Few Quantum Computation algorithms

Abstract

Some notes on few quantum algorithms.

Index Terms

Quantum Algorithms, Quantum Computation, Qubit

Contents

Images are missing in the PDF- Will fix this soon

I. Basic Elements of Quantum Algorithms

The most basic element of a QC is a quantum bit, qubit for short, which is a two-level quantum system. It spans a two dimensional Hilbert space denoted as H_2. H_2 is equipped with a fixed basis $(|0\rangle, |1\rangle)$, a so-called computational basis. States $|0\rangle$ and $|1\rangle$ are called the basis states. A general state of a single quantum bit is a vector that can be written as:

$$c_0|0\rangle + c_1|1\rangle, \tag{1}$$

where $|c_0|^2 + |c_1|^2 = 1$

We can extend this definition to multiple qubits: for example, a system of two qubits describes a four-dimensional Hilbert space $H_4 = H_2 \otimes H_2$ having orthonormal basis $(|00\rangle, |01\rangle, |10\rangle, |11\rangle)$. A state of a two-qubit system is a unit-length vector

$$c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle, \tag{2}$$

with $|c_0|^2 + |c_1|^2 + |c_2|^2 + |c_3|^2 = 1$.

One of the most important gates in QC is the Hadamard gate, denoted by H, and it is defined as follows:

$$H|\mathbf{x}\rangle \;= \frac{1}{\sqrt{2}} \sum_{\mathbf{y}} (-1)^{\mathbf{x}.\mathbf{y}} |\mathbf{y}\rangle \tag{3}$$

Applying H on the computational basis we get

$$\begin{aligned} |0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \\ |1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \end{aligned} \tag{4}$$

Hadamard gate basically creates superpositions out of pure states, and it can also be written in the matrix form as follows:

$$H = \frac{1}{\sqrt{2}} \left(\begin{array}{cc} 1 & 1 \\ 1 & -1 \end{array} \right). \tag{5}$$

Using Hadamard transformations along with phase shift operations, one can implement quantum Fourier transform (QFT), which is basically the classical discrete Fourier transform applied to the quantum state vector:

email: quarktetra@gmail.com

Find the interactive HTML-document here.

$$QFT|x\rangle = \frac{1}{\sqrt{N}}\sum_{y=1}^{N-1} e^{\frac{-2\pi i x.y}{N}}|y\rangle \tag{6}$$

This transformation is the key element in Shor's factorization algorithm as we will discuss below.

II. Shor's Algorithm

Factoring a large number, N, into its primes is a hard problem. In the 1970s, it was shown that factorization can be mapped into a period finding problem, which is also a hard problem, and there are no known classical algorithms that can do this computation efficiently. However, period finding problem has the obvious structure of periodicity, and quantum computers can make use of this internal feature of the problem to yield exponential speed up over classical algorithms.

Below are the steps of the factorization algorithm:

- You pick up a random number a which is smaller than $\sqrt{N}$.
- Calculate the *period* of a, denoted by r, so that $a^r - 1$ is a multiple of N, i.e. $a^1 = 1 \,\mathrm{Mod}\, N$
  - This means $(a^{r/2}-1)(a^{r/2}+1)$ is a multiple of N.
  - Therefore $a^{r/2} \pm 1$ and N have common dividers.
- Calculate $\mathrm{GCD(N, a^{r/2}} \pm 1)$, GCD being greatest common divider.

Except for the computation of the period r, there are very efficient methods to execute the algorithm above, and the period finding part is exactly where Shor's algorithm is applied.

Here are the steps of Shor's quantum algorithm to compute the period of a number a:

