UNIVERSIDADE FEDERAL FLUMINENSE

JOÃO MIGUEL BARBOSA BRANDÃO

ALGORITMO QUÂNTICO PARA ENCONTRAR A MODA ESTATÍSTICA

NITERÓI

2018

UNIVERSIDADE FEDERAL FLUMINENSE

JOÃO MIGUEL BARBOSA BRANDÃO

ALGORITMO QUÂNTICO PARA ENCONTRAR A MODA ESTATÍSTICA

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Algoritmos e Otimização

Orientador:
LUIS ANTONIO BRASIL KOWADA

Co-orientador:
FRANKLIN DE LIMA MARQUEZINO

NITERÓI

2018

Ficha catalográfica automática - SDC/BEE

B817a Brandão, João Miguel Barbosa
Algoritmo Quântico para Encontrar a Moda Estatística / João Miguel Barbosa Brandão ; Luis Antonio Brasil Kowada, orientador ; Franklin de Lima Marquezino, coorientador. Niterói, 2018.
64 f. : il.

Dissertação (mestrado)-Universidade Federal Fluminense, Niterói, 2018.

1. Computação quântica . 2. Produção intelectual. I. Título II. Kowada,Luis Antonio Brasil, orientador. III. Marquezino, Franklin de Lima, coorientador. IV. Universidade Federal Fluminense. Escola de Engenharia.

CDD -

Bibliotecária responsável: Fabiana Menezes Santos da Silva - CRB7/5274

João Miguel Barbosa Brandão

Algoritmo Quântico para Encontrar a Moda Estatística

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Algoritmos e Otimização

Aprovada em março de 2018.

BANCA EXAMINADORA

Prof. Dr. Luis Antonio Brasil Kowada - Orientador, UFF

Prof. Dr. Franklin de Lima Marquezino - Co-orientador, UFRJ

Profª. Drª. Karina Mochetti de Magalhães, UFF

Prof. Dr. Renato Portugal, LNCC

Prof. Dr. Uéverton dos Santos Souza, UFF

Niterói

2018

Totus Tuus

Resumo

O *valor modal* —ou simplesmente *moda*— é uma importante informação estatística, definido como um elemento de maior recorrência em um conjunto de dados. O método clássico conhecido mais eficiente para encontrar o valor modal consiste na ordenação do conjunto de entrada seguido pela busca linear da maior sequência de elementos repetidos, o que resulta em um algoritmo com a mesma ordem de complexidade que o método de ordenação utilizado. Kara e posteriormente Coffey e Prezkuta propuseram algoritmos quânticos para encontrar a moda numa lista de inteiros. As complexidades destes algoritmos dependem do maior valor possível na lista. E, em princípio, são melhores do que o clássico quando este valor é pequeno. Neste trabalho propomos um algoritmo quântico probabilístico para encontrar a moda. Este algoritmo calcula a *faixa de dados*, definida como a diferença entre o maior e o menor número do conjunto de entrada, e executa uma entre duas abordagens desenvolvidas para encontrar a moda. Nos casos em que a faixa de dados é grande, é executada uma abordagem baseada no algoritmo quântico para *k-Distinctness*. Caso contrário, se executa uma abordagem baseada no algoritmo quântico de contagem. Por fim, foram feitas simulações desta segunda abordagem. Estas simulações obtiveram uma boa taxa de sucesso, e uma breve análise dos resultados sugere uma correlação inversamente proporcional entre a homogeneidade da recorrência dos valores de entrada e a eficiência do algoritmo proposto.

Palavras-chave: computação quântica, algoritmo quântico, estatística, valor modal.

Abstract

The *modal value* —or simply *mode*— is an important statistical information, defined as an element that appears most often in a dataset. The best known classical method for finding the modal value relies on sorting the input dataset and linearly search for the largest sequence of repeated elements, which result in an algorithm with the same worst-case complexity of sorting it. Kara and posteriorly Coffey and Prezkuta proposed quantum algorithms for finding a mode in a integer list. These algorithms complexities depend on the largest possible value in the list. *A priori*, they are better than the classical approach when this value is small. In this thesis, we propose a quantum algorithm for finding the modal value. This algorithm determines the *data range*, defined as the difference between the largest and the smallest number in the input dataset, and then it computes one of two developed approaches for finding the modal value. In cases in which the data range is large, an approach based on the quantum algorithm for *k-Distinctness* is executed. Otherwise, the approach based on the quantum counting algorithm is executed. Finally, simulations of the second approach were made. These simulations showed a good sucess rate, and a brief analisys of the results indicate a proportionally inverse correlation between the homogeneity of the input data recurrence and the proposed algorithm efficiency.

Keywords: quantum computing, quantum algorithm, statistics, modal value.

Lista de Figuras

1.1 Circuito U-controlado 8

1.2 Condições de controle 9

1.3 Circuito Quântico exemplificando o operador Identidade 10

2.1 Circuito porta U_g 17

2.2 $|\psi\rangle$ em um espaço de Hulbert bidimensional, sendo rotacionado por θ a cada aplicação da iteração de Grover 18

2.3 Circuito da Transformada de Fourier Quântica 23

2.4 Circuito do procedimento de Estimativa de Fase 24

4.1 Porcentagem de acertos do Algoritmo Final 39

4.2 Diferença relativa média dos resultados do primeiro grupo de simulações . 39

4.3 Simulação com variação de t 40

Lista de Siglas e Símbolos

TFQ	:	Transformada de Fourier Quântica;
$\mathcal{H}$	:	Espaço de Hilbert;
$\otimes$	:	Operador de produto tensorial;
$\circ$	:	Sinal de operação 0-controlada;
$\bullet$	:	Sinal de operação 1-controlada;
CP	:	Algoritmo proposto por Coffey e Prezkuta[14];
$KLPF$	:	Algoritmo proposto por Kowada et al.[26];
$BBHT$	:	Algoritmo proposto por Coffey e Prezkuta[14];
G	:	Operador referente à iteração de Grover
U_g	:	Operador de Oráculo

Sumário

1 Introdução **1**

1.1 Descrição do Problema 3

1.2 Breve Introdução à Mecânica Quântica 5

1.2.1 Espaço de Hilbert 5

1.2.2 Bit Quântico 6

1.2.3 Circuitos Quânticos 8

1.2.4 Postulados da Mecânica Quântica 10

1.3 Complexidade Computacional 14

2 Preliminares **15**

2.1 Problema de Busca do Máximo e Mínimo 15

2.2 Algoritmo de Grover para busca 16

2.3 Transformada de Fourier Quântica 22

2.3.1 Estimativa de Fase 24

2.4 Algoritmo Quântico de Contagem 26

2.5 Problema *k-Distinctness* 27

3 Algoritmo para Encontrar a Moda **29**

3.1 Primeira abordagem: EncValModalA 29

3.1.1 Procedimento EncFreqModal 31

3.2 Segunda abordagem: EncValModalB 33

3.3 Algoritmo Final: EncValModalFinal 34

4 Simulações **36**

4.1 Descrição da simulação 36

4.2 Experimentos 38

4.2.1 Simulações para diferentes combinações de N e r 38

4.2.2 Simulações com N e r fixos e t variável 40

5 Conclusão **41**

Referências **43**

Apêndice A – Código da Simulação 6 **46**

Capítulo 1

Introdução

Todas as pessoas realizam tarefas, conscientemente ou não, que requerem o processamento de informações, no qual uma saída é produzida a partir da computação —em seu sentido mais amplo— de uma entrada. A própria pesquisa de como fazer algo, ou como fazê-lo da forma mais eficiente possível, pode ser visto como o processamento de informações. Por exemplo, dado uma série de compromissos com seus respectivos horários, alguém pode processar essa entrada colocando os compromissos em uma lista, um a um, do horário mais cedo ao mais tardio. Dessa forma, os compromissos e seus respectivos horários (entrada) foram processados de forma a obter uma lista ordenada dos compromissos (resultado).

Este exemplo é um caso do *problema de ordenação*, que consiste em encontrar a partir de uma lista de números $x = (x_0, x_1, ..., x_{N-1})$, uma permutação σ do conjunto $\{0, ..., N-1\}$ tal que a lista $(x_{\sigma(0)}, x_{\sigma(1)}, ..., x_{\sigma(N-1)})$ está em ordem crescente [22]. Algoritmos de ordenação baseados em comparação, possuem limite inferior $\Omega(N \log N)$, enquanto outros algoritmos, em casos restritos, podem possuir complexidade $\Theta(N)$ [15]. Entretanto, se possuíssemos um computador quântico à disposição, seria possível resolver tal problema de forma mais eficiente? Antes de responder essa indagação, entendamos um pouco melhor como funciona o modelo computacional clássico—i.e., o modelo utilizado atualmente— e o modelo quântico.

Como vimos anteriormente, através de alguma representação mental, as pessoas conseguem processar as informações da tarefa. Mas como uma máquina deve representar e trabalhar estas informações?

Comecemos analisando o modelo computacional clássico. Todos os computadores neste modelo trabalham internamente com o sistema numérico binário, que consiste em dois números, 0 e 1. Ou seja, estes dois números são os moldes que permitem construir

as informações que podem ser representadas ou modeladas pelas máquinas que usamos. Fisicamente, este sistema é modelado através de dois níveis de tensão bem estipulados, e manipulados por transistores, que são unidades elementares de transformação de informação. De forma que, o aperfeiçoamento da capacidade de processamento dos nossos dispositivos se dá pelo aumento na densidade de transistores nos chips, ou seja, pelo aumento no número de transistores menores.

Logo, na década de 60, Gordon Moore, um dos fundadores da empresa Intel, formulou a hipótese —conhecida como lei Moore— de que a cada 18 meses a capacidade de processamento dobra, e o tamanho necessário para representação de um bit se reduz pela metade. Em 1950, eram necessários 10^{19} átomos —ou seja, 10 bilhões de bilhões— para representar um único bit de informação [35]. Se o desenvolvimento das tecnologias de modelagem deste sistema continuarem avançando desta forma, em 2045 cada bit deve ser representado por um único átomo [25].

No entanto, no fim do século XIX, notou-se que a física clássica —utilizada no modelo clássico de computação— levou a previsões contrárias a resultados experimentais [7]. Logo se descobriu a influência de efeitos nos sistemas físicos que possuem dimensões próximas ou abaixo da escala atômica —como moléculas, átomos e outras partículas subatômicas— que não podem ser desprezados. Isto dá origem à mudança nos conceitos básicos do nosso entendimento da natureza e a uma nova teoria dentro da física, a mecânica quântica. Estes efeitos implicam, por exemplo, que propriedades do modelo computacional clássico, como a possibilidade de cópia de dados, passam a não serem mais válidas.

Uma proposta interessante para solução destes problemas foi o processamento de informações em sistemas quânticos, de forma que o estudo das tarefas realizadas por este processamento foi denominado computação quântica. Essa mudança de paradigma proporciona que as interferências, antes desvantagens no processamento de informações, se tornem novas ferramentas para cientistas da computação, permitindo que certos problemas sejam resolvidos com maior rapidez. Entretanto, para estas vantagens serem utilizadas, novos algoritmos devem ser desenvolvidos de forma que essas propriedades exclusivas dos computadores quânticos –como o paralelismo quântico, emaranhamento e interferência– sejam bem exploradas. Diversos problemas [19, 39] já podem ser resolvidos com algoritmos quânticos com maior eficiência do que na computação clássica.

As ideias fundamentais do modelo quântico de computação foram geradas ao longo de anos por diversas áreas, como a mecânica quântica, a ciência da computação, a teoria da informação e a criptografia. Nielsen e Chuang [34] abordam esse processo de

desenvolvimento com mais detalhes em seu livro.

No entanto, apesar de conhecermos algoritmos quânticos mais eficientes para problemas como o de busca ou de fatoração, não é possível afirmar que computadores quânticos são mais poderosos que os computadores clássicos, no sentido em que existam problemas que sejam tratáveis no modelo quântico e não os sejam no modelo clássico [34]. De toda forma, podemos afirmar até o momento, que ter um computador quântico não significa solucionar todos os problemas de forma mais eficiente que em um computador clássico. O problema de ordenação, sem conhecimento prévio ou restrições da entrada, é um destes problemas; possuindo um limite inferior igual —$\Omega(N \log N)$— em ambos os modelos, clássico e quântico [22]. Em seu livro [34], Nielsen e Chuang abordam os pormenores do assunto.

Finalmente, é importante destacar que computadores quânticos eficientes ainda são considerados objetivos distantes. Entretanto, notáveis resultados na construção de hardware quânticos escaláveis foram alcançados recentemente [16, 41], o que sugere que não há de demorar até que computadores quânticos se tornem práticos.

1.1 Descrição do Problema

Nós vivemos na era da informação, na qual uma grande quantidade de conhecimento já processado está presente nos muitos aspectos da nossa vida em sociedade. Muitas destas informações foram determinadas matematicamente usando modelos estatísticos. Portanto, a estatística é fundamental no funcionamento da nossa sociedade. Indivíduos, empresas e governo dependem fortemente de dados estatísticos —como taxa de inflação, dados para previsão do tempo, previsões econômicas, só para citar alguns— no intuito de se tomar decisões informadas. A *moda* é um destes valores estatísticos que fornece informações úteis na solução de alguns problemas, como por exemplo o de identificar o candidato mais votado em uma eleição, ou encontrar a palavra mais frequente em um texto.

A *moda*, ou *valor modal*, é o elemento com maior recorrência em um conjunto de dados. Se um conjunto de dados possui dois ou mais valores modais, estes são denominados *bimodal* ou *multimodal*, respectivamente. O valor modal é uma informação estatística importante, podendo ser usada em estimações ou aplicações especializadas, e é uma das poucas medidas de tendencia central que podem ser usadas em um nível nominal de mensuração. Nós definimos a *frequência* de um elemento como o número de ocorrências do

mesmo no conjunto de dados de entrada por execução do algoritmo, e definimos *frequência modal* como a frequência dos valores modais. Por fim, definimos a *faixa de dados* de um conjunto de elementos como a diferença entre os valores máximos e mínimos da sequência de bits que representa cada entrada do conjunto.

Os melhores algoritmos clássicos para encontrar o valor modal consistem em ordenar os dados de entrada, contar as repetições de cada um dos seus N elementos, e por fim, determinar o elemento com o maior número de repetições, o que requer $O(N \log N)$ consultas à entrada se não fizermos nenhuma suposição sobre a entrada [40]. Se assumirmos algumas restrições sobre a entrada —como por exemplo a distribuição uniforme dos valores do vetor de entrada ou a limitação intrínseca da faixa de dados de um problema—, então pode ser possível a aplicação de algoritmos de Contagem linear —como bucket sort ou radix sort— para encontrar o valor modal realizando $O(N)$ consultas à entrada.

Em 2005, Kara [23] introduziu um algoritmo quântico randomizado e aproximativo para encontrar a moda de um conjunto de de dados com N elementos, todos retirados de um conjunto de M valores possíveis, realizando $O(M^{3/2} \log(1/\delta)/\epsilon)$ consultas com aproximação $(1+\epsilon)$ e probabilidade de pelo menos $(1\text{-}\delta)$.

Em 2008, Coffey e Prezkuta [14] apresentaram um algoritmo quântico para encontrar a moda. O algoritmo CP afirma requerer $O(M\sqrt{N})$ consultas sobre uma lista de entrada de valores inteiros $(x_1, \ldots, x_N)$ com variação $x_j \in \{1, ..., M\}$. Entretanto, Montanaro [31] questiona a sua corretude, considerando que uma computação exata da frequência modal, onde $M = 2$, seria equivalente a solucionar o *majority element problem* com uma complexidade de $O(\sqrt{N})$, possuindo uma complexidade estabelecida de $\Omega(N)$ [1, 4].

