Márcio Dias de Oliveira Junior

Um Estudo sobre Computação Quântica

Belo Horizonte - MG

2026

Márcio Dias de Oliveira Junior

Um Estudo sobre Computação Quântica

Monografia apresentada à coordenação de Especialização em Matemática da Universidade Federal de Minas Gerais como parte dos requisitos para a obtenção do Grau de Especialista em Matemática

Universidade Federal de Minas Gerais

Instituto de Ciências Exatas - ICEx

Programa de Especialização em Matemática

Orientador: Prof. Dr. Csaba Schneider

Belo Horizonte - MG

2026

Agradecimentos

Agradeço, primeiramente, ao professor orientador Csaba Schneider, por ter aceitado o convite para orientar esta monografia, bem como pela dedicação, paciência e genuíno interesse no tema proposto. Agradeço pela postura descontraída e ao mesmo tempo rigorosa em nossas reuniões.

Registro também meus sinceros agradecimentos aos professores das disciplinas ofertadas no curso de Especialização em Matemática da Universidade Federal de Minas Gerais. O empenho em lecionar e a paixão pelos temas apresentados contribuíram de forma decisiva para a minha formação. Em cada disciplina, além do conteúdo abordado, tive a oportunidade de conhecer diferentes formas de trabalhar e, certamente, distintas maneiras de compreender e encarar a vida.

Agradeço, ainda, aos responsáveis administrativos do curso, pelo suporte, atenção e por todo o trabalho que, muitas vezes, permanece invisível, mas cuja ausência impactaria de forma significativa o bom andamento do curso.

Por fim, deixo um agradecimento especial à minha companheira, Elizandra Cruz, por me incentivar a retomar os estudos e pela paciência ao longo desse processo, seja pela distância, seja por me ouvir falar sobre matemática sempre que retornava.

Epígrafe

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

— *Alan Turing*

Resumo

Este trabalho apresenta um estudo introdutório da computação quântica a partir dos conceitos matemáticos de álgebra linear até a a implementação prática de algoritmos quânticos por meio de circuitos utilizando a biblioteca de circuitos quânticos da IBM denominada Qiskit. Inicialmente, são introduzidos conceitos essenciais de álgebra linear, tais como espaços de Hilbert, operadores lineares e produto tensorial, estabelecendo a base matemática mínima necessária para a descrição de sistemas quânticos. Em seguida, são discutidos os postulados da mecânica quântica, com destaque para a evolução unitária, o processo de medição e a descrição de sistemas compostos, fundamentais para a compreensão do comportamento quântico e de fenômenos como superposição e emaranhamento.

Na sequência, o trabalho aborda o modelo de computação baseado em circuitos quânticos, fazendo um paralelo com circuitos clássicos e enfatizando as restrições impostas pela reversibilidade e pela impossibilidade de clonagem de estados quânticos. São apresentados exemplos centrais da computação quântica, como a geração de estados de Bell e o protocolo de teleporte quântico, além de uma introdução prática ao framework Qiskit, utilizado para construção, simulação e execução de circuitos quânticos.

Por fim, são estudados e implementados dois algoritmos quânticos: o algoritmo de Grover, para busca não estruturada, e o Algoritmo Quântico de Otimização Aproximada (QAOA), aplicado ao problema do corte máximo em grafos. Para ambos, são discutidos os princípios teóricos e a correspondência entre a formulação matemática e sua implementação computacional. No caso do QAOA, é apresentada ainda uma justificativa (informal) baseada na aproximação da evolução adiabática e na fórmula de Lie–Trotter. O trabalho busca, aproximar os fundamentos matemáticos com a noção de circuitos clássica e a prática de implementação dos algoritmos quânticos utilizando o Qiskit.

Palavras-chave: computação quântica; circuitos quânticos; algoritmo de Grover; QAOA; Qiskit.

Lista de ilustrações

Lista de símbolos

$\mathbb{C}$	Conjunto dos números complexos
$\mathbb{R}$	Conjunto dos números reais
$\mathcal{H}$	Espaço de Hilbert complexo associado a um sistema quântico
$\dim(\mathcal{H})$	Dimensão do espaço de Hilbert
$\|\psi\rangle$	Vetor de estado quântico (ket)
$\langle\psi\|$	Vetor dual associado a $\|\psi\rangle$ (bra)
$\langle\phi\|\psi\rangle$	Produto interno entre os estados $\|\phi\rangle$ e $\|\psi\rangle$
$\|\phi\rangle\langle\psi\|$	Produto externo (ou produto "ket por bra")
$\otimes$	Produto tensorial
I	Operador identidade
U	Operador unitário
H	Porta de Hadamard
X	Matriz de Pauli X
Y	Matriz de Pauli Y
Z	Matriz de Pauli Z
CX	Porta controlada-NOT
CZ	Porta controlada-Z
M_m	Operador de medição associado ao resultado m
$p(m)$	Probabilidade do resultado m na medição
$G = (V, E)$	Grafo com vértices V e arestas E
$C(z)$	Função custo do problema de otimização
$U(C, \gamma)$	Operador unitário associado ao hamiltoniano de custo
$U(B, \beta)$	Operador unitário associado ao hamiltoniano misturador
$F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$	Valor esperado da função custo no nível p do QAOA
$O(\cdot)$	Notação assintótica de complexidade

Sumário

Introdução

O modo como o ser humano interage com o ambiente é fortemente associado com suas experiências passadas. Quando uma criança vê uma bolinha sendo arremessada verticalmente, ela pode achar o movimento de subida e descida engraçado ou curioso. Instintivamente, ao repetir o experimento, a criança esperará o mesmo resultado e ficará intrigada caso não o obtenha. Situação similar ocorre em truques de mágica, onde o mágico tira um coelho de sua cartola, surpreendendo adultos e crianças.

Com conhecimento das leis de Newton a primeira situação deixa de algo inicialmente inesperado e passa a ser algo previsível. A segunda situação, quando se descobre o segredo do truque usado pelo mágico, deixa de ser "mágica" e passa a ser uma encenação para direcionar a atenção do público de modo que não perceba o que realmente está acontecendo. Ainda sim, podemos dizer que o truque de mágica é um feito surpreendente, pois enganar uma plateia enorme e atenta não é fácil, porém, já não se trata de algo mágico.

A mecânica quântica pode ser vista como a descrição da matéria e da luz em todo o seu detalhe em escala atômica. O comportamento em escala atômica difere de tudo que o ser humano tem experiência direta. Tal comportamento não é puramente ondulatório, não é um comportamento de partícula, não são como nuvens, ou bolas de bilhar, ou pesos em molas, ou qualquer coisa da experiência humana do dia a dia.

Historicamente, imaginava-se que o elétron se comportava como partícula, mais tarde, observou-se que também se comporta como onda. Como este comportamento dual, pode-se dizer que, no fim, o elétron não se comporta nem como partícula, nem como onda.

A descrição do comportamento em escala atômica foi feita com precisão por volta dos anos de 1926 e 1927 com os trabalhos de Schrödinger, Heisenberg e Bohr. Schrödinger introduziu a formulação ondulatória através de uma equação que descreve um sistema quântico em determinado tempo. Heisenberg introduziu o princípio da incerteza, apresentando limites para o que pode ser medido em um sistema quântico. Born apresentou a interpretação probabilística da função de onda. Em conjunto com contribuições de Dirac, Bohr e Jordan tais ideias culminaram na interpretação de Copenhague da mecânica quântica firmada entre 1927 e 1930 Faye [2022].

Por volta do ano de 1936 Alan Turing e Alonzo Church mostraram a noção de computabilidade universal Turing [1936] e Church [1936]. Turing apresenta a noção de computabilidade através de determinadas máquinas (que podemos visualizar como algoritmos), mais ainda, demonstra que existe uma máquina universal capaz de executar qualquer par de especificação de máquina e entrada. Assim, a noção de computabilidade é completamente capturada por tais máquinas.

A partir de 1970 diversas técnicas para controlar sistemas quânticos foram inventadas, por exemplo, o isolamento de um único átomo de modo a permitir a investigação de seu comportamento com grande precisão. Com a possibilidade de controle de tais sistemas e a combinação em sistemas maiores (com grande dificuldade, mas possível), surge o questionamento sobre a utilização das propriedades quânticas para execução de determinadas operações, em outras palavras, para computação.

No mesmo período a teoria da complexidade computacional é desenvolvida e uma noção formal para solução eficiente e ineficiente é construída e apresentada por Cook Cook [1971] e Levin Levin [1973].

O objetivo principal deste trabalho é estudar a base matemática utilizada em algoritmos quânticos e utilizá-la para implementar alguns algoritmos bem conhecidos. Para isso, no capítulo 1 introduzimos conceitos iniciais de álgebra linear. No capítulo 2 os postulados da mecânica quântica e noções de medida. O capítulo 3 é destinado a introdução de circuitos quânticos e à biblioteca de circuitos quânticos da IBM chamada Qiskit. No capítulo 4 são apresentados dois algoritmos quânticos o algoritmo de Grover e o Algoritmo Quântico de Otimização Aproximada (QAOA). O capítulo 5 apresenta as conclusões desta monografia.

1 Conceitos Iniciais

1.1 Introdução

Neste capítulo apresentamos as notações utilizadas ao longo do texto. A computação quântica possui um conjunto de notações herdadas da física, da matemática e da computação. Essa mescla de áreas do conhecimento da origem a uma notação bastante elegante, mas que exige uma boa preparação para que o leitor possa desfrutar dos resultados apresentados.

O texto apresentado adiante admite que o leitor está familiarizado com resultados de álgebra linear tais como espaços vetoriais, vetores, matrizes, bases, independência linear, operadores lineares e outros resultados comuns. Uma boa sugestão de leitura para tais tópicos é Coelho and Lourenço [2001]. Também é esperado que o leitor tenha conhecimento básico de computabilidade, análise de algoritmos e complexidade computacional tal como notação assintótica ou de Landau; uma boa referência de leitura é Arora and Barak [2009]. Os resultados e notações aqui apresentadas são baseadas no capítulo 2 de Nielsen and Chuang [2010].

1.2 Espaços de Hilbert

Denotamos por $\mathbb{C}$ o conjunto dos números complexos. Dado $z \in \mathbb{C}$, denotamos por z^* seu conjugado. Assim,

$$z = a + bi \iff z^* = a - bi.$$

Todos os espaços vetoriais utilizados serão espaços vetoriais de dimensão finita sobre o corpo dos números complexos e serão denotados por letras latinas maiúsculas, por exemplo, $V, Z, W, \ldots$ ou pela letra curva $\mathcal{H}$, sem índices subscritos ou com índices tal como $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_A, \mathcal{H}_B, \ldots$. Se dissermos que um sistema quântico é formado por um espaço vetorial com produto interno, estamos nos referindo a um espaço vetorial de dimensão finita com produto interno sobre o corpo dos complexos. Um elemento arbitrário do espaço vetorial (um vetor) será denotado por $|\cdot\rangle$ ("*ket*") onde o símbolo "$\cdot$" poderá ser uma letra minúscula do alfabeto latino ou grego, por exemplo, $|a\rangle, |b\rangle, |c\rangle, \ldots, |\psi\rangle, |\varphi\rangle, \ldots$. A única exceção a esta regra é o vetor nulo que será denotado por 0, o vetor $|0\rangle$, como veremos mais a frente, não denotará o vetor nulo. Omitiremos o corpo ao qual os espaços vetoriais estão definidos, o leitor deve assumir que o corpo de todos os espaços vetoriais apresentados é o corpo dos números complexos.

Representaremos o produto interno definido entre dois vetores $|\psi\rangle$ e $|\varphi\rangle$, em um

espaço vetorial V, por $\langle\psi|\varphi\rangle$. Além disso, se V possui produto interno definido, indicamos por $\langle\psi|$ ("*bra*") o vetor dual relativo ao produto interno de $|\psi\rangle$. Deste modo, $\langle\psi|$ é o funcional linear no espaço dual ao espaço de $|\psi\rangle$ tal que

$$\langle\psi|\left(|\varphi\rangle\right) = \langle\psi|\varphi\rangle\,, \quad \text{para todo } |\psi\rangle \in V, \tag{1.1}$$

que indica que a notação para o vetor dual é bastante intuitiva e decorre como resultado do teorema da Representação de Riesz.

Teorema 1.1 (Teorema da Representação de Riesz)**.** *Seja $\mathcal{H}$ um espaço vetorial de dimensão finita sobre $\mathbb{C}$, munido de produto interno, e $f : \mathcal{H} \to \mathbb{C}$ um funcional linear. Então existe um único $|y_0\rangle \in \mathcal{H}$ tal que*

$$f(x) = \langle y_0|x\rangle\,, \forall\, |x\rangle \in \mathcal{H}.$$

Dois vetores $|v_1\rangle$, $|v_2\rangle$ são ortogonais se $\langle v_1|v_2\rangle = 0$. Sendo V um espaço vetorial, uma função $||\cdot|| : V \to \mathbb{R}$ que satisfaz as propriedades:

- $||\,|v\rangle\,|| \geq 0$
- $||\lambda\,|v\rangle\,|| = |\lambda|||\,|v\rangle\,||$
- $||\,|v\rangle + |w\rangle\,|| \leq ||\,|v\rangle\,|| + ||\,|w\rangle\,||$,

é denominada *norma*. A função $||\,|v\rangle\,|| = \sqrt{\langle v|v\rangle}$ é a norma induzida pelo produto interno. Se $|v\rangle$ é tal que $||\,|v\rangle\,|| = 1$ então $|v\rangle$ é um vetor *unitário*. Para qualquer $|w\rangle \neq 0$ podemos produzir um vetor unitário tomando $|v\rangle = |w\rangle\,/||\,|w\rangle\,||$.

