THE EFFECTS OF NOISE ON QUANTUM ALGORITHMS SOLVING 3SAT

Universiteit Leiden
The Netherlands

MARTIJN SWENNE (SUPERVIZED BY VEDRAN DUNJKO)

LEIDEN INSTITUTE OF ADVANCED COMPUTER SCIENCE

s1923889@liacs.leidenuniv.nl

OVERVIEW

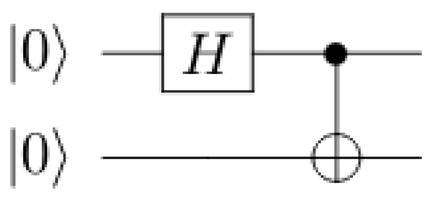
QUANTUM is a term you hear a lot lately. Quantum computing uses quantum-mechanical phenomena, such as superposition and entanglement. A quantum computer is therefor completely different than a classical computer. Whereas a classical computer uses binary digits, which are always in one of two definite states (0 or 1), a quantum computer uses quantum bits, or QUBITS. These qubits can be in a superposition of states. Quantum computers are extremely powerful, but also very noisy. This is currently a big problem in quantum computers.

BASICS

A QUANTUM COMPUTER operates on qubits using quantum gates and measurement. It maintains a sequence of n qubits, which can represent a 0, a 1 or any quantum superposition of those two states. Measurements collapse the system of qubits into one of the 2^n states, decomposing into a classical state. Quantum algorithms are often probabilistic. They provide the correct solution only with a certain known probability.
Qubits are represented in the DIRAC notation. There is the ket $|0\rangle$ and the hermitan conjugate called the bra $\langle 0|$. Qubits and quantum gates can be described with matrices. For instance:
$|0\rangle = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$ and $|1\rangle = \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$
Just as in classical computations, where you work with classical logic gates, quantum computations work with quantum logic gates, which are 1 or 2 qubit gates. Unlike classical logic gates, these gates are reversible. For instance, a basic quantum logic gate is the Hadamard gate:

$$H|0\rangle = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix}1 & 1\\1 & -1\end{smallmatrix}\right]\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right] = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right] = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$$
$$H|1\rangle = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix}1 & 1\\1 & -1\end{smallmatrix}\right]\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right] = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix}1\\-1\end{smallmatrix}\right] = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

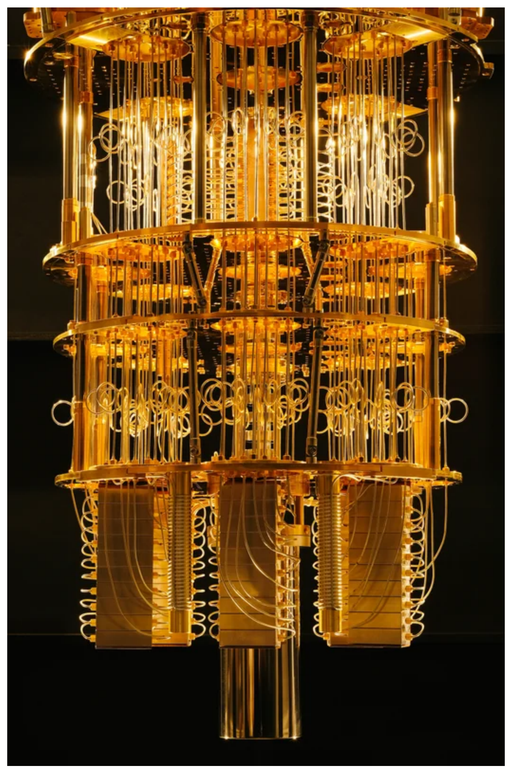
The Hadamard gate puts the qubit in a superposition of basis states. The probability that the qubit is $|0\rangle$ is $(\frac{1}{\sqrt{2}})^2 = 0.5$. The probability for $|1\rangle$ is also 0.5. There is another basic quantum gate, which is called the CNOT-gate. This is a controlled-NOT gate. The gate flips the state of the target qubit if and only if the control qubit is 1.



If you now apply a Hadamard gate on the control qubit before a CNOT-gate, the two qubits become ENTANGLED. This means that you cannot describe a qubit independently of the state of the other qubit. If you were to measure the state of the first qubit, it immediately tells you the state of the second qubit, because thats our only possibility. This representation is called a Bell-state, which is a maximally entangled state.

QUANTUM COMPUTER

The image below is a quantum computer from IBM.