Nós introduzimos duas abordagens, quânticas e probabilísticas, para encontrar o valor modal, EncValModalA e EncValModalB, com complexidades de consulta $o(N^{3/4} \log N)$ e $O(r\sqrt{N})$, respectivamente, onde r é a faixa de dados da entrada. Por fim, propomos um algoritmo que busca explorar os melhores casos de cada um dessas abordagens, em termos de complexidade. Esse algoritmo requer $\min(o(N^{\frac{3}{4}} \log N), O(r\sqrt{N}))$ consultas para encontrar o valor modal e a frequência modal de um conjunto de dados. De modo que, se $r \geq \sqrt[4]{N} \log N$, nós executamos a abordagem EncValModaA (veja Algoritmo 2) como sub-rotina, que se baseia no algoritmo quântico para solucionar o problema *element k-Distinctness* [5], com probabilidade de acerto de pelo menos $1/2$. Caso contrário, executamos a abordagem EncValModalB (veja Algoritmo 4) como sub-rotina, que se baseia no Algoritmo Quântico de Contagem (veja Seção 2.4), com probabilidade de acerto de pelo menos $2/3$ e aproximação na ordem de $O(\sqrt{\min(p, N-p)}/c)$, onde p é a frequência

do elemento contado e c uma constante.

O algoritmo proposto neste trabalho, não conflita com a observação feita por [31], a respeito do limite inferior do *majority element problem*, pois leva em conta que as frequências calculadas pelo algoritmo de Contagem são aproximadas. O *majority problem*[42] consiste em encontrar o elemento com recorrência maior que $N/2$, caso este exista, de uma lista não-ordenada de tamanho N. Por fim, vale notar que, devido a esta característica aproximativa e à probabilidade das abordagens que o compõem, o algoritmo proposto também é probabilístico.

1.2 Breve Introdução à Mecânica Quântica

A mecânica quântica é um modelo matemático capaz de descrever fenômenos físicos da natureza em escalas atômica e subatômica. Para caracterizar este modelo, é necessário definir como são suas representações básicas: estados, observabilidade, evolução etc [38].

As representações básicas da mecânica quântica estão fundamentadas na f linear, no estudo dos espaços vetoriais e das operações neles realizadas. A compreensão sólida de seus conceitos elementares implica diretamente na assimilação da teoria da mecânica quântica. Neste seção fazemos uma breve descrição de alguns de seus conceitos. Entretanto, para os leitores com pouca familiaridade no assunto, recomenda-se o livro de Nielsen e Chuang [34] como principal objeto de estudo, dos conceito de álgebra linear, seus usos e representações básicas na mecânica quântica. Outras fontes secundárias de estudo são as notas de aulas de John Preskill [38] e o livro escrito por David Mermim [29].

1.2.1 Espaço de Hilbert

Como nos lembra Nielsen e Chuang [34], trabalhos na área de mecânica quântica frequentemente mencionam o *espaço de Hilbert* ($\mathcal{H}$). No que diz respeito aos interesses da computação quântica e a informação quântica, especificamente espaços vetoriais complexos de dimensão finita, o espaço de Hilbert é o estado vetorial dotado de um produto interno no qual todos os estados quânticos estão presentes. Na computação quântica, a notação padrão utilizada para representação dos vetores em $\mathcal{H}$ é a *Bra-ket* de Dirac [17].

Na Tabela 1.1, encontra-se um resumo de alguns dos principais conceitos da álgebra linear usando a notação de Dirac.

Notação	Descrição
z*	Conjugado de z
$\|\psi\rangle$	Vetor φ, também chamado de *ket*
$\langle\psi\|$	Vetor dual de $\|\varphi\rangle$, também chamado de *bra*
$\langle\varphi\|\psi\rangle$	Produto escalar entre $\|\varphi\rangle$ e $\|\psi\rangle$
$\|\varphi\rangle \otimes \|\psi\rangle$	Produto tensorial entre $\|\varphi\rangle$ e $\|\psi\rangle$
$\|\varphi\rangle \|\psi\rangle$	Notação abreviada para o produto tensorial entre $\|\varphi\rangle$ e $\|\psi\rangle$
A^*	Conjugado da matriz A
A^T	Transposta da matriz A
$A^\dagger$	Conjugado transposto, ou matriz adjunta de A, $A^\dagger = (A^T)^*$
	$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^\dagger = \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix}$
$\langle\varphi\|A\|\psi\rangle$	Produto escalar entre $\|\varphi\rangle$ e $A\|\psi\rangle$
	Equivalentemente, produto escalar entre $A^\dagger \|\varphi\rangle$ e $\|\psi\rangle$

Tabela 1.1: Resumo da notação-padrão utilizada em mecânica quântica para conceitos de álgebra linear. Este tipo de notação é chamado de notação de Dirac. Fonte: [34]

Pela representação de Dirac, o espaço de Hilbert $\mathcal{H} : \langle\omega|\psi\rangle$, é mapeado por um par de vetores ordenados a um espaço vetorial sobre $\mathbb{C}$, e de acordo com Preskill [38], satisfaz as seguintes propriedades:

- Positividade: $\langle\omega|\omega\rangle \geq 0$, com a igualdade valendo somente se $|\omega\rangle = 0$.
- Linearidade: $\langle\psi| (\lambda_1 |\omega_1\rangle + \lambda_2 |\omega_2\rangle) = \lambda_1 \langle\psi|\omega_1\rangle + \lambda_2 \langle\psi|\omega_2\rangle$, onde λ_1 e λ_2 são escalares.
- Simetria oblíqua: $\langle\omega|\psi\rangle = \langle\psi|\omega\rangle^*$, onde * é o conjugado do número complexo.

Por fim, definimos $\mathcal{H}$ como *completo* na norma $\|\omega\| = \sqrt{\langle\omega|\omega\rangle}$. Esta é uma importante condição em espaços funcionais de dimensão infinita. Entretanto, visto que neste trabalho trabalharemos com espaços vetoriais de dimensão finita, esse ponto é de pouca relevância.

1.2.2 Bit Quântico

Na computação quântica, os estados clássicos 0 e 1 de um bit são substituídos por vetores $|0\rangle$ e $|1\rangle$ de um bit quântico [37]. O bit quântico (ou q-bit) descreve o estado do menor sistema quântico possível, representado no espaço de Hilbert bidimensional.

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ e } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{1.1}$$

Entretanto, esta ainda não é principal diferença entre um bit e um q-bit. A principal diferença está no fato de que um q-bit genérico $|\omega\rangle$, pode também ser uma combinação linear entre $|0\rangle$ e $|1\rangle$, ou seja,

$$|\omega\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{1.2}$$

onde α e β são números complexos. Os vetores $|0\rangle$ e $|1\rangle$ formam uma base ortonormal do espaço vetorial $\mathbb{C}^2$ conhecida como *base computacional*. Dizemos que o vetor $|\omega\rangle$, ou *estado* $|\omega\rangle$, é uma *superposição* e que α e β são as *amplitudes* de $|0\rangle$ e $|1\rangle$, respectivamente. Nos casos em que nenhuma das amplitudes são zero, não é possível dizer que o q-bit está em $|0\rangle$ ou em $|1\rangle$.

Devido às leis da mecânica quântica, existem algumas complicações na computação quântica não presentes no modelo de computação clássico. Uma dessas complicações é a de leitura de informações (ou medições). Ainda que no exemplo acima, através da variação da amplitude de α e β, se é possível armazenar uma quantidade infinita de informação, esta informação está no nível quântico. Para acessar esta informação, no nível clássico, uma ação sobre o sistema, a qual chamamos de medição é necessária. Trataremos melhor deste assunto quando abordarmos os postulados da mecânica quântica, na seção 1.2.4.

Como mencionado acima, para acessar as informações, no nível clássico, é necessário realizar uma medição sobre o sistema. Essa medição, altera o estado de um q-bit, fazendo-o assumir o estado $|0\rangle$ ou $|1\rangle$, onde a probabilidade desses estados serem medidos é de $\mid \alpha \mid^2$ e $\mid \beta \mid^2$, respectivamente. Uma vez que, uma medição de um q-bit sempre retorna um dos seus estados, temos que

$$\mid \alpha \mid^2 + \mid \beta \mid^2 = 1, \tag{1.3}$$

que é equivalente a $\langle\omega|\omega\rangle$, e consequentemente, concluímos que ω é unitário. De forma sucinta, podemos dizer um q-bit é um vetor de norma 1 de $\mathbb{C}^2$.

Nota-se que na Equação (1.1), os estados $|0\rangle$ e $|1\rangle$ não possuem valores com parte imaginária, e por isso são omitidas. Entretanto, considerando que $\mid \alpha \mid = a + ib (a, b \in \mathbb{R})$

e $\mid \beta \mid = c + id (c, d \in \mathbb{R})$, temos que a Equação (1.3)pode ser reescrita como:

$$\mid a \mid^2 + \mid b \mid^2 + \mid c \mid^2 + \mid d \mid^2 = 1. \tag{1.4}$$

Essa condição também é conhecida como *condição de normalização* para vetores de estado. Por fim, descrevemos um estado $|\omega\rangle$ como uma combinação linear $\omega = \sum_j \alpha_j |\omega_j\rangle$, e dizemos que $|\omega\rangle$ é uma superposição de estados $|\omega_j\rangle$, com amplitudes α_j [34]. Um exemplo disto é o estado conhecido como $|-\rangle$:

$$|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \tag{1.5}$$

que é uma superposição formada pelo somatório dos estados $|0\rangle$ e $|1\rangle$, e suas respectivas amplitudes, $1/\sqrt{2}$ e $-1/\sqrt{2}$.

1.2.3 Circuitos Quânticos

Antes de continuar com a definição dos axiomas da mecânica quântica, é de grande ajuda entender as notações utilizada na representação dos circuitos quânticos e portas lógicas. Este estudo baseia-se no fato de que toda matriz unitária 2x2 pode ser representada por um circuito quântico de um q-bit e vice-versa [34]. Sendo assim, a evolução no tempo de um sistema isolado, dado por um q-bit, pode ser representada tanto matematicamente, por uma transformação unitária, quanto logicamente, por um circuito quântico [37]. Para ser mais concretos, utilizamos o exemplo da Figura 1.1.

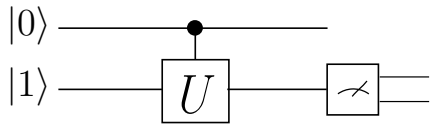

Figura 1.1: Circuito U-controlado

Primeiramente analisemos a entrada do circuito. O estado de um sistema, seja inicial ou em qualquer instante do circuito, pode ser definido pelo produto tensorial dos estados q-bits ou por um estado emaranhado —aprofundaremos mais sobre essas características na próxima seção. No exemplo acima temos que cada q-bit é assinalado como no estado $|0\rangle$ e $|1\rangle$, respectivamente. Portanto, a entrada deste circuito pode ser definida como $|0\rangle \otimes |1\rangle$.

As linhas horizontais do circuito tem como função representar a evolução do q-bit. Enquanto a linhas verticais, informam a atuação simultânea do circuito nos q-bits indicados, não havendo qualquer envio de informação entre estes. Neste caso específico temos

que a linha vertical liga os elementos da operação controlada —abordaremos o assunto com mais detalhe abaixo.

É importante destacar que, ao contrário dos circuitos clássicos, os circuitos quânticos não possuem de retroalimentações. Logo, o circuito é construído de forma que a evolução do sistema quântico no tempo ocorre da esquerda para a direita.

Por fim, os q-bits que compõem a saída do circuito podem ou não ser medidos. No exemplo da Figura 1.1, temos que uma medição está sendo realizada sobre o segundo q-bit. Uma vez que o operador de U não está sendo efetivamente aplicado ao segundo q-bit, temos que o valor medido é 1. Quanto às propriedades da medição, trataremos com mais detalhes na próxima seção.

A razão pela qual o operador U não está sendo efetivamente aplicado ao segundo q-bit está ligada à peculiaridade do símbolo $\bullet$, conectado verticalmente à U. "Se A for verdadeiro, execute B". Esse tipo de *operação controlada* é muito utilizada tanto na computação clássica quanto na quântic [34]. Desta frase, extraímos os dois principais componentes de uma operação controlada, a operação a ser realizada e a condição, que na computação quântica se traduzem no operador e nos controladores, respectivamente. De forma que um operador assume a condição de controle e recebe a notação 'operador $U-controlado$', para algum operador U. O simbolo $\bullet$ indica que o q-bit representado na linha é um q-bit de controle, ou seja, caso esteja no estado $|1\rangle$, o operador (ou porta) U realiza a operação; caso esteja no estado $|0\rangle$, o operador U não realiza operação alguma. Uma ação controlada pode ser composta por vários q-bits de controle, e todos devem atender à condição para que o operador realize a operação.

Dependendo do problema a ser trabalhado, os q-bits de controle podem estar *superpostos* ou *emaranhados*. Nestes casos, é necessário considerar a ação do operador unitário, que representa todo o circuito, atuando simultaneamente sobre os q-bits de controle e os q-bits alvos da operação.

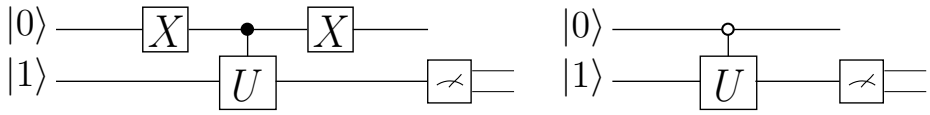

Figura 1.2: Condições de controle

Além do $\bullet$, existe outro símbolo também utilizado em operações controladas, o símbolo $\circ$. Eles tem como diferença como diferença a negação da condição de execução da operação, ou seja, nas operações controladas em que o símbolo $\circ$ é utilizado sobre o q-bit de controle, este q-bit precisa estar no estado $|0\rangle$ para que a operação seja realizada. A

operação controlada utilizando o símbolo $\circ$ pode ser obtida a partir da aplicação de uma porta X —ao qual a generalização para o caso clássico é seria a porta NOT— sobre o q-bit de controle antes e depois da operação. De forma que os circuitos da Figura 1.2 são equivalentes. Temos que o operador X é definido pela matriz

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{1.6}$$

Um outro operador muito utilizado na computação quântica é o *operador identidade* —ou simplesmente I—, definido pela *matriz identidade*. O operador identidade mapeia o estado para si próprio, i.e., ele não altera o estado. De forma que, sempre que necessita aplicar um operador a apenas alguns q-bits do sistema, se aplica o operador de identidade sobre os outros q-bits. Considere a Figura 1.3 por exemplo, onde um operador X é aplicado ao primeiro q-bit, transformando seu estado de $|0\rangle$ para $|1\rangle$, enquanto o segundo q-bit permanece com o estado inalterado. Neste caso, temos que o operador $U = I \otimes X$ está sendo aplicado ao estado do sistema. Note que, no circuito, o operador I aparece como um fio, ou linha horizontal. Aprofundaremos um pouco mais sobre o assunto na na sub-seção seguinte.

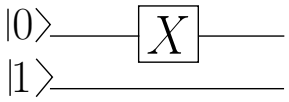

Figura 1.3: Circuito Quântico exemplificando o operador Identidade

Uma outra porta, de dimensão 2x2, muito utilizada na mecânica quântica é a porta Hadamard H, definida pelo operador

$$H = 1/\sqrt{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{1.7}$$

Como veremos no quarto postulado da mecânica quântica, na sub-seção seguinte, o operador de Hadamard é utilizado principalmente para obter a *superposição* de estados.

1.2.4 Postulados da Mecânica Quântica

Diversas informações e implementações físicas se relacionam com os postulados da mecânica quântica. Entretanto, não nos ateremos a estes detalhes neste trabalho. Esta sub-seção tem como objetivo apresentar apenas um panorama matemático-computacional dos postulados.