Uma base é dita *ortogonal* se para todo par de vetores $|v_i\rangle\, e\, |v_j\rangle$ distintos ocorrer $\langle v_i|v_j\rangle = 0$. Se além de ortogonais, todos os vetores da base forem unitários diremos que a base é *ortonormal*. Caso uma base não seja ortogonal podemos utilizar o processo de ortogonalização de vetores de Gram-Schmidt para obter uma base ortonormal.

Teorema 1.2 (Processo de ortogonalização de vetores de Gram-Schmidt)**.** *Seja $\{|w_i\rangle\}_{i=1}^{d}$ uma base para um espaço vetorial V com produto interno. Defina*

1. $|v_1\rangle := |w_1\rangle$
2. $|v_{k+1}\rangle := |w_{k+1}\rangle - \sum_{i=1}^{k} \dfrac{\langle v_i|w_{k+1}\rangle\,|v_i\rangle}{||\,|v_i\rangle\,||^2}$.

Então, o conjunto $\{|v_1\rangle\}_{i=1}^{d}$ é base ortogonal para V. Além disso, se fizermos, para cada $i = 1, \ldots, d$,

$$|v_i'\rangle = \frac{|v_i\rangle}{||\,|v_i\rangle\,||},$$

então $\{|v_i'\rangle\}_{i=1}^{d}$ é base ortonormal de V.

Por vezes, ao nos referirmos a um sistema quântico, ao invés de mencionar, por exemplo, que um sistema quântico é um espaço vetorial sobre $\mathbb{C}$ com produto interno, diremos apenas que o sistema quântico é um *espaço de Hilbert* sobre $\mathbb{C}$ ou, simplesmente, um espaço de Hilbert. Um espaço de Hilbert é um espaço vetorial com produto interno que é completo em relação a norma induzida pelo produto interno. Como trabalharemos somente com espaços vetoriais de dimensão finita e com o produto interno canônico dos complexos temos que tais espaços vetoriais são completos.

Exemplo 1.3. *$\mathcal{H} = \mathbb{C}^2$ sobre o corpo $\mathbb{C}$ e com o produto interno canônico dos complexos é um espaço de Hilbert. Além disso, os vetores*

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$
$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

formam uma base ortonormal para $\mathcal{H}$. O par de vetores $|+\rangle$ e $|-\rangle$ também é base ortonormal para $\mathcal{H}$,

$$|+\rangle := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$|-\rangle := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle),$$

Bem como o par $|+i\rangle, |-i\rangle$ também é base para $\mathcal{H}$,

$$|+i\rangle := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle)$$
$$|-i\rangle := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - i\,|1\rangle).$$

Como generalização, dado o espaço de Hilbert $\mathcal{H} = \mathbb{C}^m$ sobre $\mathbb{C}$, os vetores com m coordenadas

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ldots, |j\rangle = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \ldots, |m-1\rangle = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 1 \end{bmatrix},$$

formam uma base ortonormal.

1.3 Operadores

Matrizes ou transformações lineares serão denotadas por letras latinas maiúsculas, por exemplo, I (a matriz identidade 2×2) ou

$$A = \begin{bmatrix} 10 & 3-4i & 0 \\ 5 & 0 & 0 \\ 1 & 0 & 2 \\ 0 & i & -1 \end{bmatrix}. \tag{1.2}$$

Uma transformação linear T é uma função $T : V \to W$ onde V e W são espaços vetoriais e que satisfaz as seguintes condições:

1. $T(a\,|v\rangle) = aT(|v\rangle) = aT\,|v\rangle\,, \quad \forall a \in \mathbb{C}$ e $\forall\,|v\rangle \in V$.

2. $T(|v_1\rangle + |v_2\rangle) = T(|v_1\rangle) + T(|v_2\rangle) = T\,|v_1\rangle + T\,|v_2\rangle\,, \quad \forall\,|v_1\rangle\,, |v_2\rangle \in V$.

Estendemos a notação do produto interno dos vetores $|w\rangle$ e $A\,|v\rangle$ e denotamos seu produto interno por $\langle w|A|v\rangle$. Dadas duas transformações lineares $T_1 : V \to W$ e $T_2 : W \to Z$ a composição $T_2T_1 : V \to Z$ também é uma transformação linear, além disso, dada uma base $B = \{|v_1\rangle\,, |v_2\rangle\,, \ldots\,, |v_n\rangle\}$ de um espaço vetorial V e $\{|w_1\rangle\,, |w_2\rangle\,, \ldots\,, |w_n\rangle\} \subset W$ então existe uma única transformação linear $T : V \to W$ tal que $T\,|v_i\rangle = |w_i\rangle$ para todo i. Escolhendo-se bases para V e W, cada transformação linear pode ser representada como uma matriz (matriz de transformação) e, analogamente, toda matriz representa uma transformação linear.

Podemos fazer a aplicação do conjugado complexo. Sendo A dada pela equação (1.2) temos

$$A^* = \begin{bmatrix} 10 & 3+4i & 0 \\ 5 & 0 & 0 \\ 1 & 0 & 2 \\ 0 & -i & -1 \end{bmatrix}.$$

A transposta da matriz A é denotada por A^T e denotamos por $A^\dagger$

$$A^\dagger := (A^T)^* = \begin{bmatrix} 10 & 5 & 1 & 0 \\ 3+4i & 0 & 0 & -i \\ 0 & 0 & 2 & 1 \end{bmatrix}.$$

Note que $A^\dagger = (A^T)^* = (A^*)^T$. As matrizes

$$\sigma_x = X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_y = Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \text{e} \quad \sigma_z = Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{1.3}$$

são conhecidas como *matrizes de Pauli* e são muito utilizadas na mecânica quântica. Neste texto, utilizaremos os termos *operador linear* ou *operador* como sinônimos para transformações lineares.

Um autovetor de uma transformação linear $A : V \to V$ é um vetor não nulo $|v\rangle$ tal que $A|v\rangle = \lambda|v\rangle$, onde λ é um número complexo chamado autovalor de A associado ao autovetor $|v\rangle$. A função característica de um operador linear A é dada por

$$c(\lambda) = \det(A - \lambda I).$$

As soluções para $c(\lambda) = 0$ são os autovalores de A.

Seja $|v\rangle$ um vetor em um espaço vetorial V com produto interno e seja $|w\rangle$ um vetor em um espaço vetorial W com produto interno. Chamaremos de *produto externo* ou *produto "ket por bra"* o operador linear dado por

$$|w\rangle\langle v| : V \mapsto W, \quad |v'\rangle \mapsto |w\rangle\langle v|v'\rangle, \quad \forall |v'\rangle \in V. \tag{1.4}$$

Em outras palavras, $(|w\rangle\langle v|)(|v'\rangle)$ é o resultado da multiplicação do vetor $|w\rangle$ pelo número complexo $\langle v|v'\rangle$. Naturalmente, podemos fazer combinações lineares com operadores de produto externo; dados $|v_0\rangle, \ldots, |v_{n-1}\rangle \in V$ e $|w_0\rangle, \ldots, |w_{n-1}\rangle \in W$,

$$\sum_{i=0}^{n-1} a_i |w_i\rangle\langle v_i| : V \to W, \quad |v'\rangle \mapsto \sum_{i=0}^{n-1} a_i \langle v_i|v'\rangle |w_i\rangle.$$

Observe que, dada uma base $\{|0\rangle, |1\rangle, \ldots, |n-1\rangle\}$, podemos escrever qualquer vetor $|v\rangle$ como

$$|v\rangle = \sum_{i=0}^{n-1} b_i |i\rangle$$

para alguns números complexos b_i e, deste modo,

$$\sum_{i=0}^{n-1} |i\rangle\langle i|v\rangle = \sum_{i=0}^{n-1} \langle i|v\rangle |i\rangle = \sum_{i=0}^{n-1} b_i |i\rangle = |v\rangle.$$

Portanto,

$$\sum_{i=0}^{n-1} |i\rangle\langle i| = I \tag{1.5}$$

que é denominada *relação de completude* para vetores ortonormais. Podemos representar qualquer operador linear utilizando a notação de produto externo, para tal, seja $A : V \to W$

um operador, $\{|v_i\rangle\}_{j=1}^m$ base ortonormal de V e $\{|w_i\rangle\}_{i=1}^n$ base ortonormal de W então

$$I_V = \sum_{j=1}^{m} |v_j\rangle \langle v_j| , \quad I_W = \sum_{i=1}^{n} |w_i\rangle \langle w_i| \tag{1.6}$$

$$A = I_W A I_V \tag{1.7}$$

$$= \left(\sum_{i=1}^{n} |w_i\rangle \langle w_i| \right) A \left(\sum_{j=1}^{m} |v_j\rangle \langle v_j| \right) \tag{1.8}$$

$$= \sum_{ij} |w_i\rangle \left(\langle w_i| A |v_j\rangle \right) \langle v_j| \tag{1.9}$$

$$= \sum_{ij} (\langle w_i|A|v_j\rangle) |w_i\rangle \langle v_j| \tag{1.10}$$

$$= \sum_{ij} A_{ij} |w_i\rangle \langle v_j| , \tag{1.11}$$

onde A_{ij} é a entrada na posição (i, j) de A.

Exemplo 1.4. *As matrizes de Pauli são representadas na notação de produto externo como*

$$\sigma_x = |0\rangle \langle 1| + |1\rangle \langle 0| , \quad \sigma_y = -i |0\rangle \langle 1| + i |1\rangle \langle 0| , \quad \sigma_z = |0\rangle \langle 0| - |1\rangle \langle 1| .$$

Um operador $A : V \to V$ é diagonalizável se existe base onde a matriz de A é diagonal. Em outras palavras, existe base $\{|v_i\rangle\}_{i=1}^n$ de V formada por autovetores de A, tal que

$$A |v_i\rangle = \lambda_i |v_i\rangle .$$

Nesta base, a matriz que representa A é diagonal com $\lambda_i, \ldots , \lambda_n$ na diagonal principal. Ademais se a base $\{|v_i\rangle\}_{i=0}^n$ é ortonormal, então

$$A = \sum_{i=1}^{n} \lambda_i |v_i\rangle \langle v_i| . \tag{1.12}$$

Chamamos a equação (1.12) de *representação diagonal* do operador A; se tomarmos $P_i = |v_i\rangle \langle v_i|$ para cada i, temos

$$A = \sum_{i=1}^{n} \lambda_i P_i .$$

Cada operador P_i é projeção ortogonal sobre o espaço unidimensional gerado por v_i. A forma diagonal de um operador é bastante útil, pois permite identificar com facilidade seus autovetores e autovalores associados.

Exemplo 1.5. *É fácil verificar que para o operador σ_x os autovalores são 1 e -1 com autovetores correspondentes ortogonais $|+\rangle$ e $|-\rangle$. Portanto, a representação diagonal de σ_x é dada por*

$$\sigma_x = |+\rangle \langle +| - |-\rangle \langle -| .$$

As outras matrizes de Pauli também podem ser representadas desta forma. De fato,

$$\sigma_y = |+i\rangle\langle+i| - |-i\rangle\langle-i|, \quad \sigma_z = |0\rangle\langle 0| - |1\rangle\langle 1|.$$

Lembrando que os vetores $|-\rangle, |+\rangle, |+i\rangle$ *e* $|-i\rangle$ *foram definidos no exemplo 1.3.*

Seja A um operador em um espaço de Hilbert V chamamos de $A^\dagger$ o único operador em V tal que

$$\langle v|A|w\rangle = \langle (A^\dagger|v\rangle)|w\rangle. \tag{1.13}$$

Nos casos onde $A = A^\dagger$ dizemos que A é um operador autoadjunto ou hermitiano. Um operador linear U é chamado de unitário quando ocorrer de $U^\dagger U = I$, operadores unitários tem papel fundamental na mecânica quântica, pois preservam a norma de vetores, além de serem operadores invertíveis que, para mecânica quântica, significa que eles correspondem a *operações reversíveis*.

