QUANTUM ORACLES - COMO TRANSFORMAR PROBLEMAS CLÁSSICOS EM QUÂNTICOS

Alexandre Silva
Ciências da Computação
UNIVEM - Centro Universitário Eurípides de Marília

ABSTRACT

1 Introdução

Hoje, não é difícil ver alguém falando sobre computação quântica e como essas máquinas vão mudar o futuro. Contudo, muitas dessas frases acabam se levando por extrapolações e/ou usos indevidos de ficção. Neste artigo, mostrarei que nem tudo é possível ser feito com um computador quântico atual, assim como existem pequenas áreas que se beneficiam ao máximo dessa nova tecnologia.
Para esse feito, serão mostrado alguns testes feitos usando o qiskit, um framework open source da IBM para computação quântica, além de alguns resultados obtidos após executar os algoritmos em simuladores e máquinas reais, assim como seus relativos em computação clássica. Algoritmos dos quais tomam proveito dos quantum oracles, modelos ideias de função que não ajudam a descrever o algoritmo matematicamente, também tomam proveito de alguns efeitos quânticos, como superposição e interferência, para se sobressair à algumas estratégias clássicas.
Com isso, o projeto foi desenvolvido em cima de cinco pequenos problemas, sendo eles: conversão de milhas para quilômetros, torre de Hanoi, explorador de arquivos, Buckshot Roulette e QRAM. Todas as implementações e materiais utilizados podem ser encontrados nesse repositório do GitHub.

2 Início do projeto

Para dar início a pesquisa, foi necessário entender quais os tipos de oracles existem e como eles podem ser usados.
Em computação clássica, temos as Oracle Machines, as quais são maquinas de Turing, das quais implementam alguma função em seu interior, e ao ser chamado/invocado o resultado correto é retornado em tempo constante $O(1)$, podendo ser vista como uma caixa preta, abstraindo completamente o seu funcionamento. Devido a essa definição, as OMs são ideias matemáticos, sendo assim usados apenas para formalismo matemático.
Contudo em computação quântica, podemos de fato implementar certos modelos de Oracles e adiciona-los a um circuito maior, executando certas funções como: encoding de dados, aplicação de $f(x)$, abstração de partes do circuito, etc.

2.1 Tipos de Oracles

2.1.1 Phase Oracle

Um dos primeiros tipos de Oracles usados para a criação de algoritmos como os de: Grover e Deutsch–Jozsa; é comummente conhecido como *Phase Oracle*.
Tal dispositivo, é usado para atribuir uma fase ao circuito, sendo muito usado para configurar valores, explorar a interferência ou se aproveitar de outros efeitos como o *Phase Kickback*. Matematicamente poderiamos descrever ele da seguinte forma: $|x\rangle |-\rangle \rightarrow (-1)^{f(x)} |x\rangle |-\rangle$, do qual $|x\rangle$ é a entrada do oracle e $|-\rangle$ é a ancilla que prove a fase.

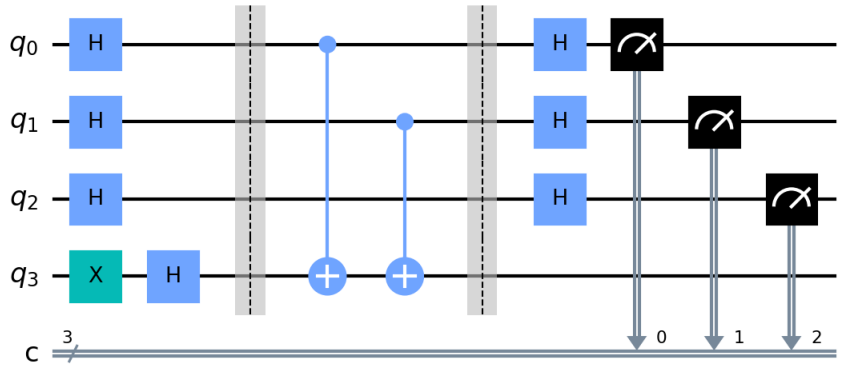


Figura 1: Exemplo de phase oracle usado para o algoritmo de Deutsch–Jozsa

No exemplo acima, utilizamos o *Phase Kickback* para adicionar uma fase nos qubits 0 e 1, transformando seus estados de $|+\rangle$ para $|-\rangle$, fazendo com que ao serem colapsados o resultado $|1\rangle$ apareça na saída.

É possível também criar um phase oracle removendo o qubit adicional (nesse exemplo o Q3), uma vez que podemos utilizar outros gates para introduzir a fase e manter ainda natureza unitária.