Postulado 1: *A qualquer sistema físico isolado existe associado um espaço vetorial complexo com produto interno (i.e., um espaço de Hilbert), conhecido como espaço de estados do sistema. O sistema é completamente descrito pelo seu vetor de estado, um vetor unitário em um espaço de estados.*

A mecânica quântica não nos diz qual exatamente é o espaço de estados de um dado sistema, tampouco o seu vetor de estado [34]. Entretanto, ainda que não explicite qual é o vetor de estados, o primeiro postulado nos indica que em um determinado instante, o sistema está associado a um estado bem definido [24]. Por fim, como dito na sub-seção 1.2.2, a partir Equação (1.4), fundamenta-se a condições de unitariedade do vetor de estados.

Postulado 2: *A evolução de um sistema quântico fechado é descrita por uma transformação unitária. Ou seja, o estado $|\omega\rangle$ de um sistema em um tempo t_1 está relacionado ao estado $|\omega'\rangle$ do sistema em t_2 por um operador unitário U que depende somente de t_1 e t_2:*

$$|\omega'\rangle = U\,|\omega\rangle\,. \tag{1.8}$$

Esse postulado descreve como os estados de um sistema quântico fechado, em dois instantes de tempos, estão relacionados. Assim como o primeiro postulado afirma da existência de um estado bem definido em um determinado instante, ainda que sem explicitar seu vetor de estados, o segundo postulado indica que a evolução dos estados em um sistema fechado se dá através da aplicação de operadores unitários —ou seja, operadores que não alteram a unitariedade do estado do sistema—, mas não explicita quais são essas transformações para um caso real em particular [24].

Os operadores unitários são representados matematicamente como matrizes, que como dito anteriormente, não alteram a unitariedade do estado do sistema. Uma característica destas matrizes é que, para qualquer matriz unitária U, $U^\dagger U = I$, onde I é a matriz identidade [34]. O estado $|-\rangle$ definido na Equação (1.5) por exemplo, é resultado da evolução

$$H\,|1\rangle = \frac{1}{\sqrt{2}}\,|0\rangle - \frac{1}{\sqrt{2}}\,|1\rangle = |-\rangle\,, \tag{1.9}$$

onde o operador de Hadamard é aplicado ao estado $|1\rangle$.

Postulado 3: *As medições quânticas são descritas por determinados operadores de medição $\{M_m\}$. Esses operadores atuam sobre o espaço de estados do sistema. O índice*

'm' se refere aos possíveis resultados da medição. Se o estado de um sistema quântico for algum estado $|\omega\rangle$, no instante antes da medição, a probabilidade de um resultado'm' ocorrer é dada por:

$$\rho(m) = \langle\omega| M_m^\dagger M_m |\omega\rangle , \tag{1.10}$$

e o estado do sistema após a medição será:

$$\frac{M_m |\omega\rangle}{\sqrt{\langle\omega| M_m^\dagger M_m |\omega\rangle}}. \tag{1.11}$$

Os operadores de medição satisfazem a relação de completude:

$$\sum_m M_m^\dagger M_m = I. \tag{1.12}$$

Primeiramente, é importante notar que os postulados definidos anteriormente, utilizam como condição o isolamento do sistema —ou seja, de que esteja fechado. Entretanto, em algum momento, uma intervenção externa que acaba com este isolamento será necessária, caso contrário todo o processamento das informações não teria fim.

A medição, que tem como função conhecer a situação dentro do sistema fechado, é um dos exemplos caracterizados por essa intervenção, além de não necessariamente ser descrita por uma transformação unitária [34].

Portanto, se por exemplo, desejamos fazer uma medição em um q-bit no estado $|\omega\rangle = \alpha |0\rangle + \beta |1\rangle$ —na base computacional—, temos os operadores de medição:

$$\begin{aligned} M_0 &= |0\rangle \langle 0| , \\ M_1 &= |1\rangle \langle 1| . \end{aligned} \tag{1.13}$$

A partir da Equação (1.12), obtém-se que

$$\sum_m M_m^\dagger M_m = M_0 + M_1, \tag{1.14}$$

ou seja, que o conjunto $\{M_0, M_1\}$ satisfaz a relação de completude. Logo, a probabilidade de se obter 0 como resultado da medição é

$$\rho(0) = \langle\omega| M_0^\dagger M_0 |\omega\rangle =| \alpha |^2 . \tag{1.15}$$

De forma análoga, temos que $\rho(1) =| \beta |^2$.

Postulado 4: *O espaço de estados de um sistema físico composto é o produto tensorial dos espaços de estado dos sistemas físicos individuais. Se os sistemas forem numerados de 1 até n, e o sistema j for preparado no estado* $|\omega_j\rangle$*, decorre que o estado do sistema composto será* $|\omega_1\rangle \otimes |\omega_2\rangle \otimes \cdots \otimes |\omega_n\rangle$.

Até o momento, diversos exemplos foram demonstrados aplicando definições apresentadas ao menor sistema quântico possível, representado em $\mathcal{H}^2$. No entanto, as aplicações a problemas reais certamente requerem um sistema com mais de um q-bit. O postulado 4 modela a criação destes sistemas maiores a partir de sistemas menores. Por exemplo, considere um sistema de dois q-bits: estando, individualmente, cada q-bit no estado $|\omega_j\rangle = \alpha_j |0\rangle + \beta_j |1\rangle$; podemos dizer que este sistema está no estado $|\omega\rangle = |\omega_0\rangle \otimes |\omega_1\rangle$, que equivale a

$$\begin{aligned} |\omega\rangle = \alpha_0\alpha_1(|0\rangle \otimes |0\rangle) + \alpha_0\beta_1(|0\rangle \otimes |1\rangle) \\ +\beta_0\alpha_1(|1\rangle \otimes |0\rangle) + \beta_0\beta_1(|1\rangle \otimes |1\rangle). \end{aligned} \tag{1.16}$$

Caso se deseje adicionar um outro q-bit ao sistema composto $|\omega\rangle$, obtemos um sistema $|\omega'\rangle = |\omega\rangle \otimes |\omega_2\rangle$, que equivale a

$$\begin{aligned} |\omega'\rangle = \alpha_0\alpha_1\alpha_2(|0\rangle \otimes |0\rangle \otimes |0\rangle) + \alpha_0\alpha_1\beta_2(|0\rangle \otimes |0\rangle \otimes |1\rangle) \\ + \alpha_0\beta_1\alpha_2(|0\rangle \otimes |1\rangle \otimes |0\rangle) + \alpha_0\beta_1\beta_2(|0\rangle \otimes |1\rangle \otimes |1\rangle) \\ + \beta_0\alpha_1\alpha_2(|1\rangle \otimes |0\rangle \otimes |0\rangle) + \beta_0\alpha_1\beta_2(|1\rangle \otimes |0\rangle \otimes |1\rangle) \\ + \beta_0\beta_1\alpha_2(|1\rangle \otimes |1\rangle \otimes |0\rangle) + \beta_0\beta_1\beta_2(|1\rangle \otimes |1\rangle \otimes |1\rangle). \end{aligned} \tag{1.17}$$

Nota-se que a Equação (1.17), que descreve o estado $|\omega'\rangle$, é bem grande. Todavia, existem algumas formas reduzidas de se descrever estes estados. Uma dessas formas é a omissão do sinal da operação de produto tensorial, de forma que um estado $|1\rangle \otimes |0\rangle$ é descrito como $|1\rangle |0\rangle$. Outros modos de redução são mais específicos, por exemplo, dado um estado $|a_0\rangle |a_1\rangle |a_2\rangle \ldots |a_n\rangle$ tal que $a_j \in \{0, 1\}$, este estado pode ser reescrito na representação binária $|a_1 a_2 \ldots a_n\rangle$, ou na representação decimal equivalente —por exemplo, o estado $|001\rangle$ na representação binária pode ser reescrito como $|7\rangle$ na representação decimal.

Como visto acima, o produto tensorial de sistemas menores descrevem um sistema maior. Entretanto, a afirmação não é recíproca, ou seja, um sistema maior não necessariamente pode ser descrito como o produto tensorial de sistemas menores. Nesses casos, dizemos que o sistema está *emaranhado* [24]. Considere, por exemplo, o estado de dois

q-bits

$$|\omega'\rangle = \frac{|00\rangle + |11\rangle}{2}. \tag{1.18}$$

É possível chegar a este estado com dois q-bits aplicando o operador controlado $CNOT$ ao estado $|\omega\rangle = |+\rangle \otimes |0\rangle$, contudo não é possível descrevê-lo como o produto tensorial de dois estados .Os únicos resultados possíveis da medição dos dois q-bits de $|\omega'\rangle$, são ambos resultarem em 1 ou ambos resultarem em 0. O estado $|+\rangle$ é obtenível a partir da transformação $H\,|0\rangle \rightarrow |+\rangle$, enquanto o operador $CNOT$ é representado matricialmente como

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{1.19}$$

1.3 Complexidade Computacional

Na computação, a noção mais utilizada para mensuração de complexidade é a de tempo e espaço. Outras medidas de complexidade também são utilizadas para tais, como a quantidade de comunicação (usada em complexidade de comunicação[27]), o número de portas e o comprimento um circuito (usado na complexidade de circuito), o número de processadores (usados em computação paralela) etc. Ao longo desse artigo, a medida utilizada para mensuração dos algoritmos quânticos é o número de consultas ao oráculo —procedimento que está relacionado ao acesso aos valores de entrada—, sua notação é denominada complexidade de consulta. De maneira geral, no modelo computacional quântico, a complexidade de consulta é determinada pelo número mínimo de consultas ao oráculo para obter um determinado resultado, não necessariamente correto [34].

Normalmente, é muito complicado provar um limite inferior forte na complexidade de tempo de um procedimento, visto que estes podem usar representações de dados muito sofisticados, que são difíceis de se raciocinar sobre. Uma solução para este problema é o uso da complexidade de consulta, posto que o limite inferior no número de acessos a entrada é também um limite inferior na complexidade de tempo [6].

Note que, a complexidade de consulta de um algoritmo não necessariamente é igual à sua complexidade de tempo, mas ela infere um limite inferior para esta. Para uma explicação mais detalhada sobre o campo da teoria da complexidade computacional quântica, recomenda-se a leitura do livro de Nielsen e Chuang [34] e a tese por Belovs [6].

Capítulo 2

Preliminares

Para solucionar o problema de encontrar a moda classicamente, diferentes abordagens podem ser utilizadas dependendo da faixa de dados da entrada. Essa faixa é calculada pela diferença entre o maior e o menor elemento da entrada. Se a variação dos dados é pequena, então um algoritmo de ordenação linear, como *Counting Sort* ou *Radix Sort*, pode ser usado ao invés dos típicos algoritmos de ordenação, que possuem complexidade $O(N \log N)$. Isso posto, a partir de um conjunto de dados ordenados, é possível encontrar linearmente o elemento mais frequente. No algoritmo proposto neste trabalho, a faixa de dados é utilizada para decidir qual sub-rotina aplicar, que por sua vez, reduz o problema de encontrar moda a outros problemas já solucionados no modelo quântico. Estes problemas são: *k-Distinctness*, Busca de Máximo e Mínimo e Contagem.

De modo geral, Mosca [33] defende a separação dos principais algoritmos quânticos em dois grupos: os que exploram as características únicas da Transformada de Fourier Quântica (TFQ) —como por exemplo, o Algoritmo de Shor para Fatoração [39] e o Algoritmo Quântico de Contagem [9]— e os que se baseiam no algoritmo de Grover de Busca Quântica [19]. Este último grupo de algoritmos pode ainda ser classificado em três subgrupos: amplificação de amplitude [9, 10, 11, 21], estimativa de amplitude quântica [11, 32] e os casos especiais destes. Visto que, o algoritmo proposto neste trabalho (veja o Capítulo 3) tem como fundamentos algoritmos e técnicas de ambos os grupos, utilizamos este capítulo introduzi-los.

2.1 Problema de Busca do Máximo e Mínimo

Dürr e Høyer [18] propuseram, em 1996, um algoritmo quântico para encontrar o valor mínimo de uma lista não-ordenada —problema de Busca do Mínimo— realizando $\sqrt{N}$

consultas ao oráculo e $\Omega(\log^2 N)$ medições. Entretanto, em 2008, Kowada *et al.* [26] apresentaram o algoritmo probabilístico KLPF (veja o Algoritmo 1), também para o problema de Busca do Mínimo, com a mesma ordem de complexidade que o algoritmo proposto por Dürr e Høyer, mas com uma redução quadrática no número de medições.

Algoritmo 1 Algoritmo KLPF

Saída: A função f

1: Escolha aleatoriamente um índice x tal que $0 \leq x \leq N-1$.
2: $y \leftarrow f(x)$
3: $i \leftarrow 0$
4: Inicialize $\lambda \leftarrow 13/14$ e $m \leftarrow \lambda$
5: **enquanto** $i < 8,96\sqrt{N}$ **faça**
6: Escolha aleatoriamente $j < m$.
7: Inicialize o sistema no estado $|\varphi_0\rangle = |0\rangle\,|y\rangle$.
8: Faça a superposição de todos a base computacional de estados no primeiro registrador.
9: **para** j iterações **faça**
10: Aplique U_g para reverter a amplitude de estados com imagem menor que y.
11: Aplique o procedimento de inversão sobre a média
12: Observe o primeiro registrador e armazene em x' o valor medido.
13: **se** $f(x') < y$ **então**
14: $y \leftarrow f(x')$
15: $i \leftarrow i + j$
16: **se** i>11 **então**
17: $m \Leftarrow \min(\lambda m, \sqrt{N})$
18: **return** y e o valor correlacionado ao primeiro registrador.

Utilizamos neste trabalho, como caixa preta, o algoritmo KLPF, no que se refere às sub-rotinas *FindMinimum* e *FindMaximum*. Essas sub-rotinas recebem como parâmetro de entrada uma lista de inteiros L, e retornam o valor mínimo ou o valor máximo presentes nesta, respectivamente. Estas sub-rotinas possuem uma probabilidade de sucesso de pelo menos 50%. Note que, o problema de Busca do Mínimo é equivalente ao problema de Busca do Máximo.

2.2 Algoritmo de Grover para busca

Desenvolvido em 1996 por Lov K. Grover, o algoritmo que toma seu nome foi desenvolvido inicialmente para a busca em uma lista não-ordenada [20], sendo similarmente capaz de solucionar problemas de busca em geral [2, 34]. Posteriormente, começou a ser utilizado na construção de algoritmos quânticos de otimização discreta [25].

Dado uma função $\boldsymbol{f : X \rightarrow Y}$ e um valor $\boldsymbol{k \in Y}$, definimos o problema de busca

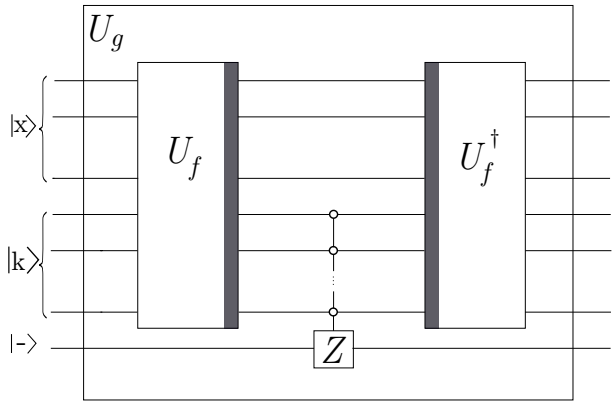

Figura 2.1: Circuito porta U_g

associado a uma função como o de encontrar um valor $\boldsymbol{x}$ tal que $\boldsymbol{f(x) = k}$[25]

O algoritmo de Grover, na sua versão original, traduz-se na aplicação sucessiva da operação conhecida como *iteração de Grover*. A iteração de Grover é dada pela concatenação de operadores $G = (-HU_0H)U_g$, que pode ser divida em duas partes: o Oráculo U_g e o procedimento de amplificação de amplitude. U_0 é conhecido como operador deslocamento de fase condicional, ele mapeia o estado $|0\rangle$ para $-|0\rangle$ e deixa os outros estados do sistema inalterados.