1.4 Produto tensorial

O produto tensorial é fundamental para a mecânica quântica; veremos suas propriedades gerais a seguir, para uma definição formal consulte [Roman, 2007, capítulo 14]. Sejam $\mathcal{H}_V$ e $\mathcal{H}_W$ dois espaços de Hilbert. O produto tensorial desses espaços, denotado por

$$\mathcal{H}_V \otimes \mathcal{H}_W,$$

é um novo espaço de Hilbert. Para vetores $|\psi\rangle \in \mathcal{H}_V$ e $|\varphi\rangle \in \mathcal{H}_W$, define-se o vetor

$$|\psi\rangle \otimes |\varphi\rangle \in \mathcal{H}_V \otimes \mathcal{H}_W,$$

satisfazendo a propriedade de bilinearidade:

$$\begin{aligned}(a|\psi_1\rangle + b|\psi_2\rangle) \otimes (c|\varphi_1\rangle + d|\varphi_2\rangle) = {} & ac\,|\psi_1\rangle \otimes |\varphi_1\rangle \\ & + ad\,|\psi_1\rangle \otimes |\varphi_2\rangle \\ & + bc\,|\psi_2\rangle \otimes |\varphi_1\rangle \\ & + bd\,|\psi_2\rangle \otimes |\varphi_2\rangle,\end{aligned}$$

para quaisquer escalares a, b, c, d. Se $\{|i\rangle\}_{i=0}^{m-1}$ é uma base ortonormal de $\mathcal{H}_V$ e $\{|j\rangle\}_{j=0}^{n-1}$ é uma base ortonormal de $\mathcal{H}_W$, então

$$\{|i\rangle \otimes |j\rangle\}_{i,j=0}^{i=m-1,j=n-1}$$

é uma base ortonormal de $\mathcal{H}_V \otimes \mathcal{H}_W$. Em particular,

$$\dim(\mathcal{H}_V \otimes \mathcal{H}_W) = m \cdot n. \tag{1.14}$$

Se $A : \mathcal{H}_V \to \mathcal{H}_V$ e $B : \mathcal{H}_W \to \mathcal{H}_W$, então podemos definir o operador

$$A \otimes B : \mathcal{H}_V \otimes \mathcal{H}_W \to \mathcal{H}_V \otimes \mathcal{H}_W$$

por sua ação em vetores decomponíveis ou tensores puros:

$$(A \otimes B)(|\psi\rangle \otimes |\varphi\rangle) = A\,|\psi\rangle \otimes B\,|\varphi\rangle\,,$$

estendendo-se linearmente a todo o espaço. A propriedade universal do produto tensorial implica que tal operador é sempre bem definido. Por exemplo, Seja $\mathcal{H}_V = \mathbb{C}^3$, $\mathcal{H}_W = \mathbb{C}^2$, $B_{\mathcal{H}_V} = \{|0\rangle, |1\rangle, |2\rangle\}$ base de $\mathcal{H}_V$ e $B_{\mathcal{H}_W} = \{|0\rangle, |1\rangle\}$ base de $\mathcal{H}_W$, então

$$(X \otimes I)(|0\rangle \otimes |1\rangle) = X\,|0\rangle \otimes I\,|1\rangle = |1\rangle \otimes |1\rangle\,. \tag{1.15}$$

Dados $|x\rangle = \sum_{k=1}^{n} |v_k\rangle \otimes |w_k\rangle$, $|y\rangle = \sum_{l=1}^{m} |v'_l\rangle \otimes |w'_l\rangle \in \mathcal{H}_V \otimes \mathcal{H}_W$. Calculamos o produto interno em $\mathcal{H}_V \otimes \mathcal{H}_W$ com base no produto interno de $\mathcal{H}_V$ e $\mathcal{H}_W$ da seguinte forma

$$\langle x|y\rangle = \sum_{k=1}^{n}\sum_{l=1}^{m} \langle v_k|v'_l\rangle \langle w_k|w'_l\rangle\,. \tag{1.16}$$

Dado o espaço de Hilbert $\mathcal{H}$ podemos construir o espaço $\mathcal{H}^{\otimes^n} = \bigotimes_{i=1}^{n} \mathcal{H}$. Sendo $A : \mathcal{H} \to \mathcal{H}$ um operador, denotamos por $A^{\otimes^n} : \mathcal{H}^{\otimes^n} \to \mathcal{H}^{\otimes^n}$ o operador

$$A^{\otimes^n}|\psi\rangle = A\,|\psi_1\rangle \otimes A\,|\psi_2\rangle \ldots \otimes A\,|\psi_n\rangle\,, \quad \text{para todo } |\psi\rangle = \bigotimes_{i=1}^{n} |\psi_i\rangle \in \mathcal{H}^{\otimes^n}. \tag{1.17}$$

Uma forma concreta de representar os elementos e os operadores do produto tensorial é através do produto de Kronecker de matrizes. Seja $A = (a_{ij})_{mn}$ e $B = (b_{ij})_{pq}$, definimos o produto de Kronecker de $A \otimes B$ como a matriz mp por nq dada por

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}.$$

Exemplo 1.6. *O produto de Kronecker aplicado na equação (1.15) é dado por*

$$(X \otimes I)(|0\rangle \otimes |1\rangle) = X\,|0\rangle \otimes I\,|1\rangle = |1\rangle \otimes |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{1.18}$$

Usando o produto de Kronecker, o espaço $\mathbb{C}^2 \otimes \mathbb{C}^2$ pode ser identificado por $\mathbb{C}^4$ e o vetor $[0,1]^T \otimes [0,1]^T$ pode ser identificado como o vetor $[0,0,0,1]^T$.

2 Os Postulados da Mecânica Quântica

2.1 Introdução

Este capítulo é baseado no conteúdo de [Nielsen and Chuang, 2010, seção 2.2]. O objetivo é apresentar os postulados e permitir ao leitor associar os conceitos iniciais de álgebra linear apresentados no capítulo 1 com os conceitos de mecânica quântica.

2.2 Postulado 1 - Espaço de Estados

A um sistema físico isolado associa-se um espaço de Hilbert complexo $\mathcal{H}$, chamado *espaço de estados* do sistema. O estado do sistema é completamente descrito por um vetor unitário

$$|\psi\rangle \in \mathcal{H}, \qquad \langle\psi|\psi\rangle = 1.$$

Vetores que diferem apenas por uma fase global $e^{i\theta}$ representam o mesmo estado físico, isto é,

$$|\psi\rangle \sim e^{i\theta}\,|\psi\rangle\,, \qquad \theta \in \mathbb{R} \tag{2.1}$$

O sistema quântico mais simples é o *qubit*. Um qubit possui um espaço de estados bidimensional. Seja $B = \{|0\rangle\,, |1\rangle\}$ base ortonormal de $\mathbb{C}^2$. Então, um vetor de estado arbitrário nesse espaço pode ser escrito como

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle\,, \tag{2.2}$$

onde a e b são números complexos tal que $|a|^2 + |b|^2 = 1$. Os estados $|0\rangle$ e $|1\rangle$ podem ser vistos como os valores 1 e 0 que um bit pode assumir na computação clássica. No entanto, o qubit pode assumir superposições dos estados $|0\rangle$ e $|1\rangle$, ou seja, enquanto o bit pode assumir somente dois estados, um qubit pode assumir uma quantidade não enumerável de estados.

Podemos representar visualmente o qubit usando a *Esfera de Bloch*. Qualquer estado de um qubit pode ser representado, a menos de uma fase global, por um ponto na superfície da esfera. Por exemplo, Seja um estado arbitrário

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle\,, \qquad a, b \in \mathbb{C}, \qquad |a|^2 + |b|^2 = 1. \tag{2.3}$$

Escrevemos os coeficientes como

$$a = |a|e^{i\alpha}, \qquad b = |b|e^{i\beta}, \tag{2.4}$$

assim,

$$|\psi\rangle = e^{i\alpha}\left(|a|\,|0\rangle + e^{i(\beta-\alpha)}|b|\,|1\rangle\right). \tag{2.5}$$

Descartando a fase global $e^{i\alpha}$, que pode ser feito pela equação (2.1), e definindo

$$\phi = \beta - \alpha,$$

obtemos um estado equivalente

$$|\psi\rangle \sim |a|\,|0\rangle + e^{i\phi}|b|\,|1\rangle. \tag{2.6}$$

Como $0 \leq |a| \leq 1$, existe um ângulo $\theta \in [0, \pi]$ tal que

$$|a| = \cos\left(\frac{\theta}{2}\right). \tag{2.7}$$

Como $||\,|\psi\rangle\,|| = 1$, segue que

$$|b| = \sqrt{1-|a|^2} = \operatorname{sen}\left(\frac{\theta}{2}\right). \tag{2.8}$$

Considerando que $e^{i\phi} = \cos\phi + i\operatorname{sen}\phi$, temos que

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\operatorname{sen}\left(\frac{\theta}{2}\right)|1\rangle \tag{2.9}$$

$$= \cos\left(\frac{\theta}{2}\right)|0\rangle + \cos\phi\operatorname{sen}\left(\frac{\theta}{2}\right)|1\rangle + \operatorname{sen}\phi\operatorname{sen}\left(\frac{\theta}{2}\right) i\,|1\rangle. \tag{2.10}$$

Como o $\mathbb{R}$-espaço $\mathbb{C}^2$ é gerado pela base $\{|0\rangle, i\,|0\rangle, |1\rangle, i\,|1\rangle\}$ e $|\psi\rangle$ encontra-se no subespaço 3-dimensional $\mathbb{R}^3$ gerado por $\{|0\rangle, |1\rangle, i\,|1\rangle\}$, representamos $|\psi\rangle$, para $0 \leq \theta \leq \pi$ e $0 \leq \phi < 2\pi$, como o vetor unitário

$$\vec{r} = (\operatorname{sen}\theta\cos\phi,\ \operatorname{sen}\theta\operatorname{sen}\phi,\ \cos\theta), r \in \mathbb{R}^3.$$

Observe que ao definir as coordenadas de $\vec{r}$ dobramos o ângulo θ, os vetores $|0\rangle$ e $|1\rangle$ são perpendiculares, e ao desenhar a esfera de Bloch fazemos uma representação dobrando o ângulo.

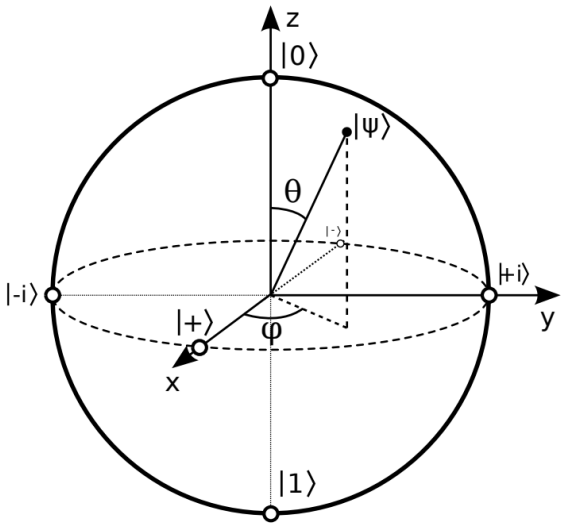

Figura 1 – Estado $|\psi\rangle$ representado na Esfera de Bloch.

2.3 Postulado 2 - Evolução

A evolução temporal de um sistema quântico fechado é descrita por uma transformação unitária. Isto é, o estado do sistema no instante t_1, representado por $|\psi\rangle$, está relacionado ao estado $|\psi'\rangle$ no instante t_2 por

$$|\psi'\rangle = U\,|\psi\rangle\,, \tag{2.11}$$

onde U é um operador unitário que depende apenas do sistema e dos instantes t_1 e t_2.