1. Select the smallest integer q satisfying $N^2 < Q < 2N^2$ where $Q = 2^q$,
2. Prepare the input register as a uniform superposition of numbers 0 to $Q-1$:
   - $|s\rangle = \frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1}|x\rangle$.
3. Append the ancillary bit $|f(x)\rangle = |a^x \,\mathrm{Mod}\, N\rangle$ to get the composite state as:
   - $\frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1}|x, f(x)\rangle$.
4. Apply inverse QFT to the input register only (i.e. exclude the ancillary bit):
   - $\frac{1}{Q}\sum_{x=0}^{Q-1}\sum_{y=0}^{Q-1} e^{\frac{2\pi i x y}{Q}}|y, f(x)\rangle$
   - The first thing we note is the periodicity and the range of f. As the index x runs from 0 to $Q-1$, $f(x)$, executing $\mathrm{Mod}\, N$ operation, will run from 0 to $N-1$. So we can reorder the summation over x with respect to the output $f(x)$, which we will name as z for simplicity. So $\sum_{x=0}^{Q-1} = \sum_{z=0}^{N-1}\sum_{x\in\{0,1,\cdots,Q-1\};f(x)=z}$
   - x ranges from 0 to $Q-1$, and let's mark the smallest value of x that satisfies the relation $f(x) = z$ as x_0. Due to the periodicity of f, the total number of instances of x that will satisfy $f(x) = z$ is $\lfloor \frac{Q-x_0-1}{r} + 1\rfloor$. Let's label these x values with a dummy index b. Essentially we are mapping x to $x_0 + rb$ to write the summation as:
   - $\sum_{x\in\{0,1,\cdots,Q-1\};f(x)=z} e^{\frac{2\pi i x y}{Q}} = \sum_{b=0}^{\lfloor\frac{Q-x_0-1}{r}\rfloor} e^{\frac{2\pi i y(x_0+rb)}{Q}} = e^{\frac{2\pi i x_0 y}{Q}}\sum_{b=0}^{\lfloor\frac{Q-x_0-1}{r}\rfloor} e^{\frac{2\pi i r b y}{Q}}$
5. Make a measurement on the ancillary bit. This will result in an integer z. The input register state will collapse into a superposition in the subspace of x values that satisfies $f(x) = z$, which is what we have calculated above.
   - This is a superposition of many states, which will cause interference. If the phase factors $e^{\frac{2\pi i r b y}{Q}}$ align, it will be a constructive interference. For the phase factors to be aligned as they summed over with the index b, it is required to have $e^{\frac{2\pi i r y}{Q}}$ to be close to the real axis, i.e. $\frac{ry}{Q}$ needs to be close to some integer c. When we make a measurement on the input register state, due to the constructive interference, we will most probably measure a value of y such that $\frac{ry}{Q}$ will be close to an integer.

6. Perform classical continued fraction expansion:
   - So we have the measured value of y, and we know the value of Q since we set it at the beginning of the algorithm. Therefore, we know the value $\frac{y}{Q}$. We also know that $\frac{yr}{Q}$ needs to be close an

integer c, which implies that $\frac{y}{Q}$ is close to $\frac{c}{r}$. What we need to do is to express $\frac{y}{Q}$ as a fraction $\frac{d}{s}$ with the conditions $s < N$ and $\left|\frac{y}{Q} - \frac{d}{s} < \frac{1}{2Q}\right|$. This computation can be executed efficiently by classical algorithms.

7. The value of s is very likely to be the period r we are looking for, and we can verify this quickly by computing if $a^s = 1 \,\mathrm{Mod}\, N$. If so, we have successfully computed the period, otherwise we try other candidates $\frac{d}{s}$ around $\frac{y}{Q}$. If none of them works, we go back to step 1 and start over.

III. Grover's Algorithm

Grover's search algorithm enables QC to find a specific item in an unsorted database of N entries using $\mathcal{O}(\sqrt{N})$ operations whereas a classical algorithm would require $\mathcal{O}(N)$ operations.

Consider a database with N entries, one of which is the target entry. The goal is to find the index of that particular entry with the least number of queries. The database can be treated as a black box, which is usually referred to as an *oracle*, that calculates a simple function:

$$f(x) = \begin{cases} 1 & x = w \\ 0 & x \neq w, \end{cases} \tag{7}$$

where w is the entry we are trying to locate. We are going to feed in a state $|x\rangle|q\rangle$ in to the oracle where x represents the index we are querying and q is an ancillary bit which will be used by the oracle to return the query result. If we hit at the index of the special entry, i.e. $x = w$, the oracle will flip the ancillary bit, otherwise it will return the same value of it1. So mathematically, the oracle is doing is the following transformation:

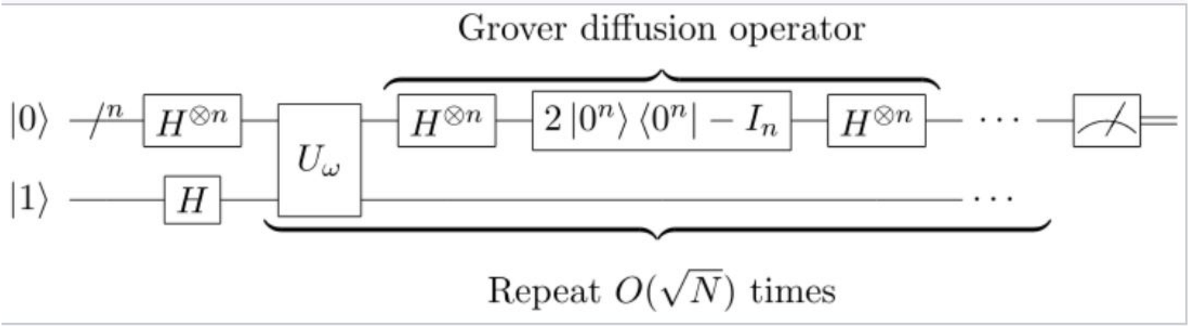
$$|x\rangle|q\rangle \rightarrow |x\rangle|f(x) \oplus q\rangle. \tag{8}$$

Here are the steps of the algorithm:

1. Prepare a uniform superposition of numbers 0 to $N-1$: $|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$,
2. Append the ancillary bit $|q\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$,
3. Feed this input to the oracle,
4. Apply the Grover diffusion operator $2|s\rangle\langle s| - I$,
5. Return to Step 3 and repeat $\sqrt{N}$ times,
6. Measure the output.

The algorithm steps are illustrated below:
\begin{figure}[H]

{



}

\caption{Grover's search algorithm, Wikipedia.} \end{figure}

Note that we are feeding in all of the possible indices at once, so the special index w is indeed fed into the oracle. However it is just one of the N states appearing in the input. With no clever algorithm, the output will also be in a superposition of N states. It will collapse into one of them when a measurement is done. The probability of this state being the correct one is just $1/N$, just like the classical one. QC's ability to process all inputs at once is not useful unless you can sift through the output using a good algorithm. To understand how Grover's algorithm enhances the chances of getting the correct output, let's take a look at what the oracle returns for that particular input: $x = w$ and $|q\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. Using Eq. (8), we get:

$$|w\rangle|q\rangle = |w\rangle\frac{|0\rangle - |1\rangle)}{\sqrt{2}} \to |w\rangle\frac{|1 \oplus 0\rangle - |1 \oplus 1\rangle}{\sqrt{2}} = |w\rangle\frac{|1\rangle - |0\rangle}{\sqrt{2}} = -|w\rangle|q\rangle, \tag{9}$$

which is simply the negative of the input value. One can repeat the same calculation and show that when $x \neq w$, the oracle output is equal to its input. Based on these two observations, we can define the effect of the oracle in the operator form:

$$U_w = I - |w\rangle\langle w|, \tag{10}$$

which is easy to understand: if the object it operates on has $|w\rangle$ content, then the sign on that component will be flipped. Let's apply the oracle operation U_w on the uniform superposition $|s\rangle$:

