$|\text{VI WECIQ}\rangle$ – Alfenas – 2022

Walking through Quantum Algorithms

Renato Portugal (LNCC)

Outline

- Begin with Deutsch
- Walking through the basic quantum algorithms
- Element distinctness
- HHL
- Hybrid classical-quantum algorithms

Basic quantum algorithms: The beginning

The area of quantum algorithms started in 1985 with the quantum Turing machine assembled by Deutsch [1]:

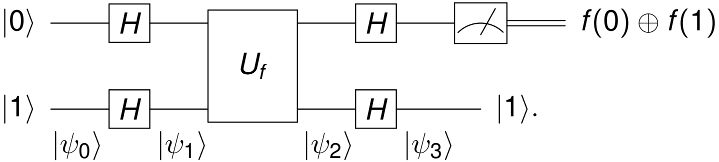
- Quantum states as input.
- Unitary operator driving the calculation.
- Measurement at the end.

In [1], Deutsch showed how quantum parallel computation would work: It is possible to determine $f(0) \oplus f(1)$ by calculating $f(0)$ and $f(1)$ at the same time without parallel physical resources.

[1] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. Proceedings of the Royal Society of London A 400, pp. 97-117, 1985.

A quantum leap 1985→1989: The circuit model [1]

Circuit of Deutsch's algorithm:



U_f is used only once, where

$$U_f|x\rangle|0\rangle = |x\rangle|f(x)\rangle,$$

The last state is

$$|\psi_3\rangle \quad = \quad \pm|f(0) \oplus f(1)\rangle \otimes |1\rangle.$$

[1] D. Deutsch. Quantum computational networks. Proc. Royal Society London A 425, pp. 73–90, 1989.

Basic quantum algorithms [1]

- Deutsch-Jozsa Algorithm – 1992
- Bernstein-Vazirani Algorithm – 1993
- Simon's Problem – 1994
- Shor's Algorithm for Factoring Integers – 1994
- Shor's Algorithm for Discrete Logarithm – 1994
- Phase Estimation Algorithm (Kitaev) – 1995
- Grover's Algorithm – 1996

[1] R. Portugal. Basic quantum algorithms. ArXiv:2201.10574. (108 pages)

Using black-boxes and querying oracles

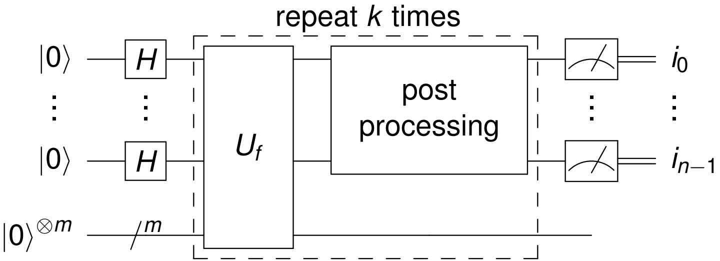
Definition

An **oracle** is a n-bit input to m-bit output function $f : \{0,1\}^n \longrightarrow \{0,1\}^m$ with some hidden property.

Algorithm	m	*Oracle*
Deutsch-Jozsa	1	f is balanced or constant
Bernstein-Vazirani	1	f is linear: $f(x) = s_1x_1 + \cdots s_nx_n$
Simon's Problem	$m > 1$	$f(x) = f(y) \leftrightarrow x \oplus y \in \{0, s\}$
Shor for Factoring	$m > 1$	f is periodic
Grover	1	$f(x) = 0$ except if $x = x_0$

Structure of the basic quantum algorithms

Algorithm	m	k	post-proc
Deutsch-Jozsa	1	1	$H^{\otimes n}$
Bernstein-Vazirani	1	1	$H^{\otimes n}$
Simon's Problem	n	1	$H^{\otimes n}$
Shor for Factoring Integers	$\approx 2n$	1	$F^{\dagger}_{2^{2n}}$
Grover	1	$\sqrt{2^n}$	$2\|u\rangle\langle u\| - I$