Exemplo 2.1 (Operadores X e Z)**.** *Considere os operadores de Pauli*

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad e \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.12}$$

Estes operadores são unitários pois $XX^\dagger = X^\dagger X = I$ *e* $ZZ^\dagger = Z^\dagger Z$*. O operador* X*, pode ser interpretado como* operador NOT quântico*, invertendo os estados* $|0\rangle$ *e* $|1\rangle$

$$X\,|0\rangle = |1\rangle\,, \qquad X\,|1\rangle = |0\rangle\,. \tag{2.13}$$

O operador Z *atua como um* operador de mudança de fase*, deixando o estado* $|0\rangle$ *inalterado e introduzindo uma fase relativa ao ângulo* π *no estado* $|1\rangle$*:*

$$Z\,|0\rangle = |0\rangle\,, \qquad Z\,|1\rangle = -\,|1\rangle\,. \tag{2.14}$$

Exemplo 2.2 (Operador de Hadamard)**.** *O operador de Hadamard dado por*

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2.15}$$

é um exemplo de operador unitário. Este operador em especial é bastante utilizado em algoritmos quânticos para preparação de estados. Considere o estado

$$|\psi\rangle = |0\rangle\,,$$

Então,

$$H\,|\psi\rangle = H\,|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle\,,$$

onde $|+\rangle$ *é o vetor definido no exemplo 1.3.*

O Postulado 2 afirma que a evolução temporal de um sistema quântico fechado é descrita por uma transformação unitária. Para evoluções contínuas no tempo, essa transformação unitária é gerada pelo hamiltoniano do sistema. O hamiltoniano H independente do tempo de um sistema quântico é um operador hermitiano que representa a energia total do sistema, de forma que a evolução temporal entre os instantes 0 e t é dada por

$$U(t) = e^{-iHt/\hbar}. \tag{2.16}$$

Na última igualdade, $\hbar$ é a constante de Planck que, em geral, pode ser incorporada em H de modo que $\tilde{H} = H/\hbar$ e

$$U(t) = e^{-i\tilde{H}t}. \tag{2.17}$$

A unitariedade de $U(t)$ decorre diretamente do fato de que H é hermitiano, pois

$$U(t)^\dagger U(t) = e^{i\tilde{H}t}e^{-i\tilde{H}t} = I. \tag{2.18}$$

2.4 Postulado 3 - Medição Quântica

A medição quântica é descrita por um conjunto de operadores de medição $\{M_i\}_{i=1}^m$ que atuam no espaço de estados do sistema, satisfazendo a relação de completude

$$\sum_{i=1}^{m} M_i^\dagger M_i = I. \tag{2.19}$$

Observe que, não é necessário que o espaço de estados tenha dimensão igual ao número de operadores de medição (caso onde $\{M_i\}_{i=1}^m$ seriam projeções para autoespaços de dimensão igual a um), basta que a relação de completude seja satisfeita. Se o sistema está no estado $|\psi\rangle$ imediatamente antes da medição, então a probabilidade de se obter o resultado $i \in \{1, \ldots, m\}$ é dada por

$$p(i) = \langle\psi|M_i^\dagger M_i|\psi\rangle\,. \tag{2.20}$$

Dado que o resultado i é obtido, o estado do sistema após a medição passa a ser

$$|\psi'\rangle = \frac{M_i\,|\psi\rangle}{\sqrt{p(i)}}. \tag{2.21}$$

Exemplo 2.3 (Medição do qubit na base computacional)**.** *A medição de um qubit na* base computacional*, trata-se de uma medição em um único qubit, com dois resultados possíveis, definida pelos operadores de medição*

$$M_0 = |0\rangle\langle 0| \qquad e \qquad M_1 = |1\rangle\langle 1|. \tag{2.22}$$

Observe que cada operador de medição é hermitiano e satisfaz

$$M_0^2 = M_0 \qquad e \qquad M_1^2 = M_1.$$

Assim, a relação de completude é satisfeita:

$$I = M_0^\dagger M_0 + M_1^\dagger M_1 = M_0 + M_1. \tag{2.23}$$

Suponha que o estado a ser medido seja

$$|\psi\rangle = a|0\rangle + b|1\rangle. \tag{2.24}$$

Então, a probabilidade de se obter o resultado de medição 0 *é*

$$p(0) = \langle\psi|M_0^\dagger M_0|\psi\rangle = \langle\psi|M_0|\psi\rangle = |a|^2. \tag{2.25}$$

De modo análogo, a probabilidade de se obter o resultado de medição 1 *é*

$$p(1) = |b|^2. \tag{2.26}$$

O estado do sistema após a medição, nos dois casos, é portanto dado por

$$\frac{M_0|\psi\rangle}{\sqrt{p(0)}} = \frac{a}{|a|}|0\rangle, \tag{2.27}$$

$$\frac{M_1|\psi\rangle}{\sqrt{p(1)}} = \frac{b}{|b|}|1\rangle. \tag{2.28}$$

Ou seja, o estado $|\psi\rangle$ "colapsa" para um estado equivalente a $|0\rangle$ ou $|1\rangle$ após a medição.

Um caso especial de operadores de medida são as projeções, vejamos o exemplo a seguir.

Exemplo 2.4. *Seja M um operador hermitiano em um espaço de estados de um sistema sendo observado. Assuma que M tem decomposição espectral*

$$M = \sum_{i=1}^{n} iP_i, \tag{2.29}$$

de modo que P_i é projeção sobre o autoespaço de M associado ao autovalor i. Ao realizar a medição em $|\psi\rangle$, obtemos i com probabilidade

$$p(i) = \langle\psi|P_i|\psi\rangle. \tag{2.30}$$

Dado que o resultado i ocorre, o estado $|\psi'\rangle$ do sistema imediatamente após a medição é dado por

$$|\psi'\rangle = \frac{P_i|\psi\rangle}{\sqrt{p(i)}}. \tag{2.31}$$

Medições projetivas são bastante úteis pois simplificam alguns cálculos, por exemplo, o valor esperado (ou média) de um operador pode ser facilmente calculado. Por definição, a média de uma medida é dada por

$$E(M) = \sum_{i=1}^{n} ip(i) = \sum_{i=1}^{n} i \langle\psi|P_i|\psi\rangle = \langle\psi| \sum_{i=1}^{n} iP_i|\psi\rangle = \langle\psi|M|\psi\rangle . \tag{2.32}$$

O desvio padrão $\Delta(M)$ para o operador M é dado por

$$(\Delta M)^2 = \langle\psi| M^2 |\psi\rangle - \langle\psi|M|\psi\rangle^2 . \tag{2.33}$$

2.5 Postulado 4 - Sistemas Compostos

O espaço de estados de um sistema quântico composto é o produto tensorial dos espaços de estados dos sistemas componentes. Em particular, se dois sistemas físicos têm espaços de estados $\mathcal{H}_A$ e $\mathcal{H}_B$, então o espaço de estados do sistema composto é

$$\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B. \tag{2.34}$$

Além disso, se os sistemas A e B estão em estados $|\psi_A\rangle$ e $|\psi_B\rangle$, respectivamente, então o estado do sistema composto é

$$|\psi_{AB}\rangle = |\psi_A\rangle \otimes |\psi_B\rangle . \tag{2.35}$$

É importante ressaltar que nem todo estado em $\mathcal{H}_A \otimes \mathcal{H}_B$ pode ser escrito como um produto tensorial de estados dos subsistemas. Estados que não admitem essa forma são chamados de *estados emaranhados* (*entangled states*).

Exemplo 2.5. *Considere o estado de dois qubits*

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}\Big(|00\rangle + |11\rangle\Big) \in \mathcal{H}_A \otimes \mathcal{H}_B. \tag{2.36}$$

Suponha, por absurdo, que $|\beta_{00}\rangle$ pode ser escrito como um produto tensorial

$$|\beta_{00}\rangle = |\psi_A\rangle \otimes |\psi_B\rangle , \qquad |\psi_A\rangle \in \mathcal{H}_A, \;\; |\psi_B\rangle \in \mathcal{H}_B \tag{2.37}$$

onde

$$|\psi_A\rangle = a\,|0\rangle + b\,|1\rangle \qquad e \qquad |\psi_B\rangle = c\,|0\rangle + d\,|1\rangle , \tag{2.38}$$

com $a, b, c, d \in \mathbb{C}$. Assim,

$$|\psi_A\rangle \otimes |\psi_B\rangle = (ac)\,|00\rangle + (ad)\,|01\rangle + (bc)\,|10\rangle + (bd)\,|11\rangle . \tag{2.39}$$

Temos então

$$ac = \frac{1}{\sqrt{2}}, \qquad bd = \frac{1}{\sqrt{2}}, \qquad ad = 0, \qquad bc = 0. \tag{2.40}$$

Ou seja,

- *ou* $a = 0$ *ou* $d = 0$,
- *ou* $b = 0$ *ou* $c = 0$.

O que contradiz $ac = bd = \frac{1}{\sqrt{2}} \neq 0$. *Portanto,* $|\beta_{00}\rangle$ *não pode ser escrito como um produto tensorial de estados dos subsistemas* A *e* B.

O exemplo a seguir ilustra o uso de operadores em sistemas compostos.

Exemplo 2.6. *Considere um sistema composto por dois qubits,* A *e* B*, cujos espaços de estados são* $\mathcal{H}_A = \mathbb{C}^2$ *e* $\mathcal{H}_B = \mathbb{C}^2$*. O espaço de estados do sistema composto é dado por*

$$\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B = \mathbb{C}^2 \otimes \mathbb{C}^2. \tag{2.41}$$

Podemos tomar a base computacional deste sistema composto $B = \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ *e considere o estado produto*

$$|\psi_{AB}\rangle = |0\rangle_A \otimes |1\rangle_B = |01\rangle. \tag{2.42}$$

Considere agora a aplicação do operador de Pauli X *sobre o qubit* A*, e o operador de Hadamard sobre o qubit* B*. O operador total que atua no sistema composto é*

$$X \otimes H. \tag{2.43}$$

Aplicando esse operador ao estado do sistema, obtemos

$$\begin{aligned}(X \otimes H)|01\rangle &= (X|0\rangle) \otimes (H|1\rangle) && (2.44)\\ &= |1\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) && (2.45)\\ &= \frac{1}{\sqrt{2}}(|10\rangle - |11\rangle). && (2.46)\end{aligned}$$

Para sistemas compostos maiores é bastante comum indicar numericamente os sistemas componentes, assim, para um sistema composto de n subsistemas a aplicação do operador X na i-ésima componente é indicada por X_i.

3 Circuitos Quânticos

3.1 Introdução

Este capítulo é baseado nos capítulos 3 e 4 de Nielsen and Chuang [2010] e na documentação oficial do Qiskit IBM Quantum [2025c]; o objetivo é adaptar os conceitos clássicos de portas lógicas e circuitos para sistemas quânticos, além disso, é feita uma introdução ao Qiskit.

3.2 Circuitos

Vamos abordar primeiramente circuitos na computação clássica. Um circuito pode envolver várias entradas e saídas, vários fios e portas lógicas. Uma porta lógica é uma função $f : \{0,1\}^k \to \{0,1\}^l$ para algum número fixo k de bits de entrada e algum número fixo l de bits de saída. Por exemplo, a porta lógica NOT (figura 2) possui um bit de entrada e um bit de saída, esta porta lógica inverte o bit de entrada.

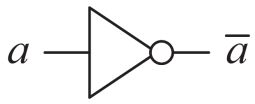

Figura 2 – Porta lógica NOT.

Existem várias outras portas lógicas elementares que são muito úteis para computação. A figura 3 apresenta algumas delas. A porta lógica AND tem saída igual a 1 se, e somente se, ambas as entradas são 1, a porta lógica OR tem saída 1 se, e somente se pelo menos uma das entradas é 1. XOR tem saída 1 se e somente se apenas uma das entradas tem valor 1. NAND e NOR são portas construídas de AND seguido de NOT e OR seguido de NOT, respectivamente.

Algumas outras operações bastante comuns em circuitos são a cópia de bits (FANOUT) e troca de valores entre dois bits CROSSOVER. Uma terceira operação bastante importante é a preparação de bits *ancilla* (bits auxiliares), para permitir espaço de trabalho extra durante a computação. Na figura 3 são dados alguns exemplos de circuitos bastante conhecidos que podem ser construídos com as portas lógicas elementares. O primeiro circuito é o meio somador (*half adder*), figura 4, que possui dois bits de entrada e produz como resultado a soma dos dois bits módulo 2 em conjunto com o *carry* (o bit "vai um").

Podemos construir um circuito somador completo (*full adder*) utilizando dois meio somadores, denotados por HA, em cascata (figura 5). O somador completo recebe três bits,

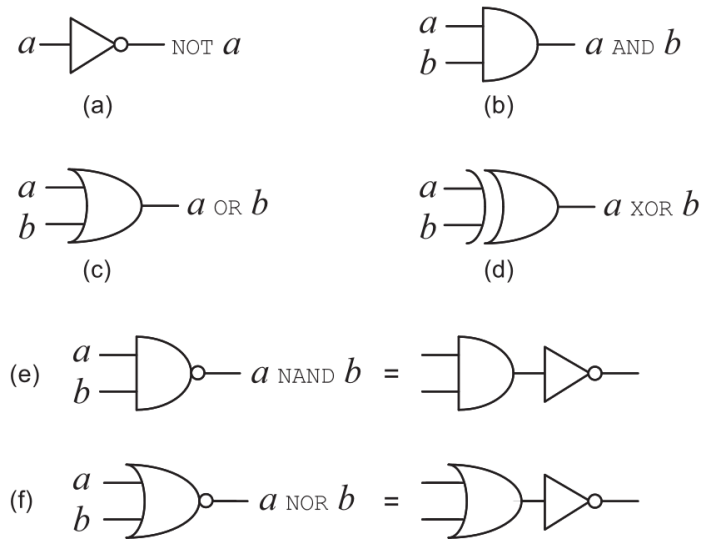

Figura 3 – Portas lógicas elementares.

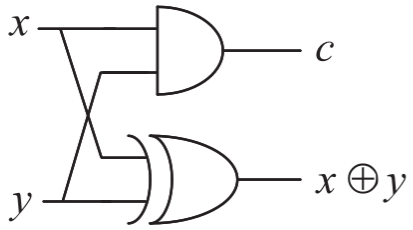

Figura 4 – Circuito Meio Somador.

x, y, e c. Os bits x e y são os valores que serão somados e c o bit carry da computação anterior. Este circuito tem como saída dois bits, o primeiro bit é a soma modulo dois de x, y, c dos três bits de entrada. O segundo bit, c', é o bit *carry*, que é definido como 1 se dois ou mais entradas são 1, caso contrário, seu valor é 0.