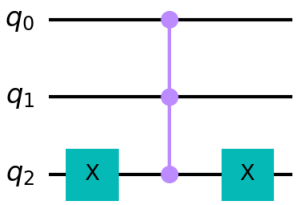
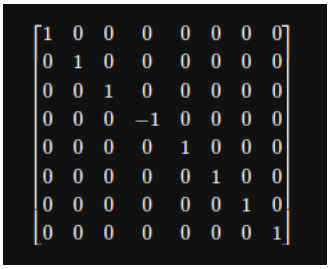


Figura 2: Exemplo fase Oracle sem a Ancilla

Dessa vez, utilizamos o MCP gate para adicionar uma fase global π e dois gates X para dizer quais qubits queremos q tenham o valor 0, codificando assim o valor 011 ou 3 na base decimal.



$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 3: Matriz unitária do Phase oracle

É possível verificar então que ao criarmos esse circuito, matemos a matriz identidade e adicionamos a fase -1 no valor da coluna relativa ao 011.
Essa versão pode ser considerada como um minimal oracle, uma vez que a própria função interna se mantém unitária, sem a necessidade de ancilla.

2.2 Boolean Oracle

O Boolean oracle, por sua vez, representa uma função booleana, sem qualquer adição de fases.
Nesse caso, $|x\rangle$ representa a entrada do oracle e $|y\rangle$ represetam os qubits auxiliares que receberam a resposta, $|x\rangle\,|y\rangle \rightarrow |x\rangle\,|y \oplus f(x)\rangle$.

2.3 Minimal oracle

Como já citado anteriormente, o minimal oracle possui uma função que em sua essência é unitária, não requerendo qubits adicionais $|x\rangle \rightarrow |f(x)\rangle$.

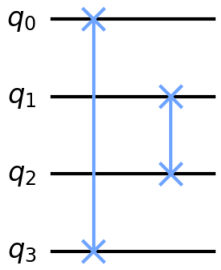


Figura 4: Exemplo de minimal oracle

Lembrando que este pode também adicionar fases ao circuito.

2.4 Simon's Oracle

O Oracle de Simon, é uma instância do Boolean Oracle. Neste visamos encontrar os períodos da função implementada, ou seja, dado x e $f(x) = y$ existe ao menos uma função em que $f(y) = x$?

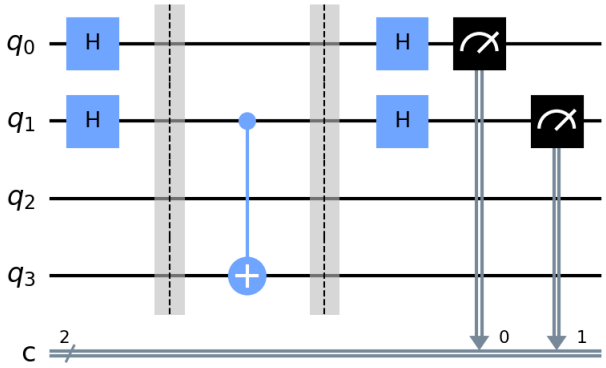


Figura 5: Exemplo do algoritmo de Simon

Nesse algoritmo, configuramos uma chave s dentro do oracle, e ao executar o algoritmo temos os possíveis períodos da função, sendo necessário rotinas de pós processamento para identificar o valor correto.

2.5 QFT(Quantum Fourier Transformation) Oracle

Por fim, o Oracle QFT aplica a versão quântica da transformada de Fourier, projetando os valores de entrada na base X (também conhecido como base de Fourier).

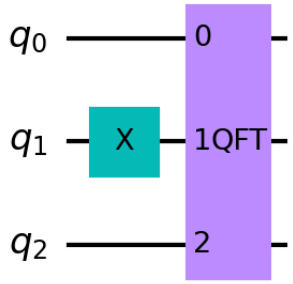


Figura 6: Exemplo do algoritmo de QFT

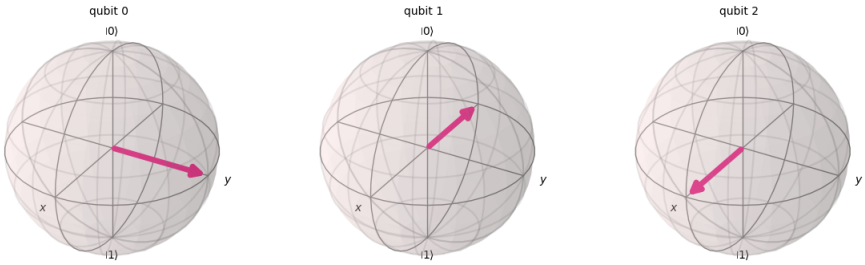


Figura 7: Valores mapeados na base de Fourier

3 Desenvolvimento

3.1 File Explorer

3.2 Miles to Kilometers

3.3 Hanoi Tower

3.4 Buckshot Roulette

3.5 QRAM

Referências

[1] Robert I. Soare. Turing oracle machines, online computing, and three displacements in computability theory. *Annals of Pure and Applied Logic*, 160(3):368–399, 2009. Computation and Logic in the Real World: CiE 2007.

[2] Ryan O'Donnell. Lecture 5: Quantum query complexity, 09 2015.