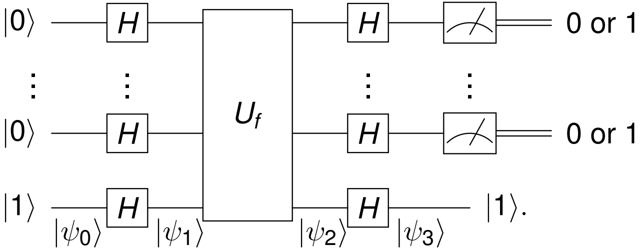


$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle.$$

Deutsch-Jozsa algorithm

Algorithm 1: Deutsch-Jozsa algorithm

Prepare the initial state $|0\rangle^{\otimes n}|1\rangle$;
Apply $H^{\otimes(n+1)}$;
Apply U_f;
Apply $H^{\otimes(n+1)}$;
Measure the first register in the computational basis.



Deutsch-Jozsa algorithm

Final state:

$$|\psi_3\rangle = \frac{1}{2^n}\left(\sum_{x=0}^{2^n-1}(-1)^{f(x)}\right)|0\rangle^{\otimes n}|1\rangle + \cdots$$

f is constant if and only if the amplitude of the 1st term is 0.

Bernstein-Vazirani algorithm

The circuit is the same as Deutsch-Jozsa algorithm.

U_f changes because f is

$$f(x) = s \cdot x = s_0 x_0 + ... + s_{n-1} x_{n-1} \mod 2.$$

The final state is

$$|\psi_3\rangle = |s\rangle \otimes |1\rangle.$$

Bernstein-Vazirani algorithm has no entanglement.

Simon's problem

Let $f : \{0,1\}^n \longrightarrow \{0,1\}^n$ be a n-bit input to n-bit output function such that

$$f(x) = f(y) \leftrightarrow x \oplus y \in \{0, s\}$$

for all $x, y \in \{0,1\}^n$.

Classical algorithm: $\Omega\left(\sqrt{2^n}\right)$

Simon's algorithm: $O\left(n^2\right)$

Simon's problem

Algorithm 2: Simon's algorithm

Input: Function $f : \{0,1\}^n \longrightarrow \{0,1\}^n$ with the promise that
$f(x) = f(y) \leftrightarrow x \oplus y \in \{0, s\}$.
Output: s with probability greater than 1/2.

Run the quantum part $n-1$ times, assume outputs $x^{(1)}, \ldots, x^{(n-1)})$;
Solve the system of linear equations $\{x^{(1)} \cdot s \equiv 0, \ldots, x^{(n-1)} \cdot s \equiv 0\} \mod 2$ (assume solution for $s_0, \ldots, s_{n-2}$);
Take $s_{n-1} = 0$ and check whether $f(s) = f(0)$;
If True return $s_0 \ldots s_{n-2}0$; otherwise, return $s_0 \ldots s_{n-2}1$.

Algorithm 3: Quantum part of Simon's algorithm

Input: A black box U_f implementing function $f : \{0,1\}^n \longrightarrow \{0,1\}^n$ with the promise that $f(x) = f(y) \leftrightarrow x \oplus y \in \{0, s\}$.
Output: Point $x \in \{0,1\}^n$ such that $x \cdot s = 0$ with prob $= 1$.

Prepare the initial state $|0\rangle^{\otimes n}|0\rangle^{\otimes n}$;
Apply $H^{\otimes n}$ to the first register;
Apply U_f;
Measure the second register in the computational basis (assume output $z_0 \ldots z_{n-1}$);
Apply $H^{\otimes n}$ to the first register;
Measure the first register in the computational basis.

Shor's algorithm for factoring integers

Periodic function:

$$f(\ell) = a^{\ell} \mod N$$

The order r of number a mod N is the smallest positive integer r such that

$$a^r = 1 \mod N.$$

For example, take $a = 2$, then

$$a^2 = 4,\ a^3 = 8,\ a^4 = 16,\ a^5 = 11,\ a^6 = 1 \mod 21.$$

Then $r = 6$ (r is the period)

After U_f, the post-processing is the inverse Fourier transform $F^{\dagger}_{2^{2n}}$.

