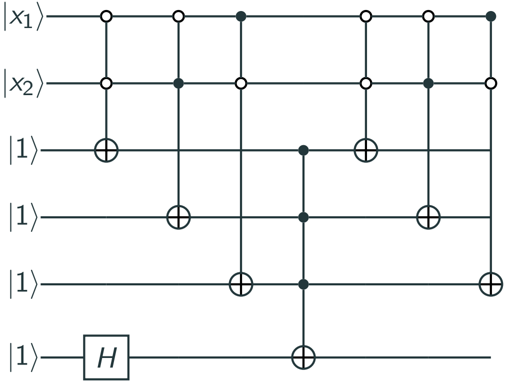
$$|x\rangle \otimes |y\rangle \rightarrow |x\rangle \otimes |y \oplus f(x)\rangle \tag{6}$$

- How to realize a *phase* oracle, as required for Grover's oracle, from a XOR oracle?

$$|x\rangle \rightarrow (-1)^{f(x)} |x\rangle \tag{7}$$

Implementing a Grover's oracle for 2-SAT

- XOR to phase oracle conversion: Simply initialize the last ancilla using the H gate



Implementing a Grover's oracle for 2-SAT

- We obtain

$$
\begin{aligned}
|x\rangle |1\rangle^{\otimes M} (H|1\rangle) &\rightarrow |x\rangle |1\rangle^{\otimes M} (|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) \\
&\rightarrow |x\rangle |1\rangle^{\otimes M} (-1)^{f(x)} (|0\rangle - |1\rangle) \\
&\rightarrow (-1)^{f(x)} |x\rangle |1\rangle^{\otimes M} (H|1\rangle) \qquad (8)
\end{aligned}
$$

- which effectively realizes our oracle (and uncomputes the last ancilla)!

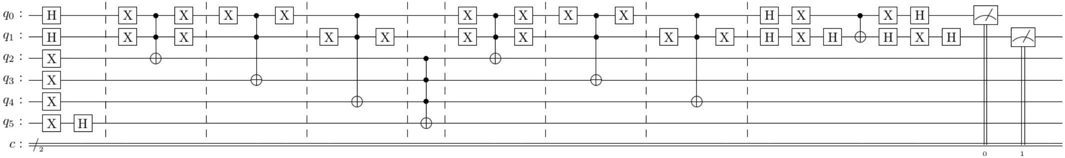
$$
|x\rangle \rightarrow (-1)^{f(x)} |x\rangle \qquad (9)
$$

The full circuit

- Problem function:

$$f(x) = (x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \tag{10}$$

- Circuit



- Qiskit output: bistring 11 with probability 1.0: I coded the problem without knowing the solution, then the algorithm gives me the solution.

Outline

Implementing a quantum oracle

Recent experimental breakthroughs with quantum computers

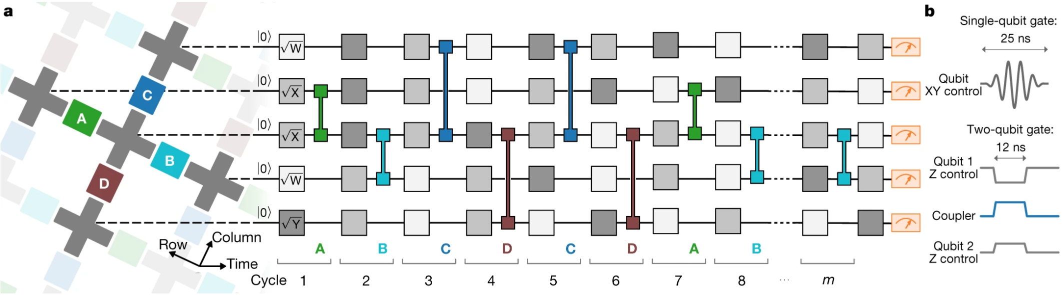
- Presentation of a superconducting qubit quantum computer
- The quantum supremacy experiment (Arute et al, 2019)

How does it work?