Baseando-se na função $\boldsymbol{f}$, é possível criar a função $\boldsymbol{g(x,k)}$, núcleo do operador de oráculo. A função $\boldsymbol{g(x,k)}$ determina se $\boldsymbol{x}$ é solução do problema ou não. Por exemplo, para o problema da busca em um conjunto, $\boldsymbol{g}$ indica a presença de $\boldsymbol{k}$ na mesma ou não, podendo ser descrita como:

$$\begin{cases} 1, & \text{se } f(x) = k \\ 0, & \text{se } f(x) \neq k \end{cases}$$

Kowada [25] mostra que, a partir da função $\boldsymbol{g}$, é possível criar o operador unitário $U_g : |x\rangle\,|k\rangle\,|-\rangle \rightarrow (-1)^{g(x,k)}\,|x\rangle\,|k\rangle\,|-\rangle$. De modo geral, o circuito quântico desta operação, detalhado na Figura 2.1, utiliza uma porta Z de Pauli controlada juntamente com o operador $\boldsymbol{U_f : |x\rangle\,|k\rangle \rightarrow |x\rangle\,|b \oplus f(x)\rangle}$, que por sua vez, pode ser decomposto a uma aplicação bit-a-bit do operador CNOT —representado pelo símbolo $\oplus$ e equivalente ao resultado da porta clássica XOR. É possível construir este operador sem o bit auxiliar, entretanto sua presença facilita o entendimento do operador.

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.1}$$

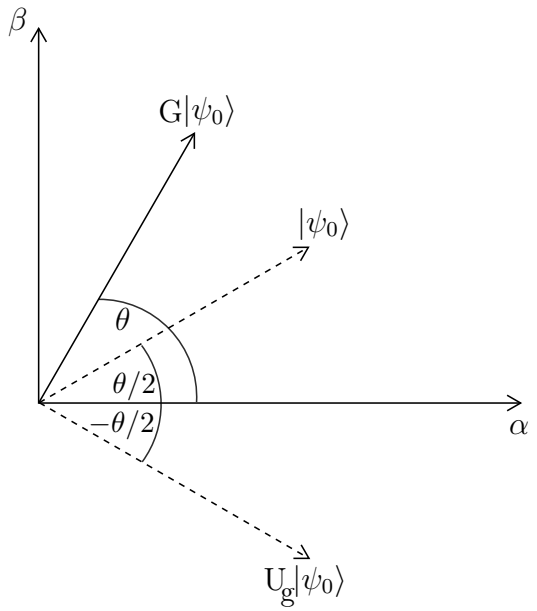

Figura 2.2: $|\psi\rangle$ em um espaço de Hulbert bidimensional, sendo rotacionado por θ a cada aplicação da iteração de Grover

Brassard *et al.* [10] não limitam a iteração de Grover ao uso da Transformada de Hadamard. Entretanto, por não ser o foco deste trabalho, não nos ateremos a esta propriedade.

Uma abordagem interessante para este problema é apresentada por Nilsen e Chuang [34], e consiste na representação do estado em um plano vetorial $\mathcal{H}^2$. Dado um estado quântico preparado $|\psi_0\rangle = \alpha\,|\psi_A\rangle + \beta\,|\psi_B\rangle$, com o objetivo de observar $|\psi_B\rangle$ como resultado de medição, é necessário que as aplicações da iteração de Grover aumentem a amplitude de β, de modo que probabilidade de leitura $|\beta|^2$ seja alta. Repare que os estados $|\psi_A\rangle$ e $|\psi_B\rangle$ são ortogonais, com $|\alpha|^2 + |\beta|^2 = 1$ e *a priori*, suas respectivas amplitudes são desconhecidas.

Através de uma pequena manipulação algébrica, podemos reescrever o estado inicial como $|\psi_0\rangle = \sqrt{(N-M)/N}\,|\psi_A\rangle + \sqrt{M/N}\,|\psi_B\rangle$, onde N é o tamanho do domínio de f e M é o número de vezes que k aparece no conjunto de entrada, ou seja, que $g(x,k) = 1$. Como vimos anteriormente, o oráculo faz uma reflexão em torno do vetor $|\psi_B\rangle$, ao marcar os elementos que desejam ser buscados —negativando a amplitude dos estados correspondentes. Da mesma forma, o procedimento de amplificação de amplitude também causa uma reflexão, no entanto, esta ocorre em torno do vetor $|\psi_0\rangle$. O resultado obtido por essas duas reflexões é uma rotação no sentido anti-horário, como pode ser visto na Figura 2.2. [34].

Algumas conclusões importantes podem ser tiradas nessa abordagem. A rotação resultante da aplicação do procedimento de Grover permanece no espaço gerado por $|\psi_A\rangle$ e $|\psi_B\rangle$. O angulo dessa rotação é dado por $\theta = 2Arccos(\sqrt{(N-M)/N})$ e as mudanças dos estados nesta representação podem ser descritas como:

$$|\psi_0\rangle = \cos\left(\frac{\theta}{2}\right)|\psi_A\rangle + \sin\left(\frac{\theta}{2}\right)|\psi_B\rangle\,, \tag{2.2}$$

$$G\,|\psi_0\rangle = |\psi_1\rangle = \cos\left(\frac{3\theta}{2}\right)|\psi_A\rangle + \sin\left(\frac{3\theta}{2}\right)|\psi_B\rangle\,, \tag{2.3}$$

$$G^k\,|\psi_k\rangle = \cos\left(\frac{3k+1}{2}\theta\right)|\psi_A\rangle + \sin\left(\frac{3k+1}{2}\theta\right)|\psi_B\rangle\,. \tag{2.4}$$

O algoritmo de busca executa a iteração de Grover $\pi/4\sqrt{M/N}$ vezes. Quando a dimensão do domínio de $\boldsymbol{f}$ é uma potência de dois, a superposição de todas entradas possíveis pode ser feita aplicando a porta Hadamard bit-a-bit, na etapa de preparação dos estados. Caso não seja, deve-se usar a Transformada de Fourier Quântica, descrita na Seção 2.3, como instruído por Lomont [28].

A visualização geométrica da iteração de Grover nos permite entender melhor o seu funcionamento. Entretanto, uma outra abordagem representa este procedimento através da estimação de autovalores. Esta abordagem nos interessa pois os autovalores contém informações úteis para o Algoritmo Quântico de Contagem, e consequentemente o algoritmo proposto neste trabalho. Para uma melhor visualização, adotamos a notação $\sum'$ como a soma de todos os estados que não são soluções para o problema de busca, e $\sum''$ indicando a soma de todos os estados que são soluções para o problema.

Para os casos em que $M = 0$ ou $M = N$, $H\,|0\rangle$ é um autovetor de G com autovalor $e^{\pi i t/N}$ [33]. Para os casos em que $0 < M < N$, defina os estados:

$$H\,|0\rangle = |\Psi\rangle = \frac{1}{M}\sum_{x=0}^{n-1}|x\rangle\,, \tag{2.5}$$

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}}\sum{}'\,|x\rangle\,, \tag{2.6}$$

$$|\beta\rangle = \frac{1}{\sqrt{M}}\sum{}''\,|x\rangle\,, \tag{2.7}$$

$$|\Psi\rangle = \sqrt{\frac{N-M}{N}}\,|\alpha\rangle + \sqrt{\frac{M}{N}}\,|\beta\rangle\,, \tag{2.8}$$

$$|\Psi_+\rangle = \frac{1}{\sqrt{2}}\,|\beta\rangle + \frac{i}{\sqrt{2}}\,|\alpha\rangle\,, \tag{2.9}$$

$$|\Psi_-\rangle = \frac{1}{\sqrt{2}}\,|\beta\rangle - \frac{i}{\sqrt{2}}\,|\alpha\rangle\,. \tag{2.10}$$

Relembrando, o operador da iteração de Grover pode ser descrito como a concatenação $G = (-HU_0H)U_g$, onde o operador U_0 mapeia o estado $|0\rangle$ para $-|0\rangle$ e deixa os $|x\rangle$ restantes inalterados. Após uma pequena manipulação algébrica, é possível provar que o operador pode ser reescrito como $G = (2|\Psi\rangle\langle\Psi| - I_N)U_f$, onde I_N é a matriz Identidade de dimensão (NxN) e $|\Psi\rangle$ uma superposição uniforme de estados pesados pela mesma amplitude [34].

Para $0 < M < N$, $|\Psi_+\rangle$ e $|\Psi_-\rangle$ são autovetores de G com os autovalores $e^{2\pi i\omega}$ e $e^{-2\pi i\omega}$ | $0 < \omega < 1/2$, respectivamente. Por fim, temos que $\cos(2\pi\omega) = 1 - 2M/N$ e $\sin(2\pi\omega) = 2\sqrt{M(N-M)}/N$.

Aplicando G aos estados de elementos não desejados $|\alpha\rangle$, e desejados $|\beta\rangle$, obtemos os estados:

$$G|\alpha\rangle = -|\alpha\rangle + 2\sqrt{\frac{N-M}{N}}|\Psi\rangle, \tag{2.11}$$

$$G|\beta\rangle = |\beta\rangle - 2\sqrt{\frac{M}{N}}|\Psi\rangle. \tag{2.12}$$

Manipulando algebricamente os estados definidos previamente, encontramos o resultado da aplicação de G ao seu autovetor $|\Psi_+\rangle$:

$$G|\Psi_+\rangle = G\left(\frac{i}{\sqrt{2}}|\alpha\rangle\right) + G\left(\frac{1}{\sqrt{2}}|\beta\rangle\right)$$

$$G|\Psi_+\rangle = \frac{i}{\sqrt{2}}\left(-|\alpha\rangle + 2\sqrt{\frac{N-M}{N}}|\Psi\rangle\right) + \frac{1}{\sqrt{2}}\left(|\beta\rangle - 2\sqrt{\frac{M}{N}}|\Psi\rangle\right)$$

$$\vdots$$

$$G|\Psi_+\rangle = \frac{1}{\sqrt{2}}\left[\left(1-\frac{2M}{N}\right)i - \frac{2\sqrt{M(N-M)}}{N}\right]|\alpha\rangle + \frac{1}{\sqrt{2}}\left[\left(1-\frac{2M}{N}\right) + \frac{2\sqrt{M(N-M)}}{N}i\right]|\beta\rangle$$

e de forma análoga, temos que:

$$G|\Psi_-\rangle = G\left(-\frac{i}{\sqrt{2}}|\alpha\rangle\right) + G\left(\frac{1}{\sqrt{2}}|\beta\rangle\right)$$

$$G|\Psi_-\rangle = -\frac{i}{\sqrt{2}}\left(-|\alpha\rangle + 2\sqrt{\frac{N-M}{N}}|\Psi\rangle\right) + \frac{1}{\sqrt{2}}\left(|\beta\rangle - 2\sqrt{\frac{M}{N}}|\Psi\rangle\right)$$

$$\vdots$$

$$G|\Psi_-\rangle = \frac{1}{\sqrt{2}}\left[\left(-1+\frac{2M}{N}\right)i - \frac{2\sqrt{M(N-M)}}{N}\right]|\alpha\rangle + \frac{1}{\sqrt{2}}\left[\left(1-\frac{2M}{N}\right) - \frac{2\sqrt{M(N-M)}}{N}i\right]|\beta\rangle$$

Por conseguinte, temos que a aplicação dos autovalores $e^{2\pi i\omega}$ e $e^{-2\pi i\omega}$ aos autovetores

de G, $|\Psi_+\rangle$ e $|\Psi_-\rangle$ respectivamente, resulta em:

$$G\,|\Psi_+\rangle = e^{2\pi\omega}\,|\Psi_+\rangle = \cos(2\pi\omega)\,|\Psi_+\rangle + i\sin(2\pi\omega)\,|\Psi_+\rangle$$
$$e^{2\pi i\omega}\,|\Psi_+\rangle = \left(1-\frac{2M}{N}\right)|\Psi_+\rangle\,g + i\left(\frac{2\sqrt{M(N-M)}}{N}\right)|\Psi_+\rangle$$
$$e^{2\pi i\omega}\,|\Psi_+\rangle = \frac{1}{\sqrt{2}}\left(1-\frac{2M}{N}\right)(i\,|\alpha\rangle + |\beta\rangle) + \frac{i}{\sqrt{2}}\left(\frac{2\sqrt{M(N-M)}}{N}\right)(i\,|\alpha\rangle + |\beta\rangle)$$
$$\vdots$$
$$e^{2\pi i\omega}\,|\Psi_+\rangle = \frac{1}{\sqrt{2}}\left[\left(1-\frac{2M}{N}\right)i - \frac{2\sqrt{M(N-M)}}{N}\right]|\alpha\rangle + \frac{1}{\sqrt{2}}\left[\left(1-\frac{2M}{N}\right) + \frac{2\sqrt{M(N-M)}}{N}i\right]|\beta\rangle$$

e de forma análoga, temos que:

$$G\,|\Psi_-\rangle = e^{-2\pi i\omega}\,|\Psi_-\rangle = \cos(2\pi\omega)\,|\Psi_-\rangle - i\sin(2\pi\omega)\,|\Psi_-\rangle$$
$$e^{-2\pi i\omega}\,|\Psi_-\rangle = \left(1-\frac{2M}{N}\right)|\Psi_-\rangle - i\left(\frac{2\sqrt{M(N-M)}}{N}\right)|\Psi_-\rangle$$
$$e^{-2\pi i\omega}\,|\Psi_-\rangle = \frac{1}{\sqrt{2}}\left(1-\frac{2M}{N}\right)(-i\,|\alpha\rangle + |\beta\rangle) - \frac{i}{\sqrt{2}}\left(\frac{2\sqrt{M(N-M)}}{N}\right)(-i\,|\alpha\rangle + |\beta\rangle)$$
$$\vdots$$
$$e^{-2\pi i\omega}|\Psi_-\rangle = \frac{1}{\sqrt{2}}\left[\left(-1+\frac{2M}{N}\right)i - \frac{2\sqrt{M(N-M)}}{N}\right]|\alpha\rangle + \frac{1}{\sqrt{2}}\left[\left(1-\frac{2M}{N}\right) - \frac{2\sqrt{M(N-M)}}{N}i\right]|\beta\rangle$$

Nota-se que não é possível preparar os auto-estados $|\Psi_-\rangle$ e $|\Psi_+\rangle$. Esse fato, aparentemente inutiliza os seus respectivos autovalores, ao qual encontramos. Contudo, nós sabemos que o estado inicial $H\,|0\rangle^{\otimes n}$ pode ser representado na base dos autovetores como:

$$|\Psi\rangle = \frac{-i}{\sqrt{2}}e^{\pi i\omega}\,|\Psi_+\rangle + \frac{i}{\sqrt{2}}e^{-\pi i\omega}\,|\Psi_-\rangle\,, \tag{2.13}$$

e que, aplicar G^k a este estado inicial equivale a:

$$G^k\,|\Psi\rangle = \frac{-i}{\sqrt{2}}e^{(2k+1)\pi i\omega}\,|\Psi_+\rangle + \frac{i}{\sqrt{2}}e^{-(2k+1)\pi i\omega}\,|\Psi_-\rangle\,. \tag{2.14}$$

Quando temos M estritamente entre 0 e N, obtemos a seguinte equivalência de estados:

$$\frac{1}{\sqrt{2}}\,|\Psi_+\rangle + \frac{1}{\sqrt{2}}\,|\Psi_-\rangle = |\beta\rangle\,. \tag{2.15}$$

Uma vez que o objetivo deste algoritmo é medir o estado $M = |\beta\rangle$, podemos utilizar esta peculiar equivalência para obter o resultado desejado. Para isto, é necessário alinhar as fases de forma que a fase relativa entre os dois autovetores seja próxima a zero na Equação (2.14). Nos casos em que M é conhecido, só é necessário escolher k de forma que $-e^{(2k+1)i\varphi/2} = e^{-(2k+1)i\varphi/2}$, ou que $(4k+2)(\varphi/2\pi)$ seja um inteiro ímpar [33]. Outras configurações, de forma a obter este estado, podem ser encontradas no artigo por Mosca [33].