Shor's algorithm for factoring integers

Algorithm 4: Shor's algorithm – Classical part

Input: Composite integer N.
Output: A nontrivial factor of N.

1 If N is even, return 2; otherwise, continue;
2 If N is a power of some prime number p, return p; otherwise, continue;
3 Pick uniformly at random an integer a such that $1 < a < N$;
4 If $\gcd(a, N) > 1$, return $\gcd(a, N)$; otherwise, continue;
5 Run the quantum part with inputs a and N $\left(\text{assume output } \ell_0, \ldots, \ell_{2n-1}\right)$;
6 Calculate $b = \ell / 2^{2n}$ (the same 2^{2n} used in the quantum part);
7 Find the convergent of the continued fraction expansion of b with the largest denominator r' such that $r' < N$;
8 If r' is odd, go to Step 3; otherwise, continue;
9 If $a^{r'/2} + 1 \not\equiv 0 \mod N$, return $\gcd(a^{r'/2} + 1, N)$; otherwise, go to Step 3.

Shor's algorithm for factoring integers

Algorithm 5: Shor's algorithm – Quantum part

Input: A composite integer N and integer $1 < a < N$ such that $\gcd(a, N) = 1$.
Output: $2n$-bit string ℓ that is the nearest integer to a multiple of $2^{2n}/r$ with probability greater than or equal to $4/\pi^2$.

Prepare the initial state $|0\rangle^{\otimes 2n}|0\rangle^{\otimes n}$, where $n = \lceil \log_2 N \rceil$;
Apply $H^{\otimes 2n}$ to the first register;
Apply $U_N^{(a)}$ to both registers;
Measure the second register in the computational basis (assume output $z_0 \ldots z_{n-1}$);
Apply $F_q^{\dagger}$ to the first register;
Measure the first register in the computational basis.

Shor's algorithm for factoring integers

A continued fraction expansion of a positive rational number $b < 1$ is

$$b = \cfrac{1}{b_1 + \cfrac{1}{b_2 + \cfrac{1}{\ddots + \frac{1}{b_Z}}}},$$

the successive convergents are $[b_1]$, $[b_1, b_2]$, $[b_1, b_2, b_3]$, and so on.

For example, $N = 21$ and $a = 2$, for the output $\ell = 85$ the successive convergents of $85/2^9$ are

$$[6] = \frac{1}{6}, \ [6, 42] = \frac{42}{253}, \ [6, 42, 2] = \frac{85}{512} = \frac{\ell}{2^{2n}}.$$

Phase estimation – Kitaev 1995

Given U and $|\psi\rangle$ such that

$$U|\psi\rangle = e^{2\pi i \phi}|\psi\rangle,$$

Find ϕ with m binary digits.

Algorithm 6: Phase estimation algorithm

Input: Eigenvector $|\psi\rangle$ of U.
Output: Number $2^m\phi$, where $\exp(2\pi i\phi)$ is the eigenvalue of $|\psi\rangle$.

1. Prepare the initial state $|0\rangle^{\otimes m} \otimes |\psi\rangle$;
2. Apply $H^{\otimes m}$ to the first register;
3. For ℓ in $[0, m-1]$ apply the controlled operation $C^{m-\ell}\left(U^{2^\ell}\right)$, where the control qubit is $m-\ell$ and the target is the 2nd register;
4. Apply $F_{2^m}^\dagger$ to the first register;
5. Measure the first register in the computational basis.

Phase estimation – Kitaev 1995

Circuit:

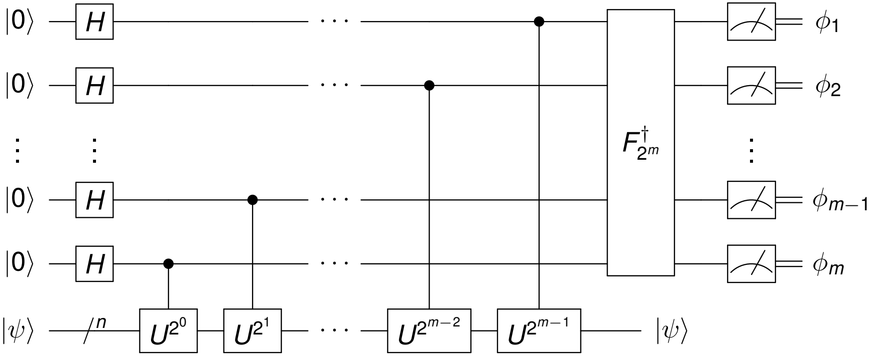


Figure: Full circuit of the phase estimation algorithm.