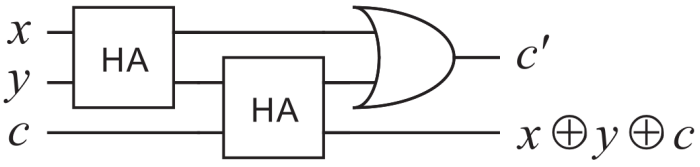

Figura 5 – Circuito Somador Completo.

Ao encadearmos vários circuitos somadores completos, obtemos um circuito para adição de inteiros com n-bits. Denotando o circuito somador completo por FA, a figura 6 exibe o circuito para adição de inteiros com três bits.

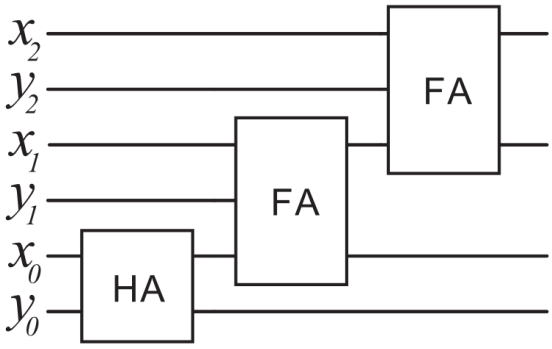

Figura 6 – Circuito Somador para inteiros de três bits.

3.3 Qubit

Vamos agora comparar os conceitos da computação clássica com a computação quântica. Na computação clássica o bit é a menor unidade de informação que pode ser armazenada, podendo assumir os valores 0 ou 1 e somente um valor por vez. Analogamente, na computação quântica temos o qubit, diferentemente do bit, o qubit pode armazenar estados além de $|0\rangle$ e $|1\rangle$, tais estados são chamados de superposições:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \qquad \text{com } |\alpha|^2 + |\beta|^2 = 1 \text{ e } \alpha, \beta \in \mathbb{C}. \tag{3.1}$$

Diferentemente dos bits, não podemos examinar o qubit e determinar seu estado quântico (veja seção 2.4), ou seja, os valores de α e β. Ao invés disso, realizarmos a medição obtemos, por exemplo, o resultado 0, com probabilidade $|\alpha|^2$ ou 1 com probabilidade $|\beta|^2$.

3.4 Múltiplos qubits

Se tivermos dois bits teremos quatro possíveis valores armazenados (ou estados): 00, 01, 10 e 11. Analogamente, para um sistema com dois qubits podemos ter também os estados $|00\rangle\,, |01\rangle\,, |10\rangle$ e $|11\rangle$, no entanto, podemos ter também superposições desses estados:

$$|\psi\rangle = \alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle\,. \tag{3.2}$$

Além disso, o resultado da medida de, por exemplo, $x = 00$ tem probabilidade $|\alpha_{00}|^2$.

Definição 3.1 (Base de Bell, Estados de Bell ou Pares EPR)**.** *Considere um sistema com 2 qubits associado ao espaço $C^2 \otimes C^2$ e defina os seguintes estados*

$$\begin{aligned} |\beta_{00}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \quad |\beta_{10}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \\ |\beta_{01}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \quad |\beta_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \end{aligned} \tag{3.3}$$

Estes quatro estados formam uma base para $C^2 \otimes C^2$ chamada de Base de Bell, estes quatro estados também são conhecidos como Estados de Bell ou Pares EPR.

3.4.1 Portas lógicas

Assim como um computador clássico construído por um circuito elétrico formado por fios e portas lógicas, podemos construir um computador quântico a partir de circuitos quânticos formados por fios e portas lógicas quânticas. A porta lógica NOT é definida por:

in	*out*
1	0
0	1

Em sistemas quânticos, podemos representar o operador NOT pela matriz

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{3.4}$$

Sendo $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ temos:

$$X\,|\psi\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}. \tag{3.5}$$

Outras portas lógicas quânticas importantes são o Z:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{3.6}$$

que quando aplicado a $|0\rangle$ mantém inalterado, e inverte o sinal de $|1\rangle$ para $-\,|1\rangle$, e o operador de Hadamard

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}; \tag{3.7}$$

este operador transforma $|0\rangle$ em $(|0\rangle + |1\rangle)/\sqrt{2}$ e transforma $|1\rangle$ em $(|0\rangle - |1\rangle)/\sqrt{2}$. A figura 7 ilustra a aplicação dos operadores X, Z e H.

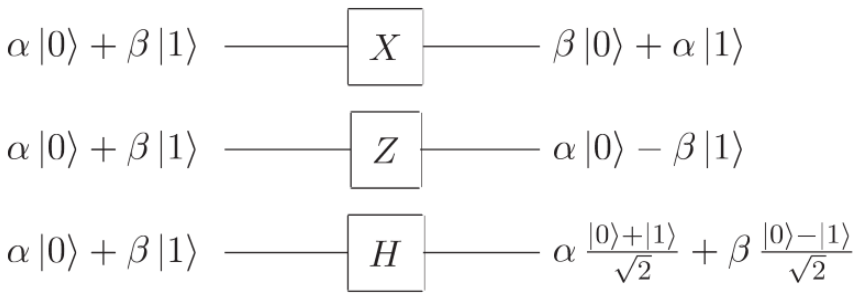

Figura 7 – Portas lógicas quânticas

De modo geral, portas lógicas quânticas podem ser descritas por matrizes unitárias (matrizes 2×2 no caso de um único qubit) e esta é a única restrição necessária para portas lógicas quânticas. Como matrizes unitárias são invertíveis temos que toda operação realizada por uma porta lógica quântica é reversível.

Podemos generalizar as portas lógicas quânticas para múltiplos qubits. No lado esquerdo da figura 8, são apresentadas algumas portas lógicas comuns da computação clássica. Um resultado bastante conhecido é de que a porta lógica NAND é universal [Mano and Ciletti, 2013, seção 3.6], ou seja, qualquer função booleana pode ser construída somente com combinações de portas lógicas NAND. No lado direito da mesma figura, vemos um protótipo de uma porta lógica quântica para múltiplos qubits chamado controlled-NOT ou CNOT. Esta porta lógica quântica possui dois qubits de entrada, conhecidos como qubits de controle e alvo. Em geral, portas lógicas de sistemas quânticos com múltiplos qubits podem ser construídas usando operações de produto tensorial para operadores (seção 1.4), bastando que o operador seja unitário.

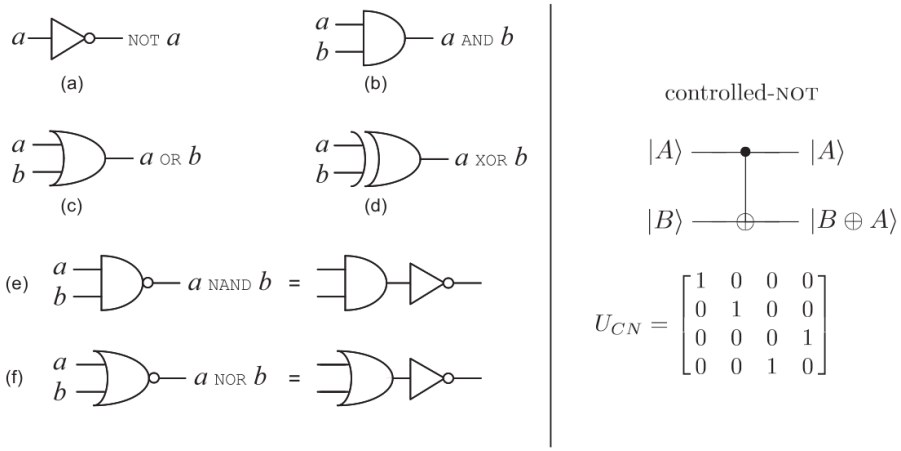

Figura 8 – Exemplos de portas lógicas clássicas e porta lógica quântica. No lado direito, $|A\rangle$ é o qubit de controle e $|B\rangle$ é o qubit alvo.

A atuação do CNOT, da figura 8, pode ser descrita como:

- Se o qubit de controle $|A\rangle$ é definido como 0, então o qubit alvo $|B\rangle$ é deixado com seu valor original.

$$|00\rangle \rightarrow |00\rangle$$
$$|01\rangle \rightarrow |01\rangle$$

- Se o qubit de controle é definido como 1, então o qubit alvo é invertido.

$$|10\rangle \rightarrow |11\rangle$$
$$|11\rangle \rightarrow |10\rangle$$

Outras portas lógicas quânticas bastante utilizadas na computação quântica são listadas a seguir:

- Porta S (*phase gate*)
$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \tag{3.8}$$

- Porta T ($\pi/8$ *gate*)
$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}. \tag{3.9}$$

- Porta R_x (rotação no eixo x na esfera de Bloch)
$$R_x(\theta) = e^{-i\frac{\theta}{2}X} = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i\,\mathrm{sen}\left(\frac{\theta}{2}\right) \\ -i\,\mathrm{sen}\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}. \tag{3.10}$$

- Porta R_y (rotação no eixo y na esfera de Bloch)
$$R_y(\theta) = e^{-i\frac{\theta}{2}Y} = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\,\mathrm{sen}\left(\frac{\theta}{2}\right) \\ \mathrm{sen}\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}. \tag{3.11}$$

- Porta R_z (rotação no eixo z na esfera de Bloch)
$$R_z(\theta) = e^{-i\frac{\theta}{2}Z} = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}. \tag{3.12}$$

- Porta Tofoli ou CCNOT, que similar ao CNOT atua em múltiplos qubits sendo os dois primeiros qubits de controle e o terceiro o qubit-alvo. Da porta Tofoli podemos generalizar o conceito para um C^{n}NOT ou CNOT com n qubits de controle e um qubit-alvo.

Assim, como na computação clássica, podemos nos perguntar sobre a existência de um conjunto finito, preferencialmente pequeno, de portas lógicas quânticas, que combinadas, permitam realizar qualquer computação. Para computação quântica, dizemos que um conjunto de portas lógicas é universal se qualquer operação unitária puder ser arbitrariamente aproximada por um circuito quântico contendo somente tais portas. Dessa forma, segundo [Nielsen and Chuang, 2010, seção 4.5] o conjunto $\{H, T, CNOT\}$ é universal.

3.4.2 Medidas em bases arbitrárias

A base $\{|0\rangle, |1\rangle\}$ é chamada de base computacional. Quando apresentamos o qubit mencionamos que o qubit
$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle, \tag{3.13}$$

pode estar em qualquer combinação linear unitária destes estados e, ao realizar a medida, temos como resultado 0 com probabilidade $|\alpha|^2$ e 1 com probabilidade $|\beta|^2$. No entanto, podemos tomar qualquer outra base que nos seja conveniente. Por exemplo, podemos tomar $\{|+\rangle, |-\rangle\}$ com $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ e $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$, assim

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \frac{\alpha+\beta}{\sqrt{2}}|+\rangle + \frac{\alpha-\beta}{\sqrt{2}}|-\rangle . \tag{3.14}$$

Deste modo, ao realizar a medida, relativa à base $\{|+\rangle, |-\rangle\}$, obteremos o resultado "+" com probabilidade $|\alpha+\beta|^2/2$ e o resultado "−" com probabilidade $|\alpha-\beta|^2/2$. De modo geral, dada qualquer base $\{|a\rangle$ e $|b\rangle\}$ ortonormal e um qubit arbitrário $|\psi\rangle = \alpha|a\rangle + \beta|b\rangle$ podemos realizar a medida e obter os resultados a e b com probabilidades $|\alpha|^2$ e $|\beta|^2$ respectivamente.

3.5 Circuitos quânticos

Vimos nas seções anteriores alguns circuitos quânticos simples (formados por uma única porta lógica quântica). A figura 9 mostra um circuito com três portas lógicas quânticas, cada linha do circuito representa um fio no circuito quântico (que não necessariamente corresponde a um fio físico, podendo ser, por exemplo, a passagem de tempo ou movimentação de uma partícula no espaço). Tomando $|ab\rangle = |a,b\rangle$ (a adição da vírgula é apenas por questões de legibilidade), na base computacional, e sendo $\oplus$ a operação de adição módulo 2, o circuito tem o seguinte comportamento

$$\begin{aligned}
|a,b\rangle &\to |a, a\oplus b\rangle \\
&\to |a\oplus(a\oplus b), a\oplus b\rangle = |b, a\oplus b\rangle , && \text{pois } a\oplus a\oplus b = 0+b = b \\
&\to |b, (a\oplus b)\oplus b\rangle = |b,a\rangle , && \text{pois } a\oplus b\oplus b = a\oplus 0 = a.
\end{aligned}$$

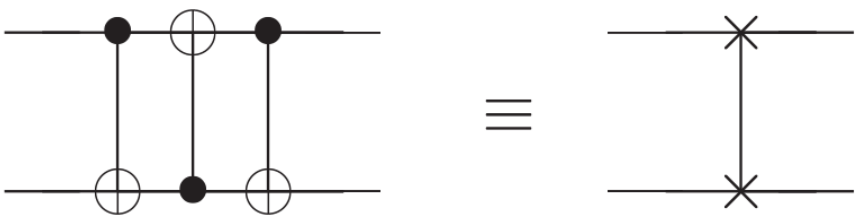

Figura 9 – Circuito quântico de troca de qubits. No lado esquerdo, é apresentado um circuito com três CNOTs ao lado direito é apresentada a notação para tal circuito.

