Investigation into the applicability of Grover's Algorithm to search problems

S. Al Hashemi

Engineering Science

Supervisor: Prof. G. Gulak

April 2019

Outline

Motivation and Objective

Motivation

- Encryption cracking for encryption schemes based on the Shortest Vector Problem (Lattice Schemes).
- Fast global optimization of mathmatical functions.
- Machine Learning
- Quantum Chemistry
- Searching problems

Objective

- Gain insight into the hardware requirements necessary to implement Grover's algorithm.
- Develop test quantum circuits for known problems.

Quantum Computers

- First proposed by Richard Feynman
- Takes advantage of several quantum theories to solve problems not computable in polynomial time on classical computers.
    - **Superposition** of quantum state
    - **Quantum entanglement** of state

Comparison with classical computing

Classical Computers

- Built on semiconducting material (ie. silicon)
- Encode information into bits which operate by manipulating charge.
- **Deterministic** in nature $\rightarrow$ you can know the state of a bit or a group of bits before any measurement is made.

Quantum Computers

- Construction depends on the model $\rightarrow$ popular forms of the qubit model are built on superconducting material.
- Encode information into qubits/qumodes.
- **Non-deterministic** $\rightarrow$ probabilistic by nature.

Qubits

Quantum Basis of Qubits

The state of a qubit is naturally described by a discrete basis of $|0\rangle, |1\rangle \rightarrow$ the ground/excited states.

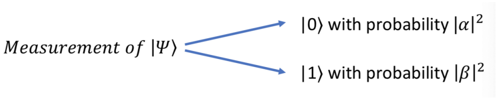
- The general state of a qubit is thus:

$$|\Psi\rangle = \alpha |0\rangle + \beta |1\rangle \tag{1}$$

- This can be further represented as:

$$|\Psi\rangle \rightarrow \begin{bmatrix} \alpha \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \beta \end{bmatrix} \tag{2}$$

- **Normalization Constraint**: $|\alpha|^2 + |\beta|^2 = 1$



$Measurement\ of\ |\Psi\rangle$
$|0\rangle$ with probability $|\alpha|^2$
$|1\rangle$ with probability $|\beta|^2$

Quantum Gates

Quantum Gates

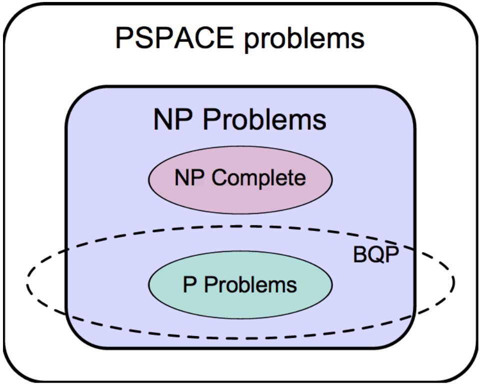
- Unitary operations that modify the state of a qubit(s)
- Must preserve probability amplitudes normalization.

NOT Gate	*HADAMARD Gate*
$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
$X\left\|0\right\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \left\|1\right\rangle$	$H\left\|0\right\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\left\|0\right\rangle + \frac{1}{\sqrt{2}}\left\|1\right\rangle$

Time Complexity

Quantum Complexity

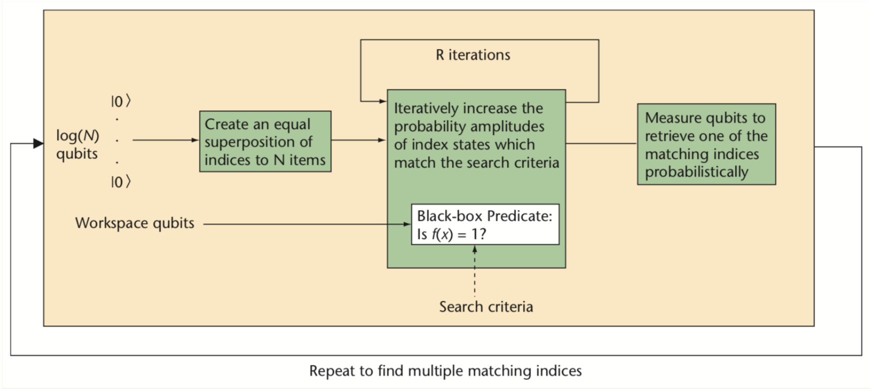
Bounded-Error Quantum Polynomial Time (BQP) is a classification for a set of problems that require a polynomial amount of resources on a quantum computer.