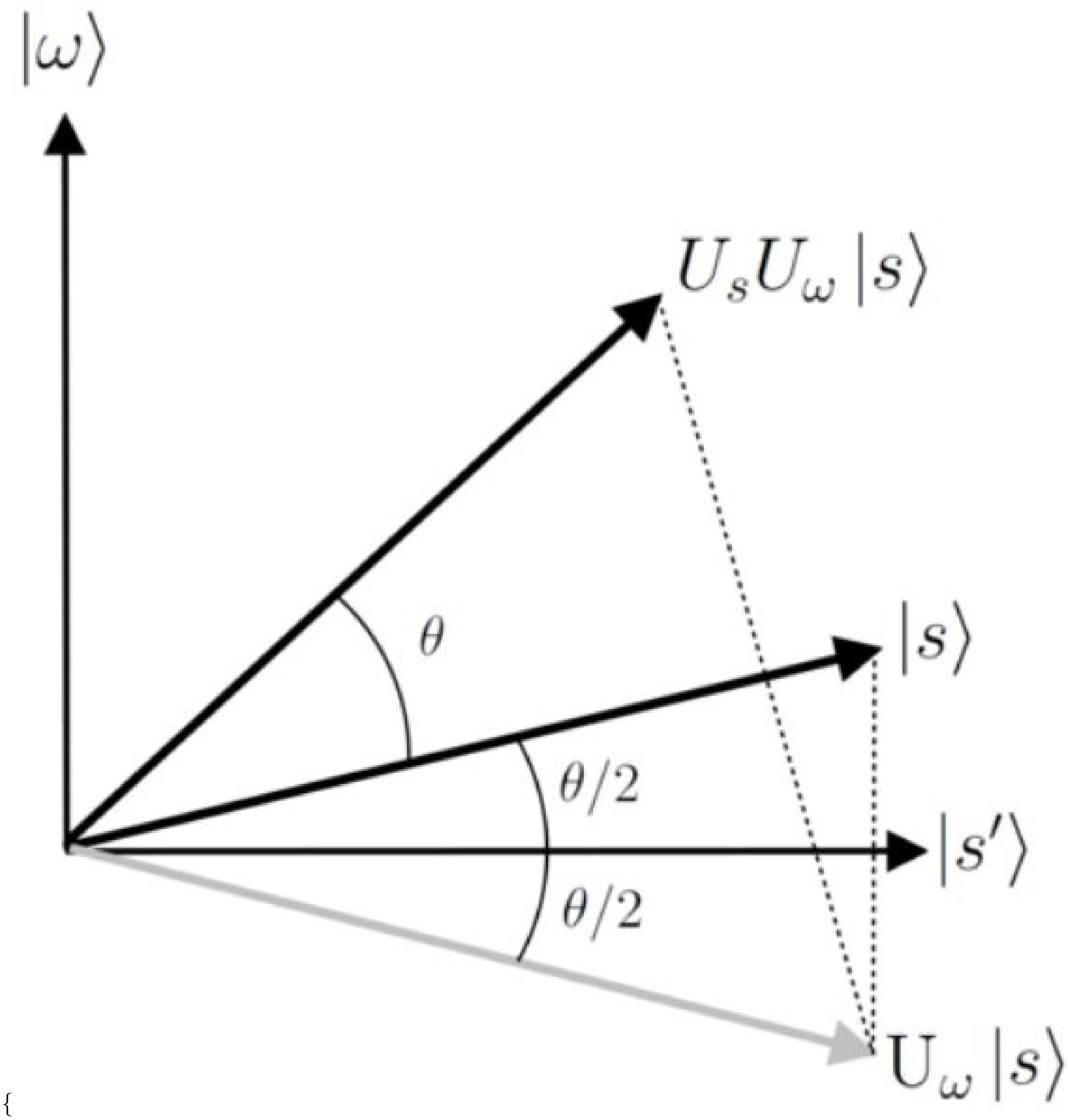
$$U_w|s\rangle \quad = (I - |w\rangle\langle w|)|s\rangle = |s\rangle - \tfrac{2}{\sqrt{N}}|w\rangle. \tag{11}$$

The diffusion operation, $U_s = 2|s\rangle\langle s| - I$ will act on the state in Eq. (11) to yield

$$U_sU_w|s\rangle \quad = \quad (2|s\rangle\langle s| - I)(|s\rangle - \frac{2}{\sqrt{N}}|w\rangle) = \frac{N-4}{N}|s\rangle + \frac{2}{\sqrt{N}}|w\rangle, \tag{12}$$

which shows the ingenuity of the algorithm: the amplitude of the state $|w\rangle$ increased from $1/\sqrt{N}$ to $\frac{2}{\sqrt{N}}$ in one iteration. In fact, if $N = 4$, the amplitude becomes 1, which means that Grover's algorithm can locate the special entry out of 4 in a single iteration with 100% probability. For larger N, you need to keep iterating U_sU_w operations $\sqrt{N}$ times to enhance the amplitude of $|w\rangle$. At each step of the iteration, you are moving the output state closer to the state represented by $|w\rangle$. The operations can be visualized as rotations in Hilbert space of quantum states as illustrated below.

\begin{figure}[H]



{

}

\caption{Geometric interpretation of Grover's algorithm, Wikipedia.} \end{figure}

[1] You may ask why we simply do not return 1 or 0 depending on $x = w$. This is because all the operations in QC have to be reversible. If the oracle over-wrote $|q\rangle$ with 0 or 1, the previous information on $|q\rangle$ would be not recoverable. This is also the reason why there is no quantum AND gate: one cannot uniquely recover inputs of AND gate from its output.