Next: Grover's algorithm

I prefer to skip, unless there are questions on this algorithm.

Followups

- Generalization of Grover's algorithm – 1998 – BBHT
- Quantum amplitude amplification – 1998 – BBHT
- Quantum counting – BHT – 1998 and 2002 – BHMT

Element Distinctness Problem – 2003

- Consider a list with N elements
- Are all elements distinct?
  - Classically, it requires N queries.
  - In the quantum case, $O(N^{2/3})$ queries.
- k-distinctness problem: are there k colliding elements?

Previous Results

- Aaronson and Shi obtained the lower bound $O(N^{2/3})$ in the query model [1].
- Ambainis described an algorithm with $O(N^{2/3})$ queries [2,3].

[1] S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *J. ACM*, 51(4):595–605, 2004.

[2] A. Ambainis. Quantum walk algorithm for element distinctness. In *FOCS '04: Proc. of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 22–31, Washington, DC, 2004.

[3] A. Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007.

Ambainis' Graph

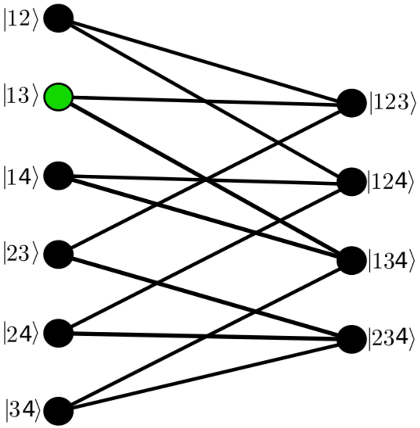
Some definitions:

- N is the number of elements in the list
- $[N]$ is the set $\{1, ..., N\}$
- r is the integer nearest to $N^{\frac{2}{3}}$
- $\mathcal{S}_r$ is the set of all r-subsets of $[N]$

Ambainis' graph: is a bipartite graph with $\binom{N}{r} + \binom{N}{r+1}$ vertices. The vertices of the first set are r-subsets of $[N]$ and of the second set are $(r+1)$-subsets. A vertex v_1 in the first set is adjacent to a vertex v_2 in the second set if and only if $|v_1 \cap v_2| = r$.

Example of Ambainis' Graph

Take $L = [13, 2, 13, 7]$. Then, $N = 4$ and $r = 2$
$\mathcal{S}_2 = \{\{1,2\},\{1,3\},\{1,4\},\{2,3\},\{2,4\},\{3,4\}\}$
$\mathcal{S}_3 = \{\{1,2,3\},\{1,2,4\},\{1,3,4\},\{2,3,4\}\}$



Ambainis' Algorithm

The algorithm has two parts: Main Block and Subroutine

- Main Block: Reapeat $t_1 = O(\sqrt{r})$ times
  - Apply a conditional phase-flip operator
  - Repeat Subroutine $t_2 = \left\lceil \frac{\pi}{3\sqrt{k}}\sqrt{r} \right\rceil$ times
- Subroutine: quantum-walk-part
  - Apply U_1
  - Query
  - Apply U_2
  - Query
- Measure

The success probability is 75% asymptotically.

Solving Linear Systems – HHL [1]

Problem: Given A and $\vec{b}$, find $\vec{x}$ such that

$$A \cdot \vec{x} = \vec{b}.$$

Limitations: To represent $\vec{b}$ and $\vec{x}$ as quantum states we have to rescale:

$$|b\rangle = \frac{\vec{b}}{\|\vec{b}\|}$$

and the solution $|x\rangle$

$$|x\rangle = \frac{A^{-1}|b\rangle}{\|A^{-1}|b\rangle\|}.$$

[1] A W Harrow, A Hassidim, S Lloyd. Quantum algorithm for linear systems of equations. Physical review letters, 103(15):150502, 2009.