Grover's Algorithm

Problem Statement

Given an input space $X_n = \{x_0, x_1, x_2, \ldots, x_n\}$ to an oracle function, $f(x) \rightarrow \{1, 0\}$ it is the goal of the algorithm to find the target element x^* such that $f(x^*) = 1$

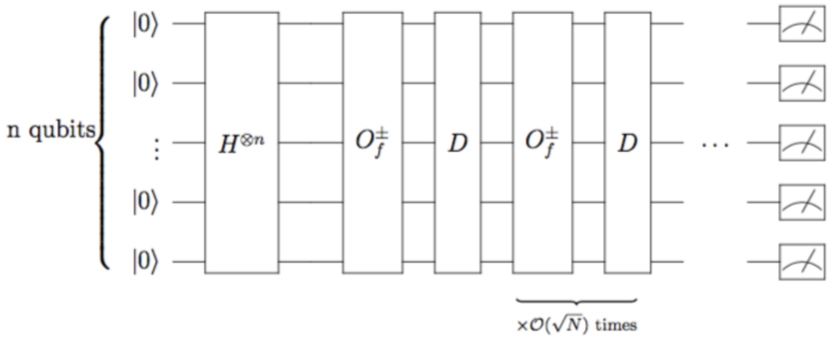

R iterations
|0⟩
log(N) qubits
|0⟩
Create an equal superposition of indices to N items
Iteratively increase the probability amplitudes of index states which match the search criteria
Measure qubits to retrieve one of the matching indices probabilistically
Workspace qubits
Black-box Predicate: Is f(x) = 1?
Search criteria
Repeat to find multiple matching indices

Components of Grover's Algorithm

Breakdown

- The Oracle query step, O_f
- Amplitude Amplification, D

```
Initialize equal superposition of
 states for $i$ in range($\sqrt{N}$) do
    Perform Oracle Query Apply
     Diffusion Operator
end
```



Oracle Query

- It's easy to say that there is a need to have a function that distinguishes between good and bad inputs, s.t. for a unique input x^*, $f(x^*) = 1$, but how is this accomplished on a quantum computer?
- An input that satisfies the search criterion can have its phase amplitude, α_i flipped. That is:

$$\alpha_i \rightarrow -\alpha_i \tag{3}$$

Grover's Diffusion Gate

Amplitude Amplification

Grover's Diffusion Gate applies this mapping:

$$\alpha_x |x\rangle \rightarrow (2\mu - \alpha_x) |x\rangle \tag{4}$$

Here, $\mu = \frac{1}{N} \sum_x \alpha_x \forall x \in \{0, 1\}^n$

- Recall that the Oracle query operation flips the phase of the state's normalized amplitude: $\alpha_x \rightarrow -\alpha_x$.
- The first application of the Oracle gives $\mu = \frac{1}{N}(\frac{N-1}{\sqrt{N}} + \frac{1}{\sqrt{N}}) \approx \frac{1}{\sqrt{N}}$

Grover's Diffusion Gate

Diffusion Gate Mapping

The result of this mapping will thus be:

- For positive amplitudes:

$$\frac{1}{N} \rightarrow (\frac{2}{\sqrt{N}} - \frac{1}{\sqrt{N}}) = \frac{1}{\sqrt{N}}$$

- For the flipped amplitude:

$$-\frac{1}{N} \rightarrow (\frac{2}{\sqrt{N}} + \frac{1}{\sqrt{N}}) = \frac{3}{\sqrt{N}}$$

The amplitude of the marked (flipped) state increases! Grover's algorithm requires $\sqrt{N}$ operations, before the marked state reaches a probability amplitude such that the correct state will *almost certainly* be measured.

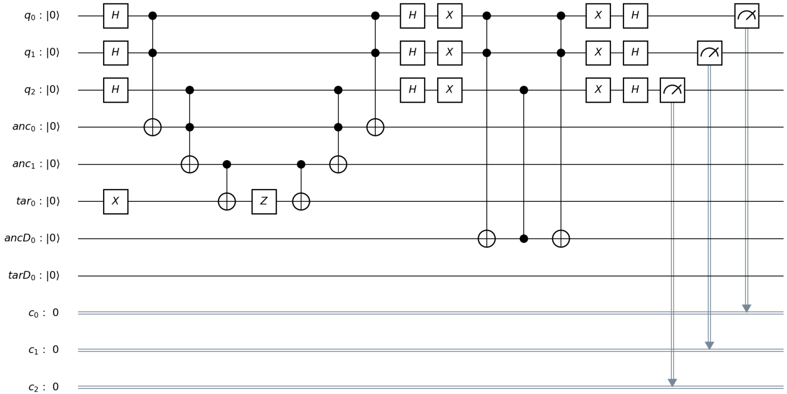
Compilation Time and Problem Size

Problem Size

- If items in the function space need n bits to be encoded, Grover's Algorithm requires approximately $2n$ qubits.
- Depending on the problem at hand, an exponential number of quantum gates (in n) are required.