[3] Dave Bacon. Cse 599d -quantum computing simon's algorithm, 2006.

[4] Robin Kothari. An optimal quantum algorithm for the oracle identification problem. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2014.

[5] Ryan O'Donnell. Lecture 13: Lower bounds using the adversary method, 10 2015.

[6] Laurel Brodkorb and Rachel Epstein. The entscheidungsproblem and alan turing, 12 2019.

[7] Sadika Amreen and Reazul Hoque. Oracle turing machines.

[8] Subrahmanyam Kalyanasyndaram. mod04lec23 - oracle turing machines, 09 2021.

[9] Martin Davis. Turing reducibility?, 11 2006.

[10] Mahesh Viswanathan. Reductions 1.1 introduction reductions, 2013.

[11] What does it mean to be turing reducible?, 03 2016.

[12] Yale Fan. A generalization of the deutsch-jozsa algorithm to multi-valued quantum logic. In *37th International Symposium on Multiple-Valued Logic (ISMVL'07)*. IEEE, May 2007.

[13] Takashi Yamakawa and Mark Zhandry. Classical vs quantum random oracles. Cryptology ePrint Archive, Paper 2020/1270, 2020. `https://eprint.iacr.org/2020/1270`.

[14] Harry Buhrman, Richard Cleve, and Avi Wigderson. Quantum vs. classical communication and computation, 1998.

[15] Javier Sanchez-Rivero, Daniel Talaván, Jose Garcia-Alonso, Antonio Ruiz-Cortés, and Juan Manuel Murillo. Some initial guidelines for building reusable quantum oracles, 2023.

[16] Austin Gilliam, Marco Pistoia, and Constantin Gonciulea. Canonical construction of quantum oracles, 2020.

[17] Elham Kashefi, Adrian Kent, Vlatko Vedral, and Konrad Banaszek. Comparison of quantum oracles. *Physical Review A*, 65(5), May 2002.

[18] Niklas Johansson and Jan-Åke Larsson. Quantum simulation logic, oracles, and the quantum advantage. *Entropy*, 21(8), 2019.

[19] William Zeng and Jamie Vicary. Abstract structure of unitary oracles for quantum algorithms. *Electronic Proceedings in Theoretical Computer Science*, 172:270–284, December 2014.

[20] Alp Atici. Comparative computational strength of quantum oracles, 2004.

[21] Kathiresan Sundarappan. How to build oracles for quantum algorithms, 04 2022.

[22] Zhifei Dai, Robin Choudhury, Jinming Gao, Andrei Iagaru, Alexander V Kabanov, Twan Lammers, and Richard J. Price. View of the role of quantum algorithms in the solution of important problems.

[23] Don Ross. Game Theory. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2024 edition, 2024.

[24] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Physical Review Letters*, 100(16), April 2008.

[25] Samuel Jaques and Arthur G. Rattew. Qram: A survey and critique, 2023.

[26] Tomasz Zawadzki and Piotr Kotara. A python tool for symbolic analysis of quantum games in ewl protocol with ibm q integration. `https://github.com/tomekzaw/ewl`.

[27] Piotr Frackiewicz. Application of the ewl protocol to decision problems with imperfect recall, 2011.

[28] Jens Eisert, Martin Wilkens, and Maciej Lewenstein. Quantum games and quantum strategies. *Physical Review Letters*, 83(15):3077–3080, October 1999.

[29] Muhammad Usman. Kilometres to miles conversion — approximation of fibonacci series, 09 2019.

[30] Lídia André. Tower of hanoi – lídia andré, 03 2021.

[31] diptokarmakar47. How to solve the tower of hanoi problem - an illustrated algorithm guide, 01 2019.

[32] Towers of hanoi: A complete recursive visualization, 05 2020.

[33] GeeksforGeeks. Program for tower of hanoi, 05 2014.

[34] Faisal Shah Khan and Ning Bao. Quantum prisoner's dilemma and high frequency trading on the quantum cloud. *Frontiers in Artificial Intelligence*, 4, 11 2021.

[35] Alexis R. Legón and Ernesto Medina. Dilemma breaking in quantum games by joint probabilities approach. *Scientific Reports*, 12, 08 2022.

[36] Brian Siegelwax. Quantum memory: Qram. what is it and why do we need it? making quantum algorithms thrive., 01 2022.

[37] Gabriel Landi. Density matrices and composite systems.

[38] V. Vijayakrishnan and S. Balakrishnan. Role of two-qubit entangling operators in the modified eisert–wilkens–lewenstein approach of quantization. *Quantum Information Processing*, 18, 03 2019.

[39] Real Python. Scientific python: Using scipy for optimization – real python.

[40] scipy optimize minimize scalar - scipy v1.12.0 manual.
[41] Matt Davis. Optimization (scipy.optimize) — scipy v0.19.0 reference guide.
[42] scipy.optimize.minimize — scipy v1.6.0 reference guide.