Uma diferença fundamental entre circuitos clássicos e circuitos quânticos está relacionada às restrições impostas pela reversibilidade das operações permitidas. Enquanto

circuitos clássicos admitem estruturas como laços e cópia arbitrária de bits, tais recursos não são compatíveis com a mecânica quântica. Essas limitações explicitamente discutidas por Nielsen e Chuang:

> There are a few features allowed in classical circuits that are not usually present in quantum circuits. First of all, we don't allow 'loops', that is, feedback from one part of the quantum circuit to another; we say the circuit is acyclic. Second, classical circuits allow wires to be 'joined' together, an operation known as FANIN, with the resulting single wire containing the bitwise OR of the inputs. Obviously this operation is not reversible and therefore not unitary, so we don't allow FANIN in our quantum circuits. Third, the inverse operation, FANOUT, whereby several copies of a bit are produced is also not allowed in quantum circuits. In fact, it turns out that quantum mechanics forbids the copying of a qubit, making the fanout operation impossible! [Nielsen and Chuang, 2010, p. 23]

Em especial, para o caso de laços em computação quântica, há tentativas de formalizar sua construção como em Aaronson [2006] e Banerjee and Pathak [2009].

Uma alternativa ao uso de laços em circuitos quânticos, é a construção prévia do circuito concatenando partes que se repetem, previamente à execução do circuito. Nesses casos, um conceito bastante importante para analisar o desempenho do algoritmo é a *profundidade do circuito*. A profundidade de um circuito quântico é o comprimento do maior caminho entre o início e o fim do circuito, onde cada porta lógica quantica conta como uma unidade. Casos de portas aplicadas em qubits disjuntos (portas que podem ser aplicadas em paralelo) também são consideradas como uma unidade.

Uma notação bastante conveniente para circuitos é a dada a seguir. Suponha U uma matriz unitária qualquer que atua sobre um número n de qubits; como vimos, U é uma porta lógica quântica para esses qubits. Podemos então definir a porta lógica controlled-U, tal porta lógica tem um único qubit de controle e n qubits alvo. Se o qubit de controle é definido em 0, então nada acontece aos qubits alvo, se é definido como 1, então U é aplicado aos qubits alvo. A figura 10 apresenta um diagrama para a porta lógica controlled-U.

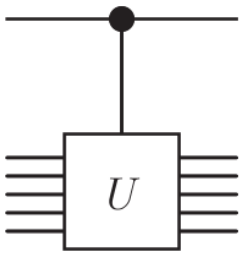

Figura 10 – Porta lógica controlled-U. A linha vertical representa o qubit de controle e as linhas horizontais os qubits-alvo.

Alguns exemplos bastante comuns de portas controladas são:

Porta CNOT ou CX

$$\mathrm{CX} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{3.15}$$

Porta CZ

$$\mathrm{CZ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \tag{3.16}$$

Uma outra operação muito importante é a medida, ao qual é representada por um símbolo de medidor analógico como mostrado na figura 11. Como mencionado anteriormente este operador converte um único qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ em um bit clássico M, com probabilidades $|\alpha|^2$ ou $|\beta|^2$ se o resultado é 0 ou 1 respectivamente.

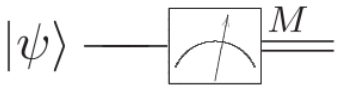

Figura 11 – Símbolo de medida para circuitos .

Mencionamos que a cópia de qubits não é possível, vamos ver um exemplo que ilustra melhor esta situação.

3.6 Cópia de Qubits

Considere a tarefa de copiar um bit clássico. Podemos realizar essa tarefa utilizando uma porta lógica CNOT conforme figura 12. Essa porta lógica recebe o bit x que queremos copiar e um bit $y = 0$; as saídas são dois bits, um simplesmente conectado na entrada e outro com o operador CNOT aplicado na entrada em $y = 0$.

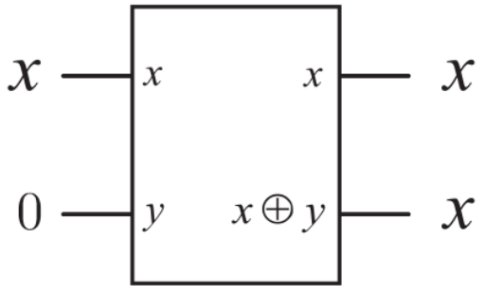

Figura 12 – Circuito para cópia do bit x.

Suponha agora que tentemos copiar o qubit $|\psi\rangle = a\,|0\rangle + a\,|1\rangle$ da mesma forma que fizemos no caso do bit clássico. Escrevemos a entrada como

$$\Big(a\,|0\rangle + b\,|1\rangle\Big)\,|0\rangle = a\,|00\rangle + b\,|10\rangle\,, \tag{3.17}$$

a função do CNOT é inverter o segundo qubit caso o primeiro qubit seja igual a 1 (figura 13), assim, a saída do CNOT será $a\,|00\rangle + b\,|11\rangle$ que, assumindo que a clonagem é possível, deve ser igual a $|\psi\rangle\,|\psi\rangle$. No entanto,

$$a\,|00\rangle + b\,|11\rangle \neq a^2\,|00\rangle + ab\,|01\rangle + ab\,|10\rangle + b^2\,|11\rangle = |\psi\rangle\,|\psi\rangle\,. \tag{3.18}$$

para complexos a, b quaisquer. No entanto, note que se $|\psi\rangle = |0\rangle$ ou $|\psi\rangle = |1\rangle$ o resultado vale.

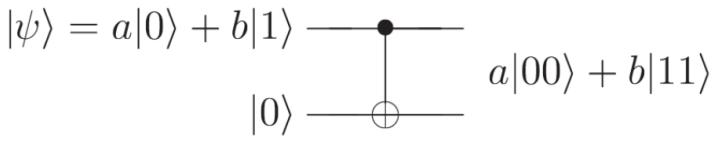

Figura 13 – Tentativa de cópia do qubit $|\psi\rangle$.

Este resultado nos dá uma indicação de que a cópia de qubits pode ser feita da mesma forma como se faz com bits classícos. No entanto, como apresentando no próximo resultado não é possível realizar cópia de qubits desconhecidos.

Teorema 3.2 (Teorema da não clonagem para um qubit)**.** *Não existem $|s\rangle \in \mathbb{C}^2$ e $U : \mathbb{C}^4 \to \mathbb{C}^4$, um operador unitário, tais que $U(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle$ para todo $|\psi\rangle \in \mathbb{C}^2$. Em outras palavras, não é possível construir um circuito quântico que realize a cópia de um qubit arbitrário $|\psi\rangle$.*

Demonstração. Dado um sistema quântico de dois qubits, nomeados A e B, A é a entrada, que inicialmente assume o estado $|\psi\rangle$ e B a saída para a cópia, inicialmente no estado $|s\rangle$. O estado inicial do sistema é dado por

$$|\psi\rangle \otimes |s\rangle \tag{3.19}$$

Suponha que seja possível realizar a cópia, então existe um operador unitário U tal que

$$|\psi\rangle \otimes |s\rangle \xrightarrow{U} U(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle\,. \tag{3.20}$$

Se a cópia funciona para os estados $|\psi\rangle$ e $|\varphi\rangle$, temos

$$U(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle \tag{3.21}$$

$$U(|\varphi\rangle \otimes |s\rangle) = |\varphi\rangle \otimes |\varphi\rangle\,. \tag{3.22}$$

Ao tomar o produto interno das equações acima temos

$$\langle U(|\psi\rangle \otimes |s\rangle)|U(|\varphi\rangle \otimes |s\rangle)\rangle = \langle(|\psi\rangle \otimes |s\rangle)|(|\varphi\rangle \otimes |s\rangle)\rangle \tag{3.23}$$
$$= \langle\psi|\varphi\rangle \langle s|s\rangle \tag{3.24}$$
$$= \langle\psi|\varphi\rangle \tag{3.25}$$
$$= \langle(|\psi\rangle \otimes |\varphi\rangle)|(|\psi\rangle \otimes |\varphi\rangle)\rangle \tag{3.26}$$
$$= \langle\psi|\varphi\rangle^2, \tag{3.27}$$

portanto,

$$\langle\psi|\varphi\rangle = \langle\psi|\varphi\rangle^2 \implies \langle\psi|\varphi\rangle = 0 \text{ ou } \langle\psi|\varphi\rangle = 1, \tag{3.28}$$

que só é possível se $|\varphi\rangle = |\psi\rangle$ ou se $|\varphi\rangle$ e $|\psi\rangle$ são ortogonais. Logo, a cópia de estados quaisquer não é possível. ■

Exemplo 3.3. *Considere o circuito da figura 14 que transforma os quatro estados da base computacional* $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ *nos Estados de Bell ou Pares EPR conforme definição 3.1. Primeiramente, aplica-se o operador de Hadamard e, em seguida, a porta lógica* CNOT *com o qubit* 1 *sendo o qubit de controle e o qubit* 2 *o qubit-alvo. Por exemplo, para* $|00\rangle$ *temos*

$$|00\rangle \xrightarrow{H\otimes I} \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |0\rangle \tag{3.29}$$
$$\xrightarrow{CNOT_{1,2}} \frac{|00\rangle + |11\rangle}{\sqrt{2}} = |\beta_{00}\rangle \tag{3.30}$$

In	Out
$\|00\rangle$	$(\|00\rangle + \|11\rangle)/\sqrt{2} \equiv \|\beta_{00}\rangle$
$\|01\rangle$	$(\|01\rangle + \|10\rangle)/\sqrt{2} \equiv \|\beta_{01}\rangle$
$\|10\rangle$	$(\|00\rangle - \|11\rangle)/\sqrt{2} \equiv \|\beta_{10}\rangle$
$\|11\rangle$	$(\|01\rangle - \|10\rangle)/\sqrt{2} \equiv \|\beta_{11}\rangle$

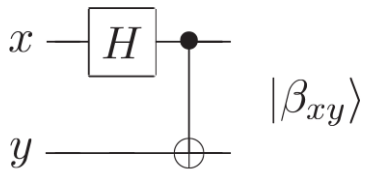

Figura 14 – Circuito para geração de estados de Bell.

3.7 Teleporte Quântico

De posse do conhecimento de circuitos quânticos podemos compreender o Teleporte Quântico. O teleporte quântico é uma técnica que permite *teleportar* estados mesmo na ausência de um canal de comunicação quântica ligando o remetente ao destinatário. O cenário é dado a seguir:

Alice e Bob se encontraram há muito tempo mas agora vivem longe um do outro. Enquanto viviam juntos eles geraram um par EPR (assumimos que seja β_{00}, conforme definição 3.1), cada um ficando com um qubit quando se separaram. Anos depois, Bob está escondido e a missão de Alice é entregar um qubit $|\psi\rangle$ para Bob. Ela não sabe o estado do qubit e, além disso, pode enviar somente informação clássica para Bob.

Como Alice não sabe qual é o estado $|\psi\rangle$ e as leis da mecânica quântica impedem que ela determine o estado se ela possuir somente uma cópia de $|\psi\rangle$, Alice não pode realizar a cópia do estado (teorema 3.2), muito menos, definir com precisão o estado somente com uma medição. Após a medição, Alice saberá o resultado com determinada probabilidade e, nesse processo, o estado $|\psi\rangle$ é alterado, conforme equação (2.21). Mesmo no cenário onde Alice conhece o estado, ela necessitaria de uma eternidade para descrever o estado para Bob, desde que $|\psi\rangle$ assume valores em um espaço contínuo.

Felizmente, Alice compartilha um par EPR com Bob e pode utilizá-lo para enviar o estado $|\psi\rangle$ para Bob. Alice une o qubit $|\psi\rangle$ com sua metade do par EPR, e então aplica uma medida aos dois qubits, obtendo um de quatro possíveis valores clássicos 00, 01, 10 e, 11. Ela então envia o resultado para Bob e, dependendo do resultado recebido, Bob executa uma de quatro operações possíveis na sua metade do par EPR. O circuito quântico da figura 15 ilustra o funcionamento do teleporte quântico. A descrição detalhada do procedimento é dada a seguir.

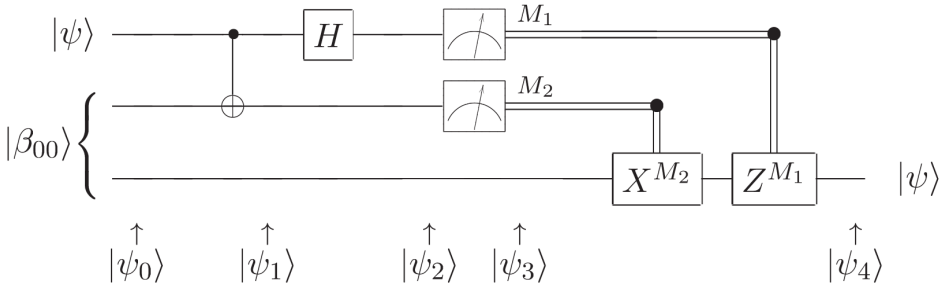

Figura 15 – Circuito quântico para teleporte de um qubit. As duas linhas superiores indicam o sistema de Alice, enquanto a linha de baixo indica o sistema de Bob.