Solving Linear Systems – HHL

Restrictions: A must be Hermitian and sparse and state $|b\rangle$ must be prepared initially.

Applications: After preparing $|x\rangle$, it is possible to calculate the expectation value

$$\langle x|\mathcal{O}|x\rangle$$

of an observable $\mathcal{O}$.

Solving Linear Systems – HHL

Let λ_j and $|u_j\rangle$ be the eigenvalues and eigenvectors of A. Then,

$$|b\rangle = \sum_{j=1}^{N} \beta_j |u_j\rangle.$$

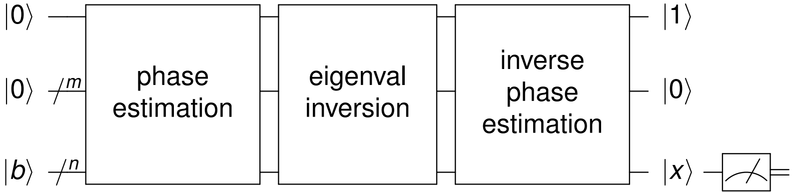
The goal is to obtain

$$|x\rangle = \sum_{j=1}^{N} \frac{\beta_j}{\lambda_j} |u_j\rangle.$$

Solving Linear Systems – HHL

If $A = R^\dagger \Lambda R$, where Λ is diagonal, HHL can be summarized as:

- $R^\dagger \Lambda R|x\rangle = |b\rangle$
- $\Lambda R|x\rangle = R|b\rangle$ (phase estimation with e^{iA})
- $R|x\rangle = \Lambda^{-1} R|b\rangle$ (eigenvalue inversion)
- $|x\rangle = R^\dagger \Lambda^{-1} R|b\rangle$ (phase estimation†)



QAOA e o Problema da Partição na Teoria de Números

- Separe uma lista L de números em duas sub-listas $L1$ e $L2$ tal que
$$\sum_i L1_i = \sum_i L2_i$$
- Exemplo: $L = [2, 1, 1]$. Solução: $L1 = [2]$ e $L2 = [1, 1]$.
- O problema de decisão associado é NP-completo
- O problema de otimização associado é NP-hard
- É considerado o problema NP-completo mais fácil

Problema da Partição como Problema de Otimização

- Seja a seguinte função objetivo:

$$f(\vec{s}) = \left(\sum_i s_i L_i \right)^2,$$

where $s_i \in \{+1, -1\}$. Ache $\min_{\vec{s}} f(\vec{s})$

- Note que

$$f(\vec{s}) = \sum_i s_i L_i \sum_j s_j L_j = \sum_{i,j} s_i s_j L_i L_j,$$

- Exemplo: $L = [2, 1, 1]$

$f(1, 1, 1) = 16,\ f(+1, 1, -1) = 4,\ f(+1, -1, 1) = 4,\ f(+1, -1, -1) = 0,$

$f(-1, 1, 1) = 0,\ f(-1, 1, -1) = 4,\ f(-1, -1, 1) = 4,\ f(-1, -1, -1) = 16$

Soluções: $\vec{s} = (1, -1, -1)$ ou $(-1, 1, 1)$ pois $f(\vec{s}) = 0$.

Otimização Clássica $\longrightarrow$ Otimização Quântica

- Use o Hamiltoniano C (operador Hermitiano ou Observável)
- Os autovalores de C são energias

$$C|\vec{s}\rangle = E_{\vec{s}}|\vec{s}\rangle,$$

  onde $|\vec{s}\rangle$ é um autovetor de C
- Defina C tal que a solução do problema de otimização clássico $\min_{\vec{s}} f(\vec{s})$ seja o menor autovalor de C.

$$C = \begin{bmatrix} f(+1,...,+1) & 0 & & 0 \\ 0 & f(+1,...,-1) & & 0 \\ & & \ddots & \\ 0 & 0 & & f(-1,...,-1) \end{bmatrix}$$

Cada energia é um solução (provavelmente ruim) do problema de otimização.