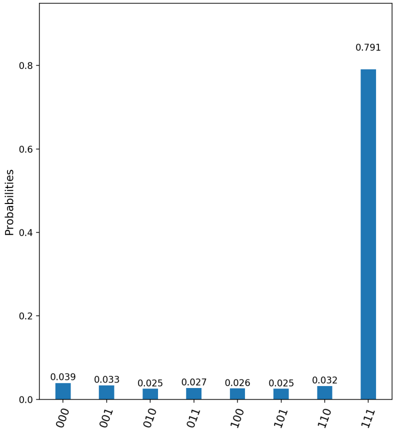
Satisfiability Application

This is the full circuit for the problem considered earlier with the search criterion $f(x == 111)$.



Satisfiability Application

Histogram of Satisfiability measurements



Conclusions

- Quantum hardware is available and readily tested on, however major challenges are still available for Grover's algorithm to be applicable.
- Grover's Algorithm can succesfully solve an unstructured search in a function space, given a constructed oracle, faster than any classical methodology.
- If practical, Grover's Algorithm presents a threat to all Lattice-based encryption schemes.

Further Work

- Investigate the scalability of the Oracle; can it be implemented for simple problems without using an extra amount of qubits and quantum gates?
- Grover's algorithm assumes the oracle is given, so for each new quantum search a new one must be constructed.
    - Can an oracle be built that probabilistically encodes classes of search problems?

Further Reading

Resources

- L. Grover, 'A fast quantum mechanical algorithm for database search,' *arXiv:quant- ph/9605043, 1996.*
- T. Laarhoven, M. Mosca, and J. V. D. Pol, "Solving the Shortest Vector Problem in Lattices Faster Using Quantum Search," *Post-Quantum Cryptography Lecture Notes in Computer Science*, vol. 1, no. 6176, pp. 83–101, Jan. 2013.
- Baritompa, William & W. Bulger, D & Wood, Graham. (2005). Grover's Quantum Algorithm Applied to Global Optimization. SIAM Journal on Optimization. 15. 1170-1184. 10.1137/040605072.

Questions?

Compilation Time and Problem Size

Quote from Craig Gidney from Google's Quantum Computing

The actual obstacle to running Grover's algorithm in practice is that the circuits are large, so you need error correction, but this adds significant overhead. Evaluating a function under superposition is easily a billion times less energy efficient than classically computing the same function, using current techniques. This requires you to go to absolutely enormous problem sizes in order for the quadratic advantage to overcome this starting penalty, and because Grover search cannot be paralelized the result is a quantum circuit that will take on the order of a year or a decade to evaluate. There are not very many problems where people would be willing to wait ten years to make the computation 10x cheaper, in terms of dollars, compared to running in parallel on a hundred thousand cloud computers for a few weeks.

3-SAT Application

3-SAT Problem

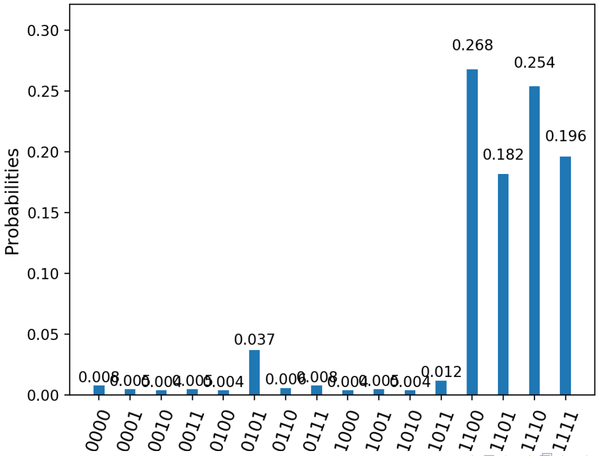
The following is an application for the 3-SAT

$$\begin{gathered}(\neg x1 \wedge \neg x3 \wedge \neg x4)\vee \\ (x2 \wedge x3 \wedge \neg x4)\vee \\ (x1 \wedge \neg x2 \wedge x4)\vee \\ (x1 \wedge x3 \wedge x4)\vee \\ (\neg x1 \wedge x2 \wedge \neg x3)\end{gathered}$$

https://firebasestorage.googleapis.com/v0/b/arduinohandler.appspot.com/o/Photos%2F3SAT.png?alt=media&token=ca052733-38ef-450e-bfd4-024bbb62d7d3

3-SAT Application

Resulting histogram of measured counts


Probabilities