O estado a ser teleportado é $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$, com a, b coeficientes desconhecidos. Usando o par EPR $|\beta_{00}\rangle$, o estado de entrada no circuito é

$$|\psi_0\rangle = |\psi\rangle \otimes |\beta_{00}\rangle \tag{3.31}$$

$$= \frac{1}{\sqrt{2}}\Big(a\,|0\rangle \otimes (|00\rangle + |11\rangle) + b\,|1\rangle \otimes (|00\rangle + |11\rangle)\Big); \tag{3.32}$$

convencionamos que os primeiros dois qubits (a partir da esquerda) pertencem à Alice e o terceiro ao Bob. Alice então aplica o operador CNOT em seus qubits com $|\psi\rangle$ o qubit de

controle e sua metade do par EPR como qubit alvo, obtendo

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}\Big(a\,|0\rangle \otimes (|00\rangle + |11\rangle) + b\,|1\rangle \otimes (|10\rangle + |01\rangle)\Big), \tag{3.33}$$

em seguida, aplica o operador de Hadamard no primeiro qubit, obtendo

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{2}\Big(a(|0\rangle + |1\rangle) \otimes (|00\rangle + |11\rangle) + b(|0\rangle - |1\rangle) \otimes (|10\rangle + |01\rangle)\Big) \\ &= \frac{1}{2}\Big(\,|00\rangle \otimes (a\,|0\rangle + b\,|1\rangle) + |01\rangle \otimes (a\,|1\rangle + b\,|0\rangle) \\ &+ |10\rangle \otimes (a\,|0\rangle - b\,|1\rangle) + |11\rangle \otimes (a\,|1\rangle - b\,|0\rangle)\Big). \end{aligned} \tag{3.34}$$

A equação (3.34) está organizada em quatro termos. O primeiro termo tem os qubits de Alice no estado $|00\rangle$ e o qubit de Bob no estado $a\,|0\rangle + b\,|1\rangle = |\psi\rangle$. Se Alice executa a medida, relativa aos primeiros dois estados, e obtém o resultado 00 então o sistema de Bob estará no estado $|\psi\rangle$. Analogamente, podemos ler o estado pós-medida de Bob, dado o resultado da medida de Alice

$$00 \mapsto |\psi_3(00)\rangle = a\,|0\rangle + b\,|1\rangle \tag{3.35}$$

$$01 \mapsto |\psi_3(01)\rangle = a\,|1\rangle + b\,|0\rangle \tag{3.36}$$

$$10 \mapsto |\psi_3(10)\rangle = a\,|0\rangle - b\,|1\rangle \tag{3.37}$$

$$11 \mapsto |\psi_3(11)\rangle = a\,|1\rangle - b\,|0\rangle \tag{3.38}$$

Dependendo do resultado da medida de Alice, o qubit de Bob estará em um de quatro possíveis estados. No entanto, para saber qual estado é necessário que Bob seja informado do resultado da medida de Alice. Este fato é justamente o que impede que o teleporte quântico seja utilizado para transmitir informação mais rápido que a velocidade da luz.

Uma vez que Bob tem ciência do resultado de Alice, ele pode:

1. 00: não precisa fazer nada pois seu estado já é $|\psi\rangle$.
2. 01: aplica o operador X.
3. 10: aplica o operador Z.
4. 11: aplica o operador X e, em seguida, o operador Z.

Após aplicar estes operadores, Bob tem em suas mãos o qubit $|\psi\rangle$ que estava em posse de Alice.

3.8 Introdução ao Qiskit

3.8.1 Introdução

O *Qiskit* é um framework de código aberto desenvolvido pela IBM para programação, simulação e execução de algoritmos quânticos Qiskit Community [2017]. Ele permite que a criação de circuitos quânticos abstratos, exibição gráfica, otimização e execução em hardware tanto em simuladores clássicos quanto em dispositivos quânticos reais disponibilizados pela IBM.

3.8.2 Modelo computacional

O Qiskit adota o modelo de computação baseado em circuitos. Nesse modelo,

- estados quânticos são representados por qubits,
- portas quânticas representadas por operadores unitários,
- medições descritas por operadores de projeção.

Matematicamente, um circuito quântico atua sobre um espaço de Hilbert complexo $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$, onde n é o número de qubits do sistema.

3.8.3 Recursos do Qiskit

Segundo a documentação do Qiskit IBM Quantum [2025e] alguns dos recursos mais úteis são:

- **Ferramentas de construção de circuitos** (`qiskit.circuit`): utilizadas para inicializar e manipular registradores, circuitos, instruções, portas, parâmetros e objetos de fluxo de controle.
- **Biblioteca de circuitos** (`qiskit.circuit.library`): uma vasta gama de circuitos, instruções e portas, que constituem os principais blocos de construção para computações quânticas baseadas em circuitos.
- **Biblioteca de informações quânticas** (`qiskit.quantum_info`): um conjunto de ferramentas para trabalhar com estados quânticos, operadores e canais, utilizando cálculos exatos (sem ruído de amostragem). Esse módulo pode ser utilizado para especificar observáveis de entrada e analisar a fidelidade dos resultados obtidos a partir das consultas de primitivas.

- **Transpilador** (`qiskit.transpiler`): responsável por transformar e adaptar circuitos quânticos para atender à topologia específica do dispositivo e otimizar a execução em unidades de processamento quântico (QPUs) reais.
- **Primitivas** (`qiskit.primitives`): módulo que contém as definições básicas e implementações de referência das primitivas `Sampler` e `Estimator`, a partir das quais diferentes fornecedores de hardware quântico podem derivar suas próprias implementações.

3.8.4 Circuitos Quânticos

No Qiskit, um circuito quântico é representado pela classe `QuantumCircuit`. Um circuito consiste em registradores quânticos e clássicos, bem como uma sequência ordenada de operações.

Formalmente, um circuito quântico implementa uma transformação unitária

$$U = U_k \cdots U_2 U_1,$$

onde cada U_i corresponde a uma porta lógica quântica aplicada em um ou mais qubits. Para acelerar a construção de circuitos o Qiskit fornece implementações das principais portas quânticas, como:

- Portas de um qubit: Operadores de Pauli X, Y e Z (equação (1.3)), H (exemplo 2.2), S (equação (3.8)), T (equação (3.9)) entre outras.
- Portas controladas: CX (equação (3.15)), CZ (equação (3.16)) entre outras.
- Outras portas lógicas IBM Quantum [2025b] e a possibilidade de construir operadores quaisquer IBM Quantum [2025d].

Cada porta é representada por um operador unitário que atua sobre o espaço de estados do sistema.

3.8.5 Medição

No Qiskit, a medição padrão de qubits é implementada como uma medição projetiva na base computacional, por exemplo, para o caso de um qubit

$$\{|0\rangle, |1\rangle\},$$

os operadores de medida associados a essa medição são as projeções

$$P_0 = |0\rangle\langle 0|, \qquad P_1 = |1\rangle\langle 1|,$$

que satisfazem

$$P_0^\dagger P_0 + P_1^\dagger P_1 = I.$$

Veja exemplo 2.4.

Assim, ao medir um qubit no Qiskit, o estado $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ colapsa para $|0\rangle$ com probabilidade $|\alpha|^2$ ou para $|1\rangle$ com probabilidade $|\beta|^2$, exatamente como apresentado no capítulo 2. O resultado da medição é armazenado em registradores clássicos. Essa separação explícita entre registradores quânticos e clássicos fornece uma implementação direta do postulado de medição.

3.8.6 Simulação e Execução

O Qiskit permite a execução de circuitos em:

- **Simuladores clássicos**: úteis para depuração e estudo teórico. Permite a execução de algoritmos quânticos em processadores clássicos, seja em CPUs ou GPUs.
- **Computadores quânticos reais**: sujeitos a ruído e erros experimentais, úteis para circuitos com maior número de qubits. De acordo com Kim et al. [2023] processadores quânticos com 100 qubits ou mais são suficientes para exploração científica, tornando a computação quântica útil (*quantum utility*) de um ponto de vista prático. Tal utilidade está longe de ser uma vantagem quântica (*quantum advantage*), isto é, o caso onde um computador quântico é capaz de resolver um problema específico mais rápido e com maior precisão do que o melhor computador clássico possível. No entanto, os QPUs abrem caminho para o uso da computação quântica em outras áreas do conhecimento tais como química e biologia.

3.8.7 Instalação e Utilização

O Qiskit é disponibilizado como um pacote Python, para instruções oficiais de instalação consulte a documentação da IBM IBM Quantum Platform [2025]. Para que seja possível realizar execuções em hardware real é necessário ter um IBMid e utilizar a chave de API disponibilizada na plataforma. Para realizar experimentos iniciais a IBM permite uso de 10 minutos por mês sem custo, consulte as informações sobre planos na documentação oficial da IBM IBM Quantum [2025a].

3.8.8 Exemplos de Circuitos

A seguir são apresentados alguns exemplos de circuitos construídos utilizando o Qiskit. Podemos criar um circuito informando a quantidade de qubits e bits utilizando a classe `QuantumCircuit`. Todos os qubits são inicializados no estado $|0\rangle$, além disso, o Qiskit utiliza o sistema *little-endian*, em que os qubits são numerados da direita para a

esquerda, $|0_{n-1}\ldots 0_0\rangle$, esta convenção é muito comum em implementações computacionais mas pode gerar um pouco de confusão pois, em geral, na matemática é comum indexar de 1 a n e da esquerda para a direita. A figura 16 apresenta a criação de um circuito com dois qubits e três bits clássicos.

```
1 from qiskit import QuantumCircuit
2
3 qc = QuantumCircuit(2, 3)
```

Figura 16 – Exemplo de criação de circuitos em Qiskit.

Um exemplo de aplicação de porta lógica e realização de medida pode ser visto na figura 17. Como esperado, ao aplicar o operador de Hadamard no qubit 0 com o valor $|0\rangle$ obtemos uma superposição, cujas probabilidades de obtermos 0 ou 1 são ambas 1/2. A figura 18 mostra o efeito da instrução `qc.draw` ao final da figura 17.

```
1 from qiskit import QuantumCircuit
2
3 qc = QuantumCircuit(2, 3)
4 qc.h(0)
5 qc.draw(output="mpl")
```

Figura 17 – Exemplo de aplicação de operador no Qiskit.

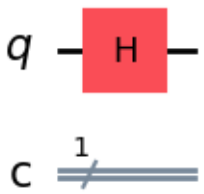

Figura 18 – Ilustração de instrução `draw`.

Um exemplo mais completo é a implementação do teleporte quântico visto na seção 3.7. A figura 19 apresenta o código de geração do par EPR de Alice e Bob. As linhas de 1 até 9 criam o circuito com o qubit que será teleportado `secret`, os qubits de Alice e Bob bem como e três registradores clássicos para obtenção dos valores de medida. As linhas 11 e 12 criam o par EPR com os qubits de Alice e Bob. A linha 14 tem a instrução `qc.barrier` que serve apenas para separação visual ao desenhar o circuito.

A figura 20 apresenta a geração do estado segredo e a aplicação de operações e medições de Alice. A linha 16 até a linha 19 o estado a ser teleportado que inicialmente possui valor $|0\rangle$ recebe um valor aleatório, a instrução `qc.u` permite rotacionar o qubit

```
1  from qiskit import ClassicalRegister, QuantumCircuit, QuantumRegister
2  import numpy as np
3
4  secret = QuantumRegister(1, "Q")
5  Alice = QuantumRegister(1, "A")
6  Bob = QuantumRegister(1, "B")
7
8  cr = ClassicalRegister(3, "c")
9  qc = QuantumCircuit(secret, Alice, Bob, cr)
10
11 qc.h(Alice)
12 qc.cx(Alice, Bob)
13
14 qc.barrier()
```

Figura 19 – Criação do par EPR de Alice e Bob.

na esfera de Bloch tanto no eixo z quanto no plano xy. As linhas 28 e 29 representam as medições de Alice.

```
16 np.random.seed(100)
17 theta = np.random.uniform(0.0, 1.0) * np.pi  # [0, pi]
18 varphi = np.random.uniform(0.0, 2.0) * np.pi  # [0 2*pi]
19 qc.u(theta, varphi, 0.0, secret)
20
21 qc.barrier()
22
23 qc.cx(secret, Alice)
24 qc.h(secret)
25
26 qc.barrier()
27
28 qc.measure(Alice, cr[1])
29 qc.measure(secret, cr[0])
30
31 with qc.if_test((cr[1], 1)):
32     qc.x(Bob)
33 with qc.if_test((cr[0], 1)):
34     qc.z(Bob)
35
36 qc.barrier()
37
38 qc.u(theta, varphi, 0.0, Bob).inverse()
39 qc.measure(Bob, cr[2])
40
41 qc.draw(output="mpl")
```

Figura 20 – Geração do estado segredo e manipulação de estados por Alice.