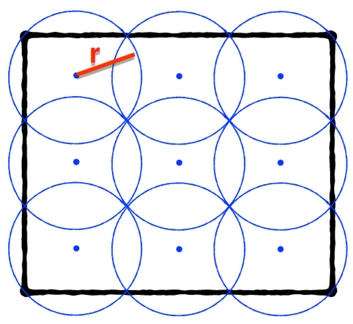
IBM Research Flickr (2018)

Below is an example of code to implement a quantum circuit. This is a python library called QISKIT.

```
10      q = QuantumRegister(n)
11      qc = QuantumCircuit(q)
12
13      qc.cx(q[2],q[1])
14      qc.barrier()
15      cx_sq(qc,q,0,1, True)
16      qc.barrier()
17      qc.cx(q[2],q[1])
18      qc.barrier()
19      cx_sq(qc,q,0,1, False)
20      qc.barrier()
21      cx_sq(qc,q,0,2, False)
```

PROMISEBALL

The PROMISEBALLSAT (or PBS) is a derandomized Schöning algorithm for a ($\leq$k)-CNF formula. Given the randomized initial assignment x, let $B_r(x)$ denote the r-ball centered at x, i.e. the set of all bitstrings y has a HAMMING DISTANCE $\leq r$. The space of possible assignments is covered by a number of r-balls.



Vedran Dunjko (2018)

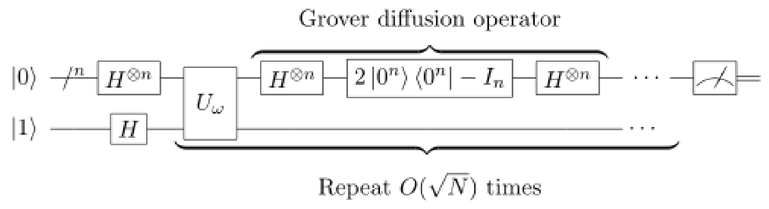
Given such a covering set, specified by the centers of the balls, the algorithm sequentially checks whether there exists a satisfying assignment within each of the r-balls. The deterministic algorithm $PromiseBall(F, r, x)$ is a simple, recursive divide-and-conquer algorithm. It first checks some obvious conditions for (un)satisfiability, or if x is a satisfying assignment. Otherwise, in the recursive step, it finds the first unsatisfied clause C, and calls $Promise - Ball(F^{[l=1]}, r1, x)$ for every literal $l \in C$. Here,$F^{[l=1]}$ denotes the formula obtained by setting the variable corresponding to l to the value ensuring $l = 1$,i.e., all clauses involving l are removed (or truncated).

SOLVING 3-SAT

The BOOLEAN SATISFIABILITY problem is the problem of determining if there exists an interpretation that satisfies a given Boolean formula. SAT is proven to be NP-complete. This means that it is solvable in polynomial time.
A special case of boolean satisfiability is the 3-SAT. This means the boolean formula is in conjunctive normal form where each clause is limited to at most three literals. Each clause has to be satisfied, while in every clause only one out of the three literals has to be satisfied. Such a boolean formula can be written like:

$$(l_1 \vee l_2 \vee l_3) \wedge \cdots \wedge (l_1 \vee l_2 \vee l_3)$$

For the 3-SAT there exists a quantum algorithm that solves it brute force in $O(2^{n/2})$, while a classical brute force solves it in $O(2^n)$. This quantum algorithm is called GROVER'S SEARCH.



Wikipedia; Grover's algorithm

Although this is significantly better than a classical brute force, the best classical algorithm for 3-SAT, the SCHÖNING algorithm (walkSAT), solves it in $O(\frac{4}{3}^n) = O(2^{0.415*n})$. This means that a brute force quantum algorithm is still slower than the fastest classical algorithm.

QUANTUM NOISE

QUANTUM DECOHERENCE, or quantum noise, happens when qubits lose information to the environment over time. This doesn't start until the qubits are used, like measuring them or performing a computation.
There's a certain threshold for noise, which is called FAULT TOLERANCE, where quantum computers will theoretically be reliable enough to be considered useful. Up to this moment, there is no good solution to this problem. Quantum noise currently gets simulated by adding random gates (such as the Pauli X gate) into the circuit.

FURTHER RESEARCH

[1] A full derandomization of schöning's k-SAT algorithm; https://dl.acm.org/citation.cfm?doid=1993636.1993670

[2] Quantum Computing in the NISQ era and beyond; https://arxiv.org/abs/1801.00862

[3] 3-SAT Faster and Simpler - Unique-SAT Bounds for PPSZ Hold in General; https://epubs.siam.org/doi/10.1137/120868177

[4] Computational speedup using small quantum devices; https://arxiv.org/pdf/1807.08970.pdf