- Superconducting material+Josephson junctions
  $\rightarrow$ 53 'anharmonic oscillators' $h_i = \omega_0 a_i^\dagger a_i + U a_i^\dagger a_i (a_i^\dagger a_i - 1) + \dots$
  $\rightarrow$ Reviews by Devoret, Girvin, etc
- Cool to 20 mK temperature via a dilution fridge
- We can control the qubit $|0\rangle_i\,, |1\rangle_i = a_i^\dagger\,|0\rangle_i$ with 'single qubit gates'.
- iSwap gates between qubits $U_{i,j} = e^{i(\pi/4)(\sigma_i^x \sigma_j^x + \sigma_i^y \sigma_j^y)}$
- Each qubit can be measured.
- We have a universal quantum computer: Every N-qubit state (dim 2^N) can be created and measured (Deutsch 1989).

$$|\psi\rangle = \sum_{s_1,\dots,s_N} c_{s_1,\dots,s_N} |s_1\rangle \otimes \cdots \otimes |s_N\rangle \tag{11}$$

Programming a quantum computer



The quantum computing roadmap

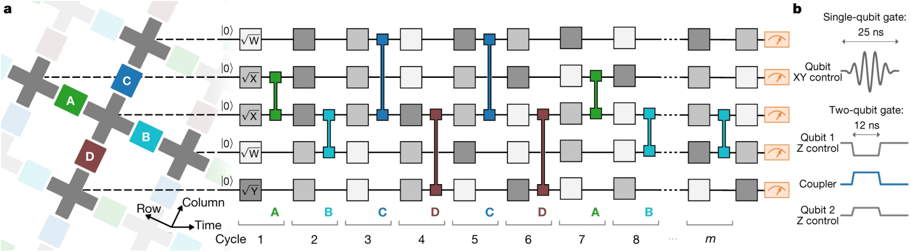
- Long-terms goals: Quantum algorithms to solve hard problems
  - Data search (Grover's 1996)
  - Factorization (Shor's 1995)
  - Large-scale optimization problems (on-going debate)
- Outstanding conceptual/technological challenges
  - Error propagation is typically exponential in problem size
  - Sophisticated algorithms for 'quantum error correction' required to achieve fault tolerance.
- Massive investments (USA, China, Europe, etc) are helpful, but should not hide conceptual challenges ('quantum hype').
- We need verification methods to validate technological progress

What is this about?

- It will probably take years before we can run a quantum algorithm solving an open problem in computer science.
- An important milestone is to check that a quantum computer is able to perform a computation that is not achievable via a classical computer with reasonable resources (Preskill 2012).
- Two important conceptual questions
  - How can I check something that I cannot compute?
  - How can I explore an gigantic Hilbert space of dimension $2^{53} \sim 10^{15}$ with a finite number of measurements?

Idea

- Use random circuits that are difficult to simulate classically for $N > 50$
- Make use of random matrix theory to create a faithful figure of merit, based on measuring the system in terms of 'bitstrings' (ex $s = 01001\ldots$).



Algorithm

1. Consider a reduced or 'shallow' circuit U that I can compute classically.
   1.1 Implement $|\psi\rangle = U|0\rangle^{\otimes N}$
   1.2 Measure $M \ll 2^N$ bitstrings $\{s_m\}$ (sampled ideally according to $P(s) = |\langle s|\psi|\rangle|^2$).
   1.3 Evaluate the fidelity based on computing

$$\mathcal{F}_{\mathrm{XEB}} = \frac{2^N}{M}\sum_{m=1}^{M} P(s_m) - 1 \tag{12}$$

2. Consider a non-simulatable 'supremacy' circuit U
   2.1 Measure the bitstrings $\{s_m\}$ as above
   2.2 Archive the results, waiting for the classical computer to 'catch up' and allow for the evaluation of the fidelity.