As abordagens apresentadas são probabilísticas com probabilidade de pelo menos 50% de observar a solução para o problema. Entretanto, uma abordagem determinística desse algoritmo de busca pode ser encontrada no artigo por Mosca [33].

2.3 Transformada de Fourier Quântica

Primeiramente, é importante destacar que a finalidade última deste capítulo é o entendimento do procedimento de Estimativa de Fase. Entretanto, para isto é necessário ter um entendimento da Transformada de Fourier Quântica.

A Transformação de Fourier Quântica (TFQ) é, como o nome já implica, a versão quântica da transformada de Fourier discreta. Não existe uma diferença, em termos de velocidade de cálculo, entre a transformada no modelo computacional clássico e quântico [34], entretanto ela possui uma tarefa importante chamada Estimativa de Fase, detalhada na Seção 2.3.1. Este procedimento se destaca por ter um papel fundamental em algoritmos importantes da área —como por exemplo o Algoritmo de Shor para Fatoração Quântica [39], o Algoritmo Quântico de Contagem [9] e o Algoritmo de criptoanálise [8].

O objetivo dessa seção é sintetizar, de uma forma mais intuitiva, as informações presentes no livro de Nielsen e Chuang [34], especificamente o capítulo dedicado a Transformada de Fourier Quântica, e as notas de aula de Birgitta Whaley [43]. Portanto, para um maior aprofundamento no assunto ou detalhamento das manipulações algébricas omitidas neste texto, recomenda-se esta bibliografia.

A Transformada de Fourier Clássica —ou Transformada de Fourier Discreta— tem na sua notação usual, um vetor de números complexos $X = (x_0, x_1, ..., x_{N-1})$ de tamanho fixo como entrada. E como saída, o vetor X transformado em um outro vetor de números

complexos $(y_0, y_1, ..., y_{N-1})$. Definido por:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi ijk/N}. \tag{2.16}$$

Como mencionado acima, a TFQ realiza a Transformada de Fourier Discreta, com uma notação diferente. Atuando sobre uma base ortonormal $|0\rangle, |1\rangle, ..., |N-1\rangle$, sua transformação sobre um estado arbitrário é dada por:

$$\sum_{j=0}^{N-1} x_j |j\rangle \rightarrow \sum_{k=0}^{N-1} \left(\sum_{j=0}^{N-1} x_j e^{2\pi ijk/N} \right) |k\rangle . \tag{2.17}$$

Definindo $N = 2^n$, onde n é o número de q-bits usados no modelo quântico com uma base computacional $|0\rangle, |1\rangle, ..., |N-1\rangle$. Podemos escrever j usando a representação binária $j = j_1 j_2 ... j_n = j_1 2^{n-1} + j_2 2^{n-2} + ... + j_n 2^0$. Através de uma manipulação algébrica, temos a TFQ na forma conhecida como *representação de produto*:

$$|j_1 ... j_n\rangle \rightarrow \frac{(|0\rangle + e^{2\pi ij/2^1} |1\rangle)(|0\rangle + e^{2\pi ij/2^2} |1\rangle)...(|0\rangle + e^{2\pi ij/2^n |1\rangle})}{\sqrt{N}}, \tag{2.18}$$

onde n corresponde ao número de q-bits do registrador $|j\rangle$, alvo da TFQ†. Esta representação facilita a construção do circuito da TFQ, que pode ser encontrado na Figura 2.3. A porta R_k, necessária para a construção de seu circuito, é representada pela matriz unitária:

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix} \tag{2.19}$$

Nota-se que todas as portas utilizadas na construção do circuito da TFQ são unitárias, e portanto temos que a própria transformação é unitária.

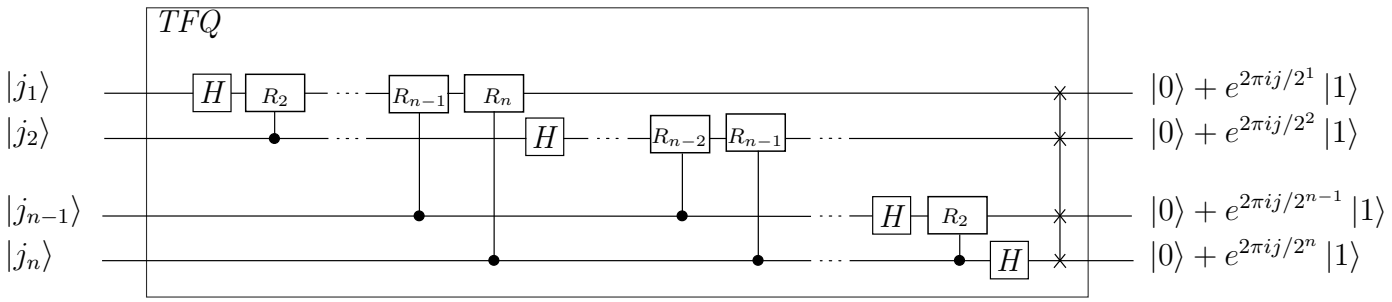

Figura 2.3: Circuito da Transformada de Fourier Quântica

2.3.1 Estimativa de Fase

Como dito anteriormente, a TFQ é o núcleo de um procedimento chamado *estimativa de fase* que, por sua vez, é um procedimento base em muitos algoritmos quânticos [8, 9, 11, 39]. Suponha que você tenha um operador U com um autovetor $|u\rangle$ e um autovalor correspondente $e^{2\pi i\omega}$, com $0 \leq \omega < 1$, o objetivo da estimativa de fase é estimar ω.

Uma das características da estimativa de fase é a necessidade de uma 'caixa-preta', ou seja, um operador ou uma concatenação de operadores que realiza uma operação controlada U^{2^k}, com k positivo, sobre um estado $|u\rangle$ preparado. Isso nos indica que esse procedimento não é um algoritmo por si só, mas um módulo a ser usado com outros procedimentos para realizar uma tarefa.

Os operadores controlados U^{2^k}, como dito anteriormente, são tratados nessa seção como caixas-pretas. Portanto, ao contrário das aplicações específicas da estimativa de fase, não entraremos em detalhe quanto à construção destes operadores. O circuito quântico do procedimento de estimativa de fase usa a porta de Hadamard H, operadores controlados U^{2^k} e a TFQ inversa.

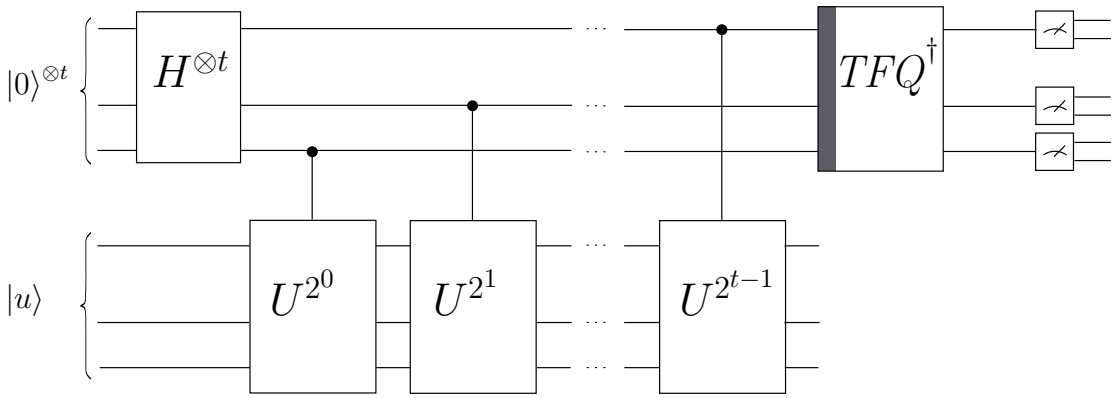

Figura 2.4: Circuito do procedimento de Estimativa de Fase

Como representado na Figura 2.4, o procedimento para a estimativa de fase é composto por dois registradores, no qual o número t de q-bits do primeiro registrador está diretamente relacionado com a precisão da estimativa de ω, e da probabilidade da estimativa correta.

Inicialmente, nós temos a preparação dos estados do primeiro registrador, no qual o operador de Hadamard é aplicado bit-a-bit, sem alterar o segundo registrador:

$$H^{\otimes t}|\mathbf{0}\rangle\,|u\rangle = \sum_{j=0}^{N-1}. \tag{2.20}$$

Posteriormente, para cada q-bit do primeiro registrador, realiza-se uma operação controlada sequencial no segundo registrador. Esta ação em um estado $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\,|u\rangle$ resulta em

$$\begin{aligned}\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\,|u\rangle &\xrightarrow{U^{2^k}} \frac{1}{\sqrt{2}}(|0\rangle + U^{2^k}\,|1\rangle)\,|u\rangle \\ &= \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\omega 2^k}\,|1\rangle)\,|u\rangle\,.\end{aligned} \tag{2.21}$$

Logo, o estado do sistema antes da $TFQ^{\dagger}$ pode ser descrito como:

$$\begin{aligned}\frac{1}{\sqrt{2^t}}\Big((|0\rangle + e^{2\pi i\omega 2^{t-1}}\,|1\rangle)(|0\rangle + e^{2\pi i\omega 2^{t-2}}\,|1\rangle)...(|0\rangle + e^{2\pi i\omega 2}\,|1\rangle)(|0\rangle + e^{2\pi i\omega 2^0}\,|1\rangle)\Big) \otimes |u\rangle \\ = \frac{1}{2^t}\sum_{y=0}^{2^t-1} e^{2\pi i\omega y}\,|y\rangle\,|u\rangle\end{aligned} \tag{2.22}$$

Através das equações anteriores, podemos perceber os fatores de fase $e^{2\pi i\omega y}$ sendo propagados do segundo registrador, do autovetor, para o primeiro, o registrador controlador. Nota-se também que, a fase ω, pode ser descrita como:

$$\omega = \frac{a}{2^t} + \delta, \tag{2.23}$$

onde temos $a = a_1a_2...a_t$ na notação binária, e uma margem de erro $0 \leq \delta \leq 1/2^{t+1}$. Seguindo o circuito, ao aplicar a TFQ† ao estado da Equação (2.22), obtemos a transformação

$$\frac{1}{2^t}\sum_{y=0}^{2^t-1} e^{2\pi i\omega y}\,|y\rangle\,|u\rangle \xrightarrow{TFQ^{\dagger}} |\tilde{\omega}\rangle\,|u\rangle\,, \tag{2.24}$$

onde o estado $|\tilde{\omega}\rangle$ é uma superposição na qual as amplitudes estão concentradas perto do valor de a. Por fim, temos que $\tilde{\omega} = (a/2^t)$, a melhor aproximação binária de ω com t q-bits.

Inicialmente, em seu artigo, Cleve *et al.* [13] provaram que a probabilidade de leitura de $\tilde{\omega}$ é $\rho(\tilde{\omega}) \geq 4/\pi^2 \approx 0.405$. Entretanto, no mesmo artigo, uma configuração do procedimento de estimativa de fase com probabilidade de sucesso amplificada é apresentada. Nesta configuração, se define o número de q-bits do primeiro registrador como $t \equiv m + \lceil \log(1/2 + 1/2\gamma) \rceil$. De forma que medindo estes q-bits ao fim do circuito, a solução encontrada tem probabilidade $1 - \gamma$ de ter a precisão $\Delta\omega = \omega - \tilde{\omega} \leq 2^{-m}$.

2.4 Algoritmo Quântico de Contagem

O objetivo do algoritmo de Contagem é encontrar a frequência M, de um dado valor k em uma entrada $x_0, x_1, ..., x_{N-1}$. A solução encontrada por Boyer *et al.* [9] utiliza o procedimento de estimativa de fase, estratégia parecida com a utilizada por Shor em seu algoritmo para Fatoração [39], encontrando uma aproximação de M a partir da estimativa de φ. Baseando-se nas definições estabelecidas neste capítulo, definimos o Algoritmo Quântico de Contagem por autovalor a seguir. Primeiramente, iniciamos o sistema no estado

$$|\Psi\rangle = H^{\otimes t} |\mathbf{0}\rangle H^{\otimes n} |\mathbf{0}\rangle , \tag{2.25}$$

que é equivalente a

$$\begin{aligned} |\Psi\rangle &= \sum_{j=0}^{2^t-1} |j\rangle \left(\frac{-i}{\sqrt{2}} e^{i\varphi/2} |\Psi_+\rangle + \frac{i}{\sqrt{2}} e^{-i\varphi/2} |\Psi_-\rangle \right) \\ &= \frac{-ie^{i\varphi/2}}{\sqrt{2}} \sum_{j=0}^{2^t-1} |j\rangle |\Psi_+\rangle + \frac{ie^{-i\varphi/2}}{\sqrt{2}} \sum_{j=0} |j\rangle |\Psi_-\rangle . \end{aligned} \tag{2.26}$$

Aplicando os operadores controlados $U^{2^k} = G^{2^k}$, onde G é o procedimento de Grover, ao estado inicial, obtemos o estado anterior à aplicação da TFQ inversa

$$\begin{aligned} |\Psi_1\rangle = &\frac{-ie^{i\varphi/2}}{\sqrt{2}} \left[((|0\rangle + e^{i\varphi 2^{t-1}} |1\rangle)...(|0\rangle + e^{i\varphi 2^0} |1\rangle)) |\Psi_+\rangle \right] \\ &+ \frac{ie^{-i\varphi/2}}{\sqrt{2}} \left[((|0\rangle + e^{-i\varphi 2^{t-1}} |1\rangle)...(|0\rangle + e^{-i\varphi 2^0} |1\rangle)) |\Psi_-\rangle \right] . \end{aligned} \tag{2.27}$$

Em seguida, aplicando a TFQ inversa sobre o primeiro registrador, com t q-bits, obtemos o estado

$$|\Psi_2\rangle = \frac{-ie^{i\varphi/2}}{\sqrt{2}} |\tilde{\varphi}\rangle |\Psi_+\rangle + \frac{ie^{-i\varphi/2}}{\sqrt{2}} |-\tilde{\varphi}\rangle |\Psi_-\rangle . \tag{2.28}$$

Os estados $|\tilde{\varphi}\rangle$ e $|-\tilde{\varphi}\rangle$ são superposições com amplitudes concentradas perto de $\tilde{\varphi}$ e $-\tilde{\varphi}$, respectivamente. Uma vez que a estimativa de $-\tilde{\varphi}$ é claramente equivalente a estimativa de $-\tilde{\varphi}$, tanto em probabilidade como em precisão [34], consideraremos somente o primeiro. Portanto, no caso geral existe uma probabilidade da fase estimada $\tilde{\varphi} \equiv -\tilde{\varphi}$ não ser medida.

Como vimos anteriormente, o Algoritmo Quântico de Contagem foi construído de forma que a partir do número de aplicações da iteração de Grover (i.e., 2^t), é possível definir a probabilidade de leitura de uma fase $\tilde{\omega}$ e um limite sua precisão. Que ,por sua

vez, influenciam diretamente e respectivamente na probabilidade e precisão da frequência $\tilde{M}$ resultante.

O Algoritmo Quântico de Contagem, usado neste trabalho, utiliza uma destas definições propostas por Mosca em seu artigo [33]. Esta proposição garante que, aplicando $\lceil c\sqrt{N} \rceil$ vezes a iteração de Grover ,i.e. $2^t = \lceil c\sqrt{N} \rceil$, tem-se uma probabilidade de pelo menos 2/3 de que a margem de erro do resultado seja

$$| \tilde{M} - M |= \frac{1}{4^2} + \frac{\sqrt{\min(M, N-M)}}{c} \in O\left(\frac{\sqrt{\min(M, N-M)}}{c}\right) \tag{2.29}$$

Um outro resultado interessante, alcançado por Mosca em seu artigo, é a prova de uma proposição tal que, sendo G como definido anteriormente, é possível medir a fase ω e determinar o M correto com probabilidade maior ou igual a 2/3. De forma que o número de aplicações de G está na ordem de $O(\sqrt{(t+1)(N-t+1)})$.