Como construir C?

- C é muito grande para ser processado no computador clássico
- Como construir C com menos recursos? Resp.: Use um CQ.
- Para o problema da partição, a função objetivo é

$$f(\vec{s}) = \sum_{i,j} s_i s_j L_i L_j,$$

- Seja s_i um autovalor de Z_i associado ao autovetor $|s_i\rangle$. Então

$$C = \sum_{i,j} Z_i Z_j L_i L_j,$$

- Note que

$$C|\vec{s}\rangle = f(\vec{s})|\vec{s}\rangle$$

Conversão spin $\longrightarrow$ qubit

- Definição de Z (x é um bit, $|x\rangle$ é um qubit):

$$Z|x\rangle = (-1)^x |x\rangle$$

- Definição de Z_i:

$$\begin{aligned} Z_i|x_1, ..., x_n\rangle &= I \otimes ... \otimes Z \otimes ... \otimes I|x_1, ..., x_n\rangle \\ &= (-1)^{x_i}|x_1, ..., x_n\rangle \end{aligned}$$

- Então

$$Z_i Z_j|x_1, ..., x_n\rangle = (-1)^{x_i}(-1)^{x_j}|x_1, ..., x_n\rangle$$

- Conversão spin $\longleftrightarrow$ bit:

$$s_i = (-1)^{x_i} \text{ ou } x_i = \frac{1 - s_i}{2}$$

Preparação inicial e o objetivo do QAOA

Comece com o computador quântico na superposição uniforme

$$|\psi_0\rangle = \frac{1}{\sqrt{\| |\psi_0\rangle \|}} (|0...0\rangle + |0...1\rangle + ... + |1...1\rangle)$$

e obtenha o estado final

$$|\psi_f\rangle = ... + 0.9999e^{i\theta}|\text{sol}\rangle + ...$$

onde sol é a solução ideal. Se houver mais do que uma solução ideal, $|\psi_f\rangle$ deve ser uma superposição das soluções ideais.

Importante: Queremos achar $|\psi_f\rangle$ que minimiza a energia média

$$\langle\psi_f|C|\psi_f\rangle$$

no final do algoritmo.

O Algoritmo QAOA

Step 1. Escolha os ângulos γ_1 and β_1 que minimiza a energia média.

Step 2. Inicialize o CQ na superposição uniforme:
$|\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_j |j\rangle$

Step 3. Aplique

$$U(C, \gamma_1) = \exp(-i\gamma_1 C) = \mathrm{diag}_j\{\exp(-i\gamma_1 C_{j,j})\}$$

Step 4. Aplique o mixer $B = \sum_j X_j$:

$$U(B, \beta_1) = \exp(-i\beta_1 B) = \prod_j \exp(-i\beta_1 X_j) = \prod_j R_x^{(j)}(2\beta_1)$$

Step 5. Repita p vezes os passos 3 e 4 escolhendo novos ângulos $\gamma_2, ..., \gamma_p$ and $\beta_2 ... \beta_p$ em cada rodada e faça no final a medição dos qubits na base computacional.

Análise do QAOA

O estado final é

$$|\psi_f\rangle = \exp(-i\beta_p B)\exp(-i\gamma_p C)\ldots\exp(-i\beta_1 B)\exp(-i\gamma_1 C)\,|\psi_0\rangle$$

onde

$$B = \sum_i X_i.$$

Após a medição quando o estado é $|\psi_f\rangle$, obtemos uma cadeia de n bits, que é a solução-candidato.

Devemos analisar a qualidade do resultado usando

$$F(\vec{\beta},\vec{\gamma}) = \left\langle \psi_f(\vec{\beta},\vec{\gamma}) \middle| C \middle| \psi_f(\vec{\beta},\vec{\gamma}) \right\rangle$$

Final comments

- Three decades of quantum algorithms
- Just the beginning of the area
- But the end of this talk
- Hope you have enjoyed the walk

Thank you

Questions?