The test is meaningful

- For sufficiently large M,

$$\mathcal{F}_{\mathrm{XEB}} = \frac{2^N}{M}\sum_{m=1}^{M} P(s_m) - 1 = 2^N \sum_{s=0}^{2^N-1}\sum_{m=1}^{M}\frac{\delta_{s,s_m}}{M}P(s) - 1 \approx 2^N \sum_{s=0}^{2^N-1} P(s)^2 - 1 \quad (13)$$

- For a random circuit this sum can be calculated analytically.

The test is meaningful

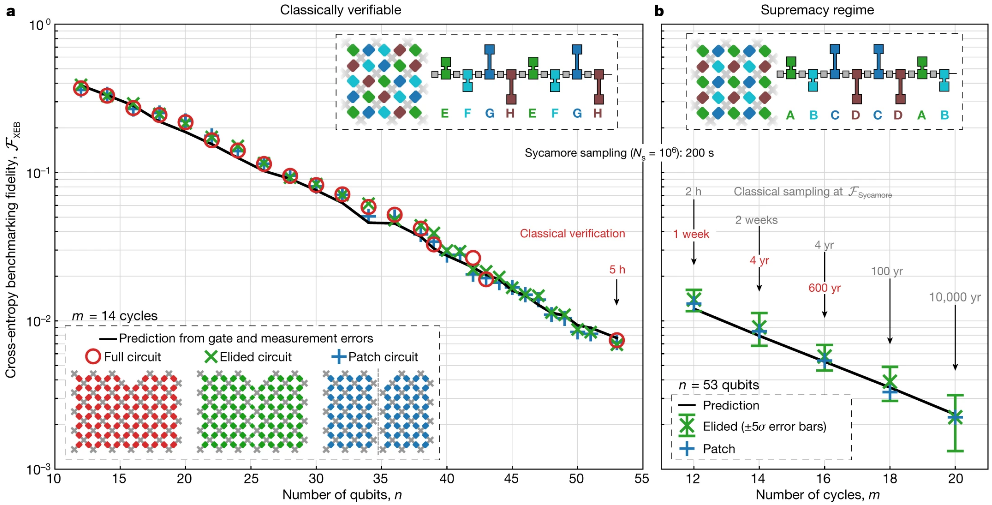
For sufficiently large N,

$$\mathcal{F}_{\mathrm{XEB}} \approx 2^N \sum_{s=0}^{2^N-1} P(s)^2 - 1 \approx 2^{2N} \langle P(s)^2 \rangle - 1 \tag{14}$$

$$\mathrm{Prob}(P(s) \in [p, p+dp]) = 2^N exp(-2^N p) dp \tag{15}$$

- In average, $\langle P(s) \rangle = 2^{-N}$, as for a uniform distribution $P_{\mathrm{uni}}(s) = 2^{-N}$.
- However $\langle P(s)^2 \rangle = 2 \times 2^{-2N}$, *twice* compared to $P_{\mathrm{uni}}(s)^2 = 2^{-2N}$!
- Therefore, $\mathcal{F}_{\mathrm{XEB}} = 1$. For uniform distribution, we get instead $\mathcal{F}_{\mathrm{XEB}} = 0$.
- Fast convergence of the estimation of $\mathcal{F}_{\mathrm{XEB}}$ with $\sim 10^6$ measurements.

Results


a
Classically verifiable
b
Supremacy regime
E F G H E F G H
A B C D C D A B
Sycamore sampling (N_s = 10^6): 200 s
Classical verification
5 h
Classical sampling at $\mathcal{F}_{\text{Sycamore}}$
2 h
1 week
2 weeks
4 yr
4 yr
600 yr
100 yr
10,000 yr
m = 14 cycles
Prediction from gate and measurement errors
Full circuit
Elided circuit
Patch circuit
n = 53 qubits
Prediction
Elided (±5σ error bars)
Patch
Cross-entropy benchmarking fidelity, $\mathcal{F}_{\text{XEB}}$
Number of qubits, n
Number of cycles, m

Conclusion

- A remarkable technological achievement.
- Exponential propagation of errors. This was expected
- The power of classical simulations was underestimated (see eg X. Waintal et al, PRX 2020).