2.5 Problema *k-Distinctness*

Em granjas de suínos, que tem como objetivo a criação e venda de porcos, existe um problema de logística referente à venda e transporte dos animais para o abatedouro. Com o objetivo de minimizar o custo de transporte deste animais para os abatedouros, os produtores de suínos buscam vender caminhões cheios de porcos, na mesma faixa de peso, para os abatedouros. Portanto, sempre antes de uma venda, é necessário saber se dentre todos os suínos da granja, existem pelo menos k animais na mesma faixa de peso, de forma a encher um ou mais caminhões. Este problema apresentado, é um dos inúmeros problemas específicos do problema *Element k-Distinctness*.

Dada uma lista não-ordenada $(x_1, x_2, \ldots, x_N)$, com $1 \leq x_v \leq M$ para cada índice $1 \leq v \leq N$, definimos *Element k-Distinctness* (ou simplesmente *k-distinctness*) como o problema de decidir se esta lista contém pelo menos k elementos repetidos, i.e, se existe um conjunto de índices $1 \leq a_1, a_2, \ldots, a_k \leq N$ tal que $a_v \neq a_j$ e $x_{a_v} = x_{a_j}$ para todo $v \neq j$. Quando $k = 2$, referimos ao problema simplesmente como *Element Distinctness*. A complexidade clássica para solucionar o problema *Element k-Distinctness* é $\Theta(N \log(N/k))$ [30].

Em 2003, Ambainis [3] propôs um algoritmo quântico para solucionar este problema — baseando-se na ferramenta de caminhada quântica— com complexidade de tempo $O(N^{k/k+1})$, alcançando o limite inferior para o problema *Element Distinctness*. Este algoritmo foi re-

visado por Portugal [36] em 2017, de forma a analisar detalhes do algoritmo original e propor uma nova evolução, em um espaço de Hilbert diferente. Por sua vez, em 2012, Belovs [5] introduziu uma técnica de *learning graphs* quântico, e introduziu um algoritmo para este problema com complexidade de consulta $O(N^{1-2^{k-2}/(2^k-1)})$, e probabilidade de sucesso de pelo menos 50%. Através de uma manipulação algébrica, obtemos que a complexidade do algoritmo de Belovs $O(N^{1-2^{k-2}/(2^k-1)}) = O(N^{3/4-1/(4(2^k-1))})$. Usando a ferramenta matemática de *limite* nesta função ,com $k \to \infty$, conclui-se que independente de k, a complexidade do algoritmo é $o(N^{3/4})$.

Capítulo 3

Algoritmo para Encontrar a Moda

Nesta seção, iremos apresentar um algoritmo final composto por duas abordagens quânticas, para encontrar o valor modal. A primeira abordagem (veja Algoritmo 2) possui uma complexidade de consulta $o(N^{3/4} \log N)$, utilizando o algoritmo de *k-Distinctness* de Belovs em conjunto com um procedimento similar à busca binária como base. A segunda abordagem possui uma complexidade de consulta $O(r\sqrt{N})$, e utiliza o Algoritmo Quântico de Contagem como base. Por fim, o algoritmo final é uma combinação da primeira e da segunda abordagem, calculando a faixa de dados r, e selecionando o melhor caso de execução de cada uma destas, resultando em uma complexidade de consulta $\min(o(N^{3/4} \log N), O(r\sqrt{N}))$.

3.1 Primeira abordagem: EncValModalA

A primeira abordagem, *EncValModalA* (veja Algoritmo 2), tem como primeiro passo encontrar a frequência modal a partir do método *EncFreqModal* (veja Seção 3.1.1), que recebe como parâmetro a lista de entrada L. Em seguida, através da combinação de uma busca binária com o algoritmo de Belovs [5], para o problema *Element k-Distinctness*, um valor modal x_F é encontrado.

Dada uma lista não-ordenada $L \equiv (x_1, x_2, \ldots, x_N)$ como entrada, essa abordagem primeiramente encontra a frequência modal —frequência do valor modal—, utilizando o método *EncFreqModal* (veja Seção 3.1.1). A partir da frequência modal, um procedimento similar à busca binária em conjunto com o algoritmo de Belovs [5], para o problema *Element k-Distinctness*, é executado para encontrar um valor modal x_F. O valor modal x_F é caracterizado pelas seguintes propriedades:

i) x_F é um valor modal.

ii) Não há nenhum elemento com o mesmo valor que x_F em uma posição mais alta —ou seja, com um índice maior— na lista.

iii) x_F é o primeiro valor modal na lista que satisfaz as propriedades anteriores.

Por exemplo, dada uma lista $L \equiv (x_1, x_2, \ldots, x_9) \equiv (1,\ 1,\ 2,\ 1,\ 2,\ 3,\ 2,\ 2,\ 1)$, então cada elemento na sublista $L_1 \equiv (x_1, x_2, x_3, x_4, x_5, x_7, x_8, x_9)$ satisfaz a primeira propriedade —x_6 é o único elemento neste exemplo que não é um valor modal em L. Os elementos na sublista $L_2 \equiv (x_8, x_9)$ também satisfazem a segunda propriedade, uma vez que eles são os últimos elementos na lista original representando cada valor modal. Finalmente, o elemento x_8 também satisfaz a terceira propriedade, uma vez que é o primeiro elemento na sublista anterior L_2. Portanto, o valor modal x_F neste exemplo é $x_8 \equiv 2$. Vale notar que uma entrada pode conter mais de um elemento com frequência igual à frequência modal.

Definindo *lower* e *upper* como a faixa de possíveis posições para x_F em L, respectivamente, e considerando que no começo da iteração de busca —similar à busca binária—, a posição de x_F é totalmente desconhecida, temos que $lower = 1$ e $upper = N$.

Durante o laço de busca, o algoritmo de *k-Distinctness* verifica se $x_1 \leq x_F \leq x_{mid}$, onde o valor do meio $mid \equiv (lower + upper)/2$. Se o algoritmo de *k-Distinctness* retorna *verdadeiro*, então o limite superior é reduzido ao valor do meio. Caso contrário, então o limite inferior é aumentado ao valor do meio. Quando a diferença entre a limite superior e inferior for igual a um, x_F é encontrado. O parâmetro f é a frequência modal, e corresponde ao parâmetro k do algoritmo para solucionar o problema *k-Distinctness*.

Vale lembrar que, o algoritmo para solucionar o problema *k-Distinctness* utilizado tanto nesta abordagem quanto no procedimento 3 é probabilístico. Isso implica que no momento em que este algoritmo retornar uma solução errada, um falso positivo ou falso negativo, as soluções desta sub-rotina e do procedimento 3 se "corrompem", i.e., a resposta final certamente será a errada.

Nós propomos duas soluções para reduzir a probabilidade desse erro. A primeira solução é aplicar o algoritmo para *k-Distinctness* um número constante de vezes, de forma a não alterar a complexidade final, sempre que a solução para o mesmo for necessária, utilizando o resultado mais recorrente.

A segunda opção consiste em reverter alguns passos e continuar a execução do algo-

ritmo. Essa segunda opção foi planejada depois de percebermos que em muitos casos o primeiro falso negativo ou falso positivo provoca uma cascata de negações, de forma que a resposta final do algoritmo continua a mesma deste o primeiro erro. A ideia por traz desta opção é armazenar os dados da última alteração na resposta temporária da busca binária, e ao fim do algoritmo, executá-lo novamente a partir daquele ponto um número constante de vezes.

Dado um procedimento genérico composto por γ passos, onde cada passo $0 < j \leq \gamma$ possui uma complexidade de consulta c_j, temos que a complexidade de consulta final deste procedimento, i.e. o número de consultas realizadas aos dados armazenados, é da ordem de $\sum_y c_j$, levando em consideração estruturas de repetições, recursivas etc. A partir disto, provamos as proposições a seguir.

Proposição 1. *O algoritmo 2 possui complexidade de consulta* $o(N^{3/4} \log N)$.

Demonstração. O primeiro passo da sub-rotina 2 —executar o método $EncFreqModal(L)$— possui complexidade $c_1 = o(N^{3/4} \log N)$. Por serem operações elementares de consulta, os passos 2,3 e 12 possuem complexidades $c_2 = c_3 = c_{12} = O(1)$. A estrutura de repetição *enquanto*, que age sobre os passos 4 à 11, comporta-se como uma busca binária, que no pior caso execução os passos dentro do seu bloco $\log N$ vezes. Multiplicando a complexidade de cada passo (passos 5 a 11) dentro da estrutura pelo número de repetições de pior caso, obtemos a complexidade total deste bloco de operações. A estrutura condicional *se-senão* entre os passos 8 e 11 e o passo 5 realizam somente operações elementares, somando uma complexidade na ordem de $O(1)$ consultas. Enquanto o passo 6 possui complexidade já determinado de $o(N^{3/4})$ [6]. Finalmente, temos que a complexidade de consultas final deste procedimento é de $c = o(N^{3/4} \log N) + O(1) + \sum^{\log N} \big(o(N^{3/4}) + O(1)\big) = o(N^{3/4} \log N)$. □

3.1.1 Procedimento EncFreqModal

Note que, a primeira abordagem (veja Algoritmo 2) requer o cálculo da frequência modal, que é obtida através do procedimento EncFreqModal (veja Procedimento 3). Esse procedimento consiste em uma busca binária da frequência modal F sobre uma faixa de possíveis frequências, com $O(\log N)$ execuções do algoritmo de *k-Distinctness*.

A faixa de possíveis frequências é delimitada por um limite inferior inf e superior sup. Durante o laço da busca binária, o valor do meio $k \equiv (inf + sup)/2$ é recalculado, e

Algoritmo 2 EncValModalA: Algoritmo para encontrar o valor modal quando $r \geq \sqrt[4]{N} \log N$.

Entrada: Uma lista não-ordenada $L \in \{1, \ldots, M\}^N$
Saída: O Valor Modal de L

1: $f \leftarrow$ EncFreqModal(L)
2: $lower \leftarrow 1$
3: $upper \leftarrow N$
4: **enquanto** $lower \leq upper - 1$ **faça**
5: $\quad mid \leftarrow (lower + upper)/2$
6: $\quad found \leftarrow$ KDistinctness(L, f, mid)
7: $\quad$ {chamada anterior retorna **verdadeiro** se e somente se encontra o elemento com frequência pelo menos f}
8: $\quad$ **se** found **então**
9: $\quad\quad upper \leftarrow mid$
10: $\quad$ **senão**
11: $\quad\quad lower \leftarrow mid$
12: **retorna** $upper$

então o algoritmo de *k-Distinctness* verifica se existe um elemento ocorrendo pelo menos k vezes na lista L. Se o algoritmo de *k-Distinctness* retorna *verdadeiro*, então o limite inferior é aumentado ao valor do meio k. Em contra-partida, se o algoritmo retorna *falso*, então o limite superior é reduzido ao valor do meio k. Quando a diferença entre o limite superior e inferior for igual a um, a frequência modal f é encontrada. Vale lembrar, que este procedimento também é probabilístico, em consequência do uso do algoritmo probabilístico de *k-Distinctness*. Métodos para aumentar esta probabilidade foram discutidos na Seção 3.1.

Proposição 2. *O Procedimento 3, para encontrar a frequência modal, tem complexidade de consulta* $o(N^{3/4} \log N)$.

Demonstração. Os passos 1 e 2, e a estrutura condicional nos passos 12 e 13 realizam operações elementares, com complexidades $c_1 = c_2 = c_{12} = c_{13} = O(1)$. O passo 11 chama um algoritmo de *k-Distinctness*; usando o algoritmo de Belovs [5], este passo possui complexidade de consulta $o(N^{3/4})$. A estrutura de repetição *enquanto* —que compreende os passos 3 à 10— possui um funcionamento similar ao procedimento de busca binária, de forma que no pior caso, cada passo dentro do bloco é executado $\log N$ vezes e esta complexidade levada em conta. Os passos 4 e 6 à 10, incluindo as complexidades de checagem da estrutura de condição *se-senão*, também possuem complexidade $O(1)$. O passo 11, assim como o passo 5 possui complexidade de execução do algoritmo para o problema *k-Distinctness*. Finalmente, temos que a complexidade de consulta final deste procedimento é $c = O(1) + \sum^{\log N} \big(O(1) + O(N^{3/4} \log N)\big) = o(N^{3/4} \log N)$. $\square$

Procedimento 3 EncFreqModal: Procedimento para encontrar a frequência modal

Entrada: Uma lista não-ordenada $L \in \{1, \ldots, M\}^N$
Saída: A Frequência Modal de L

1: $inf \leftarrow 1$
2: $sup \leftarrow N$
3: **enquanto** $inf \leq sup - 1$ **faça**
4: $k \leftarrow \lceil (inf + sup)/2 \rceil$
5: $found \leftarrow$ KDistinctness(L, k, N) {retorna **verdadeiro** sse encontra um elemento com frequência maior ou igual a k}
6: **se** $found$ **então**
7: $inf \leftarrow k$
8: $f \leftarrow inf$
9: **senão**
10: $sup \leftarrow k - 1$
11: $found \leftarrow$ KDistinctness(L, sup, N) {**verdadeiro** sse encontra um elemento com frequência maior ou igual a sup}
12: **se** $found$ **então**
13: $f \leftarrow sup$
14: **retorna** f

3.2 Segunda abordagem: EncValModalB

A segunda abordagem, EncValModalB (veja Algoritmo 4), é composta por uma estrutura de repetição *for* com r iterações, cada uma executando o Algoritmo Quântico de Contagem um número constante de vezes, omitindo-se o pseudo-algoritmo, de forma que cada um dos r elementos na variação $[min, max]$ tem sua frequência contada. Por fim, durante este mesmo laço, usando um simples procedimento de comparação, a maior frequência calculada até o momento e seu elemento correspondente são armazenados. Os elementos pertencentes à variação $[min, max]$ que não ocorrem na lista de entrada podem ser contados com precisão 1, de modo que o resultado do algoritmo não é influenciado pelas "falhas" da variação contada.

Note que, o Algoritmo Quântico de Contagem é aproximativo —quanto a frequência dos elementos— e probabilístico, consequentemente seus resultados também. Entretanto, a aproximação e a probabilidade podem ser limitadas usando as configurações apresentadas por Mosca [33], presentes na Seção 2.4.

Proposição 3. *O algoritmo 4 possui complexidade de consulta* $O(r\sqrt{N})$.

Demonstração. Todos os passos do algoritmo 4, com exceção do primeiro —que possui complexidade $O(1)$— estão estão dento do bloco da estrutura de repetição *para*, configu-

rado para realizar r iterações. Dentro deste bloco, o passo 5 a 7 possuem complexidades $c_5 = c_7 = O(1)$, enquanto o passo 3 possui um gasto $c_3 = O(\sqrt{N})$ —determinado pelo Algoritmo Quântico de Contagem (veja Seção 2.4). Portanto, a complexidade de consulta final deste algoritmo é $c = O(1) + r(O(1) + O(\sqrt{N}) = O(r\sqrt{N}))$. □

Algoritmo 4 EncValModalB: Algoritmo para encontrar o valor modal

Entrada: Uma lista não-ordenada $L \in \{1, \ldots, M\}^N$, seu menor elemento min e seu maior elemento max

Saída: O Valor Modal de L

1: $mfreq \leftarrow 0$
2: **para** $i \leftarrow min$ **to** max **faça**
3: $tmpfreq \leftarrow$ QuantumCounting(L, i)
4: {chamada anterior retorna a frequência do elemento i em L}
5: **se** $tmpfreq \geq mfreq$ **então**
6: $mfreq \leftarrow tmpfreq$
7: $m \leftarrow i$
8: **retorna** m

3.3 Algoritmo Final: EncValModalFinal

O Algoritmo Final, EncValModalFinal (veja Algoritmo 5), proposto para encontrar o valor modal, explora, se utilizando dos melhores casos de complexidade dos algoritmos apresentados anteriormente nas Seções 3.1 e 3.2. Isso se da a partir do calculo da faixa de dados r, ao qual previamente definimos como a diferença entre o maior e o menor número na lista de entrada, e depois da seleção de qual abordagem executar como subrotina. A fim de encontrar os valores de limite da faixa de dados, aplicamos uma versão ligeiramente modificada do algoritmo quântico KLPF [26], originalmente proposto para encontrar o menor valor de uma lista não-ordenada. Essa fase inicial do algoritmo requer $O(\sqrt{N})$ consultas, e como dito anteriormente, seu propósito é o de determinar qual dos algoritmos propostos é mais eficiente em termos de número de consultas, para cada entrada particular. Portanto, ao fim, o procedimento executa o algoritmo selecionado, como subrotina uma única vez.

Proposição 4. *O algoritmo 5 para encontrar o valor modal, tem complexidade de consulta* $\min(o(N^{3/4} \log N), O(r\sqrt{N}))$.

Demonstração. A partir do artigo de Kowada *et al* [26], conhecemos que as complexidades c_1 e c_2 do algoritmo para encontrar a moda é $O(\sqrt{N})$. O terceiro passo do algoritmo, o de checagem da estrutura condicional *se-senão*, é composto por operações elementares,

consequentemente com complexidades $c_4 = c_6 = O(1)$. Baseando-se nas Definições 1 e 3, as complexidades dos passos 5 e 7 são $c_5 = o(N^{3/4} \log N)$ e $c_7 = O(r\sqrt{N})$, respectivamente. Entretanto, devido à estrutura condicional mencionada acima, somente a sub-rotina com menor complexidade computacional, selecionada a partir de r, é executada, resultando em uma complexidade de consulta final $c = O(\sqrt{N}) + O(1) + \min(o(N^{3/4} \log N), O(r\sqrt{N})) = \min(o(N^{3/4} \log N), O(r\sqrt{N}))$. □

Algoritmo 5 Algoritmo Quântico Final para encontrar o Valor Modal

Entrada: Uma lista não-ordenada $L \in \{1, \ldots, M\}^N$
Saída: O valor modal de L
  $min \leftarrow$ EncMinimo(L) {usa o algoritmo quântico KLPF}
  $max \leftarrow$ EncMaximo(L) {idem}
  $r \leftarrow |max$ - $min|$
  se $r \geq \sqrt[4]{N} \log N$ **então**
    $m \leftarrow$ EncValModalA(L)
  senão
    $m \leftarrow$ EncValModalB(L, min, max)
  retorna m

Capítulo 4

Simulações

Atualmente, não existem computadores quânticos universais de porte prático —os computadores desenvolvidos pela D-Wave não fazem parte desta categoria—, de forma que os melhores existentes possuem por volta de 20 q-bits [12]. Logo, uma forma encontrada para estudar o desempenho dos algoritmos quânticos são as simulações feitas em computadores clássicos, não muito eficientes. Percebe-se esta ineficiência, por exemplo, na representação de informações, uma vez que um estado de um sistema quântico com n q-bits é um vetor normalizado no espaço de Hilbert de dimensão 2^n, sendo necessário armazenar a amplitude de cada elemento da base [25]. Esta quantidade exponencial de bits necessários para representar o estado de um sistema quântico em um sistema clássico é um obstáculo considerável, limitando o tamanho de instâncias de problemas, principalmente os caracterizados pela sua intratabilidade no modelo clássico, como por exemplo os problemas pertencentes ao conjunto **NP-Difícil**.

4.1 Descrição da simulação

O software Scilab é capaz de simular quase todas as ações e fenômenos físicos de um algoritmo quântico via álgebra linear. Os estados tornam-se vetores, os operadores matrizes, e as transformações e relações entre estados são simuladas através de operações algébricas. No entanto, a medição de um estado quântico não pode ser fielmente simulada, por envolver um fator de aleatoriedade verdadeiro. A solução para esse problema é a utilização de métodos pseudo-aleatórios para a reprodução desse procedimento.

Neste trabalho, optou-se por fazer a simulação (veja Simulação 6) somente da segunda sub-rotina (veja Algoritmo 4) do algoritmo proposto. A decisão de não simular a sub-rotina 2 se deu pela dificuldade de implementação do algoritmo de *k-Distinctness*, que

utiliza uma uma técnica original —denominada "learning graphs"— na construção do seu algoritmo e outros fatores diversos —como por exemplo, o uso de programação semi-definida— que não são o foco deste trabalho.

Simulação 6 Algoritmo 4

1: Definição do tamanho da entrada N
2: Definição da faixa de dados r
3: $c \leftarrow 1$
4: $t = \lceil \log_2(c\sqrt{N}) \rceil$
5: **para** $i = 0 \leq N$ **faça**
6: $\quad entrada[i] \leftarrow random()^* r$
7: **para todo** elemento $\in \{0, ..., r\}$ **faça**
8: $\quad$ Inicializar o sistema em $|0\rangle\,|u\rangle$
9: $\quad$ Fazer a superposição do primeiro registrador: $\mathrm{H}^{\otimes t}\,|\mathbf{0}\rangle \mathrm{I}^{\otimes n}\,|0\rangle$
10: $\quad$ **para** $i = 0 < t - 1$ **faça**
11: $\quad\quad$ Gere o operador C_G, operador da iteração de Grover controlado pelo q-bit i do primeiro registrador
12: $\quad\quad$ Aplique o operador C_G sobre o segundo registrador 2^i vezes
13: $\quad$ Aplique a $TFQ^\dagger$ sobre o primeiro registrador
14: $\quad$ Realize o processo de medição sobre o primeiro registrador três vezes
15: $\quad$ Armazene a maior frequência medida e seu elemento correspondente
16: **retorne:** o elemento com maior frequência

Os primeiros passos da simulação consistem na definição dos parâmetros de entrada da sub-rotina. A faixa de dados r, o tamanho $N = 2^n$ da entrada a ser trabalhada (utilizada para construção do oráculo no procedimento de Grover) e o cálculo do número t de q-bits do primeiro registrador influenciam diretamente na probabilidade de sucesso do algoritmo. A variável t também influencia na complexidade do algoritmo. Uma vez definidas estas três variáveis, um número constante de execuções da sub-rotina propriamente são realizadas. Para cada execução, uma instância do problema é gerada randomicamente a partir do método *rand* do Scilab, usando parâmetros N e r.

Gerando as instâncias randomicamente, buscamos replicar a entrada que leva ao pior caso do algoritmo, que ocorre quando os elementos da entrada estão distribuídos de maneira homogenia ,i.e. todos os elementos com frequências próximas uma da outra. A dificuldade presente nessas entradas está no fato de que o Algoritmo Quântico de Contagem —usado para determinar a frequência de cada elemento da entrada— possui um fator de aproximação, que pode agrupar frequências próximas como se fossem iguais. Por exemplo, a função do Algoritmo Quântico de Contagem configurado com $t = 4$ e $n = 7$ possui o conjunto imagem, com os valores arredondados: $\{0,\ 5, 19,\ 40,\ 64,\ 88,\ 109,\ 123,\ 128\}$, de forma que a melhor aproximação dos elementos que possuem frequência entre 6 e 18, seja 5 ou 19. É essa faixa de erro que, como já comentado, está diretamente relacionada

com o número t de q-bits do primeiro registrador.

Depois de gerada a entrada do sistema, o estado do sistema e os operadores da simulação, como por exemplo o oráculo, e a QFT, são preparados. Continuando, para cada um dos t q-bits do primeiro registrador (com índices variando entre 0 e t-1, do menos significativo ao mais significativo) um operador C_G_y (iteração de Grover controlado pelo y-ésimo q-bit) é gerado e aplicado um total de 2^y vezes. A seguir, aplica-se a $TFQ^\dagger$ sobre o primeiro registrador, com o objetivo de estimar a fase ω. Como explicado na Seção 2.4, o estado obtido depois da aplicação da $TFQ^\dagger$ é uma superposição de estados com amplitudes concentradas em $\tilde{\omega}$, melhor aproximação da fase ω com t q-bits. Finalmente, a partir do resultado da medição de $|\tilde{\omega}\rangle$, uma aproximação $\tilde{M}$ da frequência do elemento contado é calculada. Devido a sua característica probabilística, a 'pseudo-medição' é realizada três vezes, com o objetivo de aumentar a probabilidade de medição de $\tilde{\omega}$.

Todos os passos descritos no parágrafo anterior correspondem ao Algoritmo Quântico de Contagem. O procedimento quântico de Contagem é repetido r vezes na simulação, uma vez para cada elemento na faixa de dados. Ao fim deste, a maior frequência contada até o momento e seu respectivo elemento são armazenados. Note que, no parágrafo anterior, o processo de marcação do elemento a ser contado está oculto no passo de preparação dos operadores.

4.2 Experimentos

Neste trabalho, foram realizadas simulações da sub-rotina $EncValModalB$, que é usada no algoritmo final, quando a faixa de dados é pequena, variando os parâmetros N, r e t.

4.2.1 Simulações para diferentes combinações de N e r

No primeiro grupo de simulações, são feitas 500 execuções para cada valor de N e r escolhidos. Nestas simulações, N assume os valores 2^5, 2^6, 2^7 e 2^8. E para para cada um destes valores, r assume os valores $\ell/2$, $3\ell/4$ e ℓ, onde $\ell = \sqrt[4]{N}\log N$ é o valor máximo do tamanho da faixa de dados, para o qual o algoritmo final escolhe a sub-rotina $EncValModalB$. Nestas simulações, tomamos $t = \lceil \frac{\log N}{2} \rceil$.

Os gráficos resultantes destas simulações são mostrados nas Figuras 4.1 e 4.2.

Para cada uma das 500 execuções, extraímos duas informações importantes: a taxa de acertos e a diferença relativa média. Definimos a taxa de acertos como a porcentagem

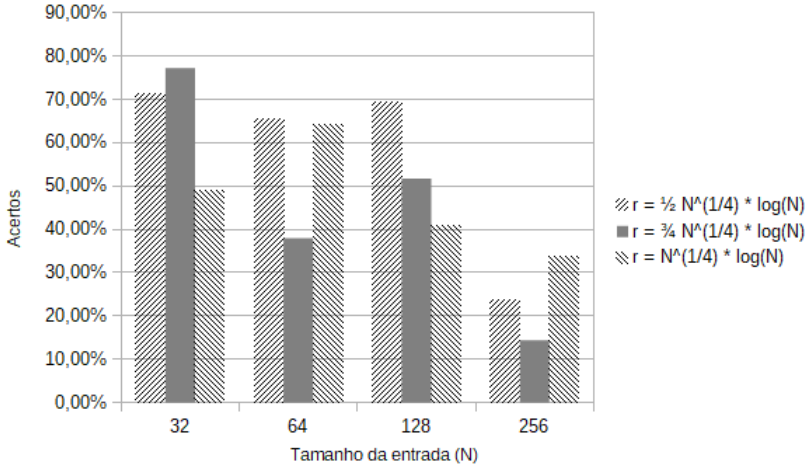

Figura 4.1: Porcentagem de acertos do Algoritmo Final

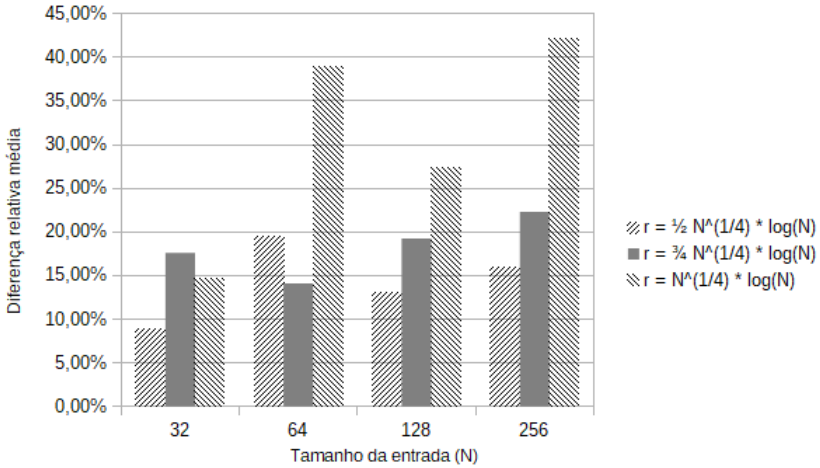

Figura 4.2: Diferença relativa média dos resultados do primeiro grupo de simulações

de soluções corretas retornadas pela sub-rotina dentre todas as 500 execuções. Enquanto a diferença relativa média é definida pela formula

$$difRelMed = \frac{1}{500} \sum_{j=0}^{500} \frac{FreqModalReal_j - FreqModalCalc_j}{FreqModalReal_j}, \tag{4.1}$$

onde $FreqModalReal_j$ é a frequência modal na execução j e $FreqModalCalc_j$ a frequência modal calculada pelo Algoritmo Quântico de Contagem na execução j.

Considerando a ineficiência do Algoritmo Quântico de Contagem para entradas com elementos distribuídos de forma homogênea, os resultados obtidos pela simulações podem ser considerados bastante satisfatórios, alcançando uma porcentagem de acertos próxima a 70% em alguns casos, como se nota no gráfico da Figura 4.1.

Comparando os gráficos das Figura 4.1 e 4.2, podemos perceber uma certa relação

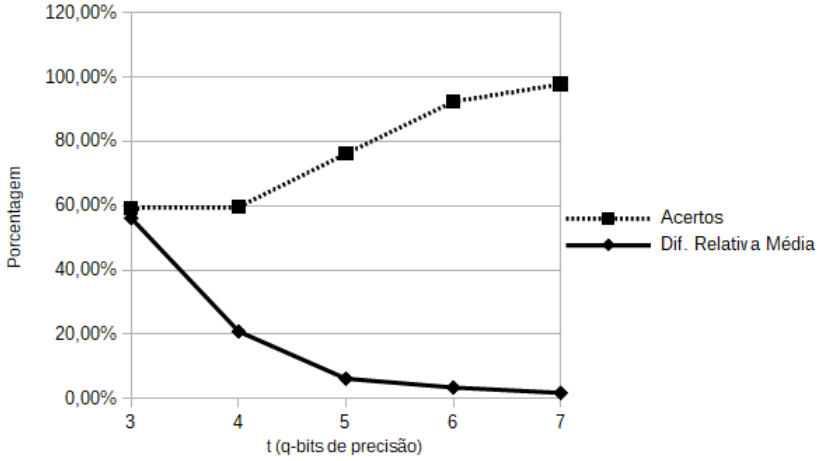

Figura 4.3: Simulação com variação de t

de proporcionalidade inversa entre a porcentagem de acertos e a diferença relativa média. Acreditamos que as pequenas variações desta proposição —como por exemplo, quando $N = 2^6$ e $r = \ell$— são causadas pelas proporções entre N e r. Essas duas teorias motivaram a realização de um segundo grupo de simulações.

4.2.2 Simulações com N e r fixos e t variável

Com base na motivação apresentada anteriormente, foram executados um segundo grupo de simulações. São feitas 500 execuções para cada valor de t escolhido. Nestas simulações, t assume os valores 3, 4, 5, 6 e 7; e os valores de N e r são fixados em 2^5 e 5, respectivamente.

Através de uma análise simples do gráfico da Figura 4.3, podemos confirmar parcialmente as teorias levantadas na seção anterior. É possível enxergar uma relação de proporcionalidade inversa entre a porcentagem de acertos e a diferença relativa média nos dados obtidos. Entretanto, o ideal é realizar um número maior de experimentos, com outros valores de N e r fixos, e se possível maiores.

Neste trabalho, não foi possível realizar simulações com N maiores, devido a limitação computacional encontrada. A máquina utilizada para os testes —restringida pela compatibilidade com o software Scilab— não foi capaz de simular nosso algoritmo com 14 q-bits ou mais, considerando a soma do número de q-bits dos dois registradores da abordagem baseada em contagem.

Capítulo 5

Conclusão

Nesse trabalho, propomos um novo algoritmo quântico para encontrar o valor modal e a frequência modal de um conjunto. O algoritmo proposto demonstra-se eficiente comparado ao limite inferior no modelo computacional clássico, necessitando apenas de $\min(o(N^{\frac{3}{4}} \log N), O(r\sqrt{N}))$ consultas à entrada.

Como dito anteriormente (veja Seção 1.1), os melhores algoritmos clássicos para encontrar o valor modal utilizam um método de ordenação, e sua complexidade é herdada deste passo. Quando r é pequeno, conseguimos uma complexidade de $O(N)$ usando um método de ordenação como o *Counting Sort*. Entretanto, nosso algoritmo ainda teria uma redução quadrática em relação a este, necessitando de apenas $\mathrm{O}(\sqrt{N})$ consultas à entrada.

Outra contribuição deste trabalho foi o desenvolvimento do código da simulação do Algoritmo Quântico Final (veja Algoritmo 5), para uma determinada faixa de dados, e consequentemente a implementação de uma simulação para o Algoritmo Quântico de Contagem. Isso possibilitou a análise de certas hipóteses quanto aos piores casos desta abordagem. Note que, o código gerado pode, com pequenas alterações, ser utilizado para implementação de um circuito quântico, executável em um computador quântico.

Como trabalhos futuros, pode-se explorar uma série de pontos: expandir o tamanho das instâncias utilizadas nas simulações e fazer uma análise comportamental mais precisa; investigar possíveis configurações do Algoritmo Quântico de Contagem com o objetivo de aumentar a sua precisão sem influenciar a ordem de complexidade; verificar a possibilidade de reduzir a complexidade de consultas do Algoritmo de Contagem para os casos em que a entrada possui elementos distribuídos de forma homogênea, com frequências próximas a N/r, usando como referência a redução de complexidade do algoritmo de Grover para

$O(\sqrt{N}/M)$, onde M é a frequência do elemento buscado; a implementação da simulação da primeira sub-rotina; e por fim a implementação da simulação do algoritmo proposto por Kara [23], comparando seus resultados com os obtidos neste trabalho.

Referências

[1] Ambainis, A. Quantum lower bounds by quantum arguments. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing* (2000), ACM, pp. 636–643.

[2] Ambainis, A. Quantum search algorithms. *ACM SIGACT News 35*, 2 (2004), 22–35.

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

[4] Beals, R.; Buhrman, H.; Cleve, R.; Mosca, M.; De Wolf, R. Quantum lower bounds by polynomials. *Journal of the ACM 48*, 4 (2001), 778–797.

[5] Belovs, A. Learning-Graph-Based Quantum Algorithm for k-Distinctness. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science* (2012), IEEE, pp. 207–216.

[6] Belovs, A. *Applications of the adversary method in quantum query algorithms*. Tese de Doutorado, University of Latvia, Faculty of Computing, Riga, Latvia, 2014.

[7] Benenti, G.; Casati, G.; Strini, G. *Principles of Quantum Computation and Information*, vol. Volume I: Basic Concepts. World Scientific, 2004.

[8] Boneh, D.; Lipton, R. J. Quantum cryptanalysis of hidden linear functions. In *Annual International Cryptology Conference* (1995), Springer, pp. 424–437.

[9] Boyer, M.; Brassard, G.; HØyer, P.; Tapp, A. Tight bounds on quantum searching. *Fortschritte der Physik 46*, 4-5 (1998), 493–505.

[10] Brassard, G.; Hoyer, P. An exact quantum polynomial-time algorithm for simon's problem. In *Fifth Israeli Symposium on Theory of Computing and Systems* (1997), IEEE, pp. 12–23.

[11] Brassard, G.; HØyer, P.; Tapp, A. Quantum counting. *Automata, Languages and Programming* (1998), 820–831.

[12] Castelvecchi, D. Quantum cloud goes commercial. *Nature 543*, 7644 (2017), 159–159.

[13] Cleve, R.; Ekert, A.; Macchiavello, C.; Mosca, M. Quantum algorithms revisited. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* (1998), vol. 454, The Royal Society, pp. 339–354.

[14] Coffey, M.; Prezkuta, Z. A quantum algorithm for finding the modal value. *Quantum Information Processing 7*, 1 (2008), 51–54.

[15] Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C. *Introduction to algorithms second edition.* The MIT Press, 2001.

[16] Debnath, S.; Linke, N.; Figgatt, C.; Landsman, K.; Wright, K.; Monroe, C. Demonstration of a small programmable quantum computer with atomic qubits. *Nature 536*, 7614 (2016), 63–66.

[17] Dirac, P. A. M. A new notation for quantum mechanics. In *Mathematical Proceedings of the Cambridge Philosophical Society* (1939), vol. 35, Cambridge University Press, pp. 416–418.

[18] Durr, C.; Hoyer, P. A quantum algorithm for finding the minimum. *arXiv preprint quant-ph/9607014* (1996).

[19] Grover, L. K. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing* (1996), ACM, pp. 212–219.

[20] Grover, L. K. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters 79*, 2 (1997), 325.

[21] Grover, L. K. A framework for fast quantum mechanical algorithms. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing* (1998), ACM, pp. 53–62.

[22] Hoyer, P.; Neerbek, J.; Shi, Y. Quantum complexities of ordered searching, sorting, and element distinctness. *Algorithmica 34*, 4 (2002), 429–448.

[23] Kara, A. A quantum algorithm for finding an ϵ-approximate mode. Master's thesis, University of Waterloo, Waterloo, Canada, 2005.

[24] Kowada, L. A. B. Mini-curso de computação quântica. IV Semana da Matemática da UFF, 2008.

[25] Kowada, L. A. B. *Construção de Algoritmos Reversíveis e Quânticos.* Novas Edições Acadêmicas, 2016.

[26] Kowada, L. A. B.; Lavor, C.; Portugal, R.; De Figueiredo, C. A new quantum algorithm for solving the minimum searching problem. *International Journal of Quantum Information 6*, 3 (2008), 427–436.

[27] Kushilevitz, E. Communication complexity. In *Advances in Computers*, vol. 44. Elsevier, 1997, pp. 331–360.

[28] Lomont, C. A quantum fourier transform algorithm. *arXiv preprint quant-ph/0404060* (2004).

[29] Mermin, D. *Quantum computer science: an introduction.* Cambridge University Press, 2007.

[30] Misra, J.; Gries, D. Finding repeated elements. *Science of Computer Programming 2*, 2 (1982), 143–152.

[31] Montanaro, A. Quantum algorithms: an overview. *NPJ Quantum Information 2*, 15023 (2016).

[32] Mosca, M. Quantum searching, counting and amplitude amplification by eigenvector analysis. In *MFCS'98 Workshop on Randomized Algorithms* (1998), pp. 90–100.

[33] Mosca, M. Counting by quantum eigenvalue estimation. *Theoretical Computer Science 264*, 1 (2001), 139–153.

[34] Nielsen, M. A.; Chuang, I. L. *Quantum Computation and Quantum Information.* AAPT, 2002.

[35] Oliveira, I. S.; Sarthour, R.; Bulnes, J.; Belmonte, S.; Guimaraes, A.; de Azevedo, E. R.; Vidoto, L.; Bonagamba, T. J.; Freitas, J. C. Computação quântica: Manipulando a informação oculta do mundo quântico. *Ciência Hoje* (2003), 22–29.

[36] Portugal, R. Element distinctness revisited. *arXiv preprint arXiv:1711.11336* (2017).

[37] Portugal, R.; Lavor, C. C.; Carvalho, L. M.; Maculan, N. Uma introdução à computação quântica.

[38] Preskill, J. Lecture notes for physics 229: Quantum information and computation, 1998.

[39] Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review 41*, 2 (1999), 303–332.

[40] Skiena, S. *The algorithm design manual: Text*, vol. 1. Springer Science & Business Media, 1998.

[41] Trabesinger, A. Quantum computing: towards reality. *Nature 543*, 7646 (2017), S1–S1.

[42] Vöcking, B.; Alt, H.; Dietzfelbinger, M.; Reischuk, R.; Scheideler, C.; Vollmer, H.; Wagner, D. *Algorithms unplugged.* Springer Science & Business Media, 2010.

[43] Whaley, B. Lecture note 20 for c191: Quantum phase estimation, eigenvalue calculations, 2007.

APÊNDICE A – Código da Simulação 6

```
// definição dos parâmetros de entrada

//numero de q-bits da entrada
n=5

//tamanho da entrada
N=2^n

//numero de q-bits controladores do alg. de contagem
t=6

//numero de operações de groover no alg. de contagem
T=2^t

//faixa de dados
r = 5


//configurando método randomico
rand('seed',7)

//número de vezes que a simulação encontrou uma resposta correta
contagemAcertos = 0

//armazena a frequência real de cada uma das 500 execuções da simulação
vetFreqModalReal = zeros(500,1)
```

```
//armazena a frequência calculada de cada uma das 500 execuções
vetFreqModalCalc = zeros(500,1)

//criação de alguns operadores e estados

//estado |0>
z = [1 0]'

//estado |1>
u = [0 1]'

//gera o operador QFT de dimensão TxT
omega = %e^(2*%pi*%i/T)
QFT = zeros(T,T)

for i = 0:T-1
for j = 0:T-1
QFT(i+1,j+1) = omega^(modulo(i*j,T))
end
end
QFT = (1/sqrt(T))*QFT
//termina a criação do operador QFT


//gera a matriz Identidade de dimensão NxN
I_N = eye(N,N)
//gera a matriz Identidade de dimensão TxT
I_T = eye(T,T)

//operador componente do procedimento de inversão sobre a média de Grover
U_0 = eye(N,N)
U_0(1) = -1

//procedimentos para gerar o operador de Hadamard H de dimensão n+t e n
H = (1/sqrt(2))*[1 1; 1 -1]
```

```
for j=1:(n+t)
if(j==1) then
H_NT = H
else
H_NT = H_NT.*.H
end
end

for j=1:(n)
if(j==1) then
H_N = H
else
H_N = H_N.*.H
end
end

//fim da criação de H de dimensão n+t

//termina a criação de operadores

//inicio simulações

contDiferentes = 0
//executa a simulação q vezes
for q=1:500

//mostra a iteração atual
disp(q,'q:')

//criação do vetor da entrada
entrada = zeros(1,N)

//procedimento para gerar os dados da entrada
for i = 1:N
entrada(1,i) = int(rand()*r)
```

```
end

//criação do vetor que irá conter as frequências de cada elemento da entrada
vetFreqElemEntrada = zeros(1,r)


// procedimento de contagem de cada elemento da entrada, para determinar a frequência
modal real etc
for i=0:(r-1)
for j=1:N
if(entrada(1,j)==i) then
vetFreqElemEntrada(1,i+1)=vetFreqElemEntrada(1,i+1)+1
end
end
end
//fim do procedimento de contagem

for elemento=0:(r-1)

//criação do estado do sistema simulado
Psi = z
for i=2:(t+n)
Psi= Psi.*.z
end
Psi = H_NT*Psi

//construção direta do oráculo para marcar o 'elemento'
O = eye(N,N)
for i=1:N
if entrada(1,i)==elemento then
O(i,i)=-1
end
end
//fim da construção do oráculo
```

```
//construção do operador G (iteração de Grover)
G = -H_N*U_0*H_N*O
//fim da construção de G

//inicializa variavel ProbsPsi, que armazena as probabilidades de leitura de cada estado
da superposição Psi (estado do sistema)
ProbsPsi = zeros(T,1)

//gera os operadores controlados de Grover e os aplica ao estado do sistema Psi
faixaTotal = N*T
for tipoGroverCtr=0:(t-1)
Ctr_G = I_T.*.I_N
var_div = 2^(t-tipoGroverCtr)

faixaParcial = faixaTotal/var_div
variador = %F
for i=0:((faixaTotal/faixaParcial)-1)
if(variador)then
posicaoInicialBloco = i*faixaParcial+1
posicaoFinalBloco = (i+1)*faixaParcial

for rep=0:(faixaParcial/N - 1)
posicaoInicialSubBloco = posicaoInicialBloco+(rep*N)
posicaoFinalSubBloco = posicaoInicialSubBloco+N-1

for k=0:(N-1)
for j=0:(N-1)
Ctr_G(posicaoInicialSubBloco+k,posicaoInicialSubBloco+j) = G(k+1,j+1)
end
end
end
variador = %F
else
variador = %T
end
```

```
end
//aplica os operadores descritos no comentario acima, logo depois da sua construção
for aplicGrover=1:(2^tipoGroverCtr)
Psi = Ctr_G*Psi
end
end
//fim da aplicação dos operadores G

//calcula as probabilidades a partir da aplitude dos estados de Psi
Psi2 = (QFT'.*.I_N)*Psi
Densi_Psi = Psi2*Psi2'
for h = 0:(T-1)
for k = 0:(N-1)
ProbsPsi(h+1,1) = Densi_Psi(h*N+(k+1),h*N+(k+1)) + ProbsPsi(h+1,1)
end
end

ProbsPsiSimples = zeros(T/2+1,1)
for h = 2:(T/2)
ProbsPsiSimples(h,1) = ProbsPsi(h,1)*2
end
ProbsPsiSimples(1,1) = ProbsPsi(1,1)
ProbsPsiSimples(T/2+1,1) = ProbsPsi(T/2+1,1)
//termina de calcular as probabilidades


//realiza tres medições, para diminuir a probabilidade de medir uma fase diferente de w̃
vet_MFreq = zeros(3,1)
for p=1:3
//inicia procedimento de medição
precisao = 1000
med_rand = int(rand()*precisao)
//probabilidade com precisao de 3 casas depois da virgula
temp_T = 1
lowerLim = 1
```

```
upperLim = (real(ProbsPsiSimples(temp_T,1)*precisao))
estado = 0

//metodo de medição
if(med_rand==0)then
estado = temp_T-1
else
while(~(med_rand>=lowerLim && upperLim>med_rand))
temp_T = temp_T+1
lowerLim = upperLim
upperLim = lowerLim+(real(ProbsPsiSimples(temp_T,1)*precisao))
end
estado = temp_T-1
end
//o valor medido está contido na variavel 'estado'

fase = estado/(2^(t))
vet_MFreq(p,1) =  ((cos(fase*2*%pi)-1)*N)/(-2)

//termina processo de medição
end
//realiza 3 medições e decide qual a "verdadeira"

//método para determinar qual a frequência medida mais perto da frequencia real do
elemento
if(vet_MFreq(1,1)==vet_MFreq(2,1)&&vet_MFreq(2,1)==vet_MFreq(3,1))then
temp_ModalFreq = vet_MFreq(1,1)
else if (vet_MFreq(1,1) == vet_MFreq(2,1)) then
temp_ModalFreq = vet_MFreq(1,1)
else if (vet_MFreq(1,1) == vet_MFreq(3,1)) then
temp_ModalFreq = vet_MFreq(3,1)
else if (vet_MFreq(2,1) == vet_MFreq(3,1)) then
temp_ModalFreq = vet_MFreq(2,1)
else
//nenhum deles são iguais
```

```
contDiferentes = contDiferentes+1
temp_ModalFreq = (vet_MFreq(1,1)+vet_MFreq(2,1)+vet_MFreq(3,1))/3
end
end
end
end
//fim do método


if(temp_ModalFreq > vetFreqModalCalc(q,1)) then
vetFreqModalCalc(q,1) = temp_ModalFreq
modalVal = elemento
end

end
//termina de contar todos os elementos da entrada

//determina a frequência modal, para comparar o resultado
for j=0:(r-1)
if(vetFreqModalReal(q,1)<vetFreqElemEntrada(1,j+1)) then
vetFreqModalReal(q,1) = vetFreqElemEntrada(1,j+1)
end
end

//conta o número de vezes, dentro as q execuções, que a simulação foi capaz de encontrar
um valor modal
if(vetFreqModalReal(q,1) == vetFreqElemEntrada(1,modalVal+1))then
contagemAcertos = contagemAcertos+1
end
end

//fim da simulação
```