A figura 21 apresenta as manipulações que Bob deve fazer dependendo dos valores obtidos nos registradores $c[0]$ e $c[1]$. Bob deverá aplicar os operadores X, Z ou X e Z em seguida, ou não aplicar nenhum operador (linhas 31 até 34). A partir deste ponto, o qubit de Bob contém o estado segredo; no entanto, como não é possível verificar o valor de um

qubit diretamente, fazemos uma manipulação de estado. A linha 38 aplica a inversa da operação `qc.u` que foi definida na linha 19 e foi usada para criar o estado segredo. Esta operação adicional vai "resetar" o estado de Bob para o estado original no qubit `secret`, nomeadamente $|0\rangle$. A figura 22 apresenta o diagrama completo do circuito.

```
31 with qc.if_test((cr[1], 1)):
32     qc.x(Bob)
33 with qc.if_test((cr[0], 1)):
34     qc.z(Bob)
35
36 qc.barrier()
37
38 qc.u(theta, varphi, 0.0, Bob).inverse()
39 qc.measure(Bob, cr[2])
40
41 qc.draw(output="mpl")
```

Figura 21 – Manipulações de Bob.

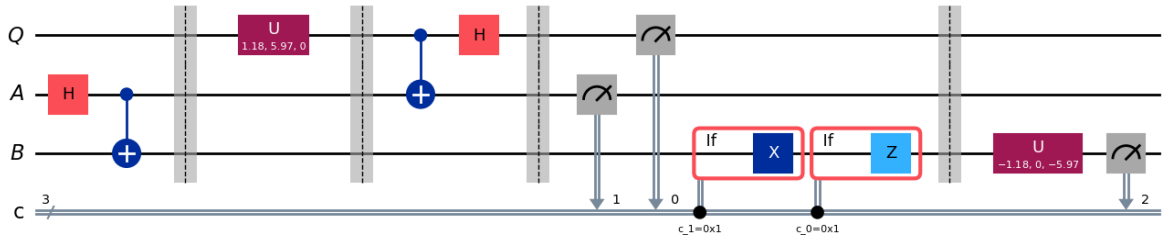

Figura 22 – Diagrama do circuito para o teleporte quântico.

Para finalizar o exemplo, podemos nos perguntar se o registrador que amazena o valor da medição no qubit de Bob realmente possui o valor 0. A figura 23 apresenta o código de simulação e a figura 24 mostra o histograma obtido, ao simular 10000 vezes; veja que os bits das posições 0 e 1 possuem igual chance de assumir os valores 1 ou 0. No entanto, o bit 2 que armazena o resultado da medição do qubit de Bob (após assumir o valor do qubit de segredo) apresenta sempre o valor 0 indicando que, de fato, antes de Bob aplicar a operação inversa à operação que produziu o estado segredo, Bob tinha o estado armazenado em seu qubit.

```
from qiskit import QuantumCircuit, transpile
from qiskit_aer import AerSimulator
from qiskit.visualization import plot_histogram

simulator = AerSimulator()
qc = transpile(qc, simulator)

result = simulator.run(qc, shots=10000).result()
counts = result.get_counts(qc)
plot_histogram(counts, title='Resultados de Medida')
```

Figura 23 – Circuito para simulação do teleporte quântico.

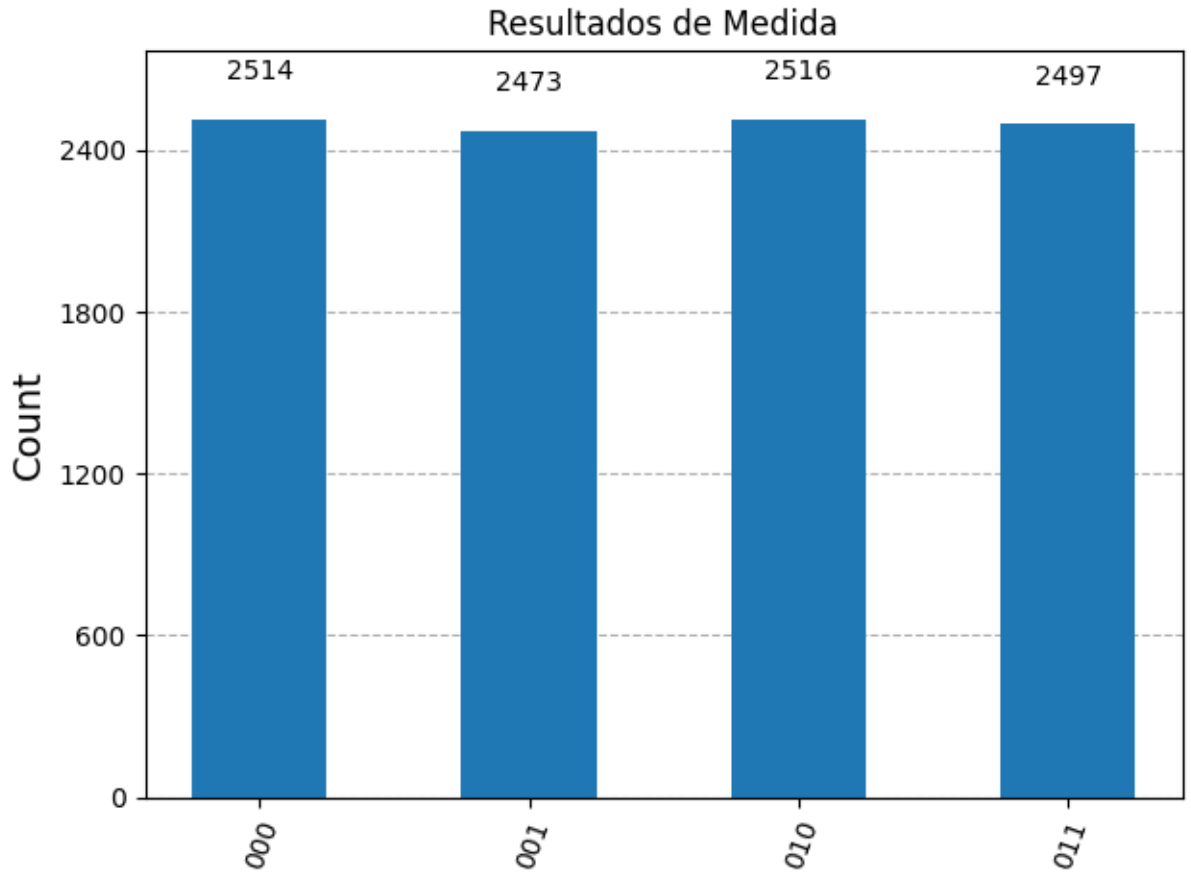

Figura 24 – Histograma para simulação de medições no circuito do teleporte quântico.

4 Exemplos de algoritmos quânticos

4.1 Introdução

Neste capítulo será feito o estudo de dois algoritmos quânticos, o algoritmo de Grover e o Algoritmo Quântico de Otimização Aproximada (QAOA). O algoritmo de Grover é um algoritmo para busca não estruturada em um conjunto de dados que utiliza uma função oráculo para avaliar o valor (ou valores) procurados, publicado em 1996 Grover [1996]. A vantagem deste algoritmo é que é exigido somente $O(\sqrt{N})$ avaliações do oráculo para encontrar os valores, onde N é a cardinalidade do domínio da busca. A busca não estruturada clássica possui complexidade $O(N)$, o algorítmo de Grover oferece um ganho bastante expressivo em relação aos algoritmos clássicos. O segundo algoritmo, QAOA, publicado em Farhi et al. [2014], é um algoritmo quântico utilizado para otimização combinatória, com o objetivo de encontrar soluções aproximadas, podendo a precisão ser aumentada com um aumento linear no tamanho do circuito quântico. O QAOA é de grande interesse pois oferece uma alternativa quântica aos algoritmos clássicos para busca de soluções aproximadas em problemas combinatórios. Alguns exemplos de problemas que podem ser mapeados como problemas de otimização combinatória e pertencem à classe NP-HARD são o Problema do Caixeiro Viajante, Problema da Mochila, Coloração de Grafos, Córte Máximo em grafos entre outros. Neste capítulo, o QAOA será aplicado para buscar de corte máximo aproximado em grafos.

4.2 Algoritmo de Grover (busca não-estruturada)

4.2.1 Problema de busca

Seja $N = 2^n$ e considere o conjunto de estados computacionais $\{x : x \in \{0,1\}^n\}$. Queremos encontrar um elemento marcado $x^\star$ por uma função booleana

$$f : \{0,1\}^n \to \{0,1\}$$

que satisfaça $f(x^\star) = 1$ e $f(x) = 0$ para $x \neq x^\star$ (caso de uma única solução). A suposição é que nós não temos nenhuma informação sobre a função f, mas é permitido calcular $f(x)$ para todo elemento x. O problema é encontrar um único $x^\star$ tal que $f(x^\star) = 1$. A Computação de $f(x)$ pode ser realizada pelo operador unitário

$$U_f \left|x\right\rangle = (-1)^{f(x)} \left|x\right\rangle . \tag{4.1}$$

O operador U_f é um operador do espaço de Hilbert gerado pela base ortonormal $|x\rangle$ onde $x \in \{0,1\}^n$. Note que, U_f aplica uma fase -1 exatamente no estado marcado.

4.2.2 O algoritmo de Grover

Iniciamos a computação por aplicar o operador de Hadamard (ver exemplo 2.2) em todos os qubits e calculamos

$$|s\rangle \;=\; H^{\otimes n}\,|0\rangle^{\otimes n} \;=\; \frac{1}{\sqrt{N}}\sum_{x\in\{0,1\}^n}|x\rangle\,. \tag{4.2}$$

Calculamos o estado ortogonal ao estado marcado

$$|x^\star_\perp\rangle \;=\; \frac{1}{\sqrt{N-1}}\sum_{x\neq x^\star}|x\rangle\,. \tag{4.3}$$

Então, o estado inicial $|s\rangle$ pertence ao subespaço V gerado por $\{|x^\star\rangle\,,|x^\star_\perp\rangle\}$ e se decompõe como

$$|s\rangle = \operatorname{sen}(\theta)\,|x^\star\rangle + \cos(\theta)\,|x^\star_\perp\rangle \implies |s\rangle = \begin{bmatrix}\operatorname{sen}\theta\\ \cos\theta\end{bmatrix}, \tag{4.4}$$

onde $\operatorname{sen}(\theta) = \dfrac{1}{\sqrt{N}}$ e $\cos(\theta) = \sqrt{\dfrac{N-1}{N}}$.

O subespaço V é U_f-invariante e a matriz da restrição de U_f na base $\{|x^\star\rangle\,,|x^\star_\perp\rangle\}$ é dada por

$$U_f = \begin{pmatrix}-1 & 0\\ 0 & 1\end{pmatrix}. \tag{4.5}$$

A computação acontece apenas no espaço V de dimensão 2.

Usando o produto externo definido na seção 1.3, definimos o operador difusor que age no espaço V como

$$D \;=\; 2\,|s\rangle\langle s| - I = \begin{pmatrix}2\operatorname{sen}^2\theta - 1 & 2\operatorname{sen}\theta\cos\theta\\ 2\operatorname{sen}\theta\cos\theta & 2\cos^2\theta - 1\end{pmatrix} = \begin{pmatrix}-\cos(2\theta) & \operatorname{sen}(2\theta)\\ \operatorname{sen}(2\theta) & \cos(2\theta)\end{pmatrix}, \tag{4.6}$$

onde $|s\rangle$ é o estado definido na equação (4.2). É fácil verificar que este operador pode ser implementado (a menos de uma fase global) por

$$D \;=\; H^{\otimes n}\left(2\,|0\rangle\langle 0| - I\right)H^{\otimes n}, \tag{4.7}$$

onde $(2\,|0\rangle\langle 0| - I)$ é uma reflexão que deixa $|0\rangle^{\otimes n}$ invariante e muda o sinal de todos os demais estados computacionais.

4.2.3 Iteração

A *iteração de Grover* é o operador $G = DU_f$ que age em V como

$$G = DU_f = \begin{pmatrix}-\cos(2\theta) & \operatorname{sen}(2\theta)\\ \operatorname{sen}(2\theta) & \cos(2\theta)\end{pmatrix}\begin{pmatrix}-1 & 0\\ 0 & 1\end{pmatrix} = \begin{pmatrix}\cos(2\theta) & \operatorname{sen}(2\theta)\\ -\operatorname{sen}(2\theta) & \cos(2\theta)\end{pmatrix}. \tag{4.8}$$

O operador de Grover atua em $V^\perp$ como a identidade, além disso, G atua como uma rotação no plano V gerado por $\{|x^\star\rangle, |x^\star_\perp\rangle\}$, aumentando a amplitude no estado marcado. De fato, partindo de $|s\rangle$, após r iterações temos

$$G^r |s\rangle = \begin{pmatrix} \cos(2r\theta) & \operatorname{sen}(2r\theta) \\ -\operatorname{sen}(2r\theta) & \cos(2r\theta) \end{pmatrix} \begin{pmatrix} \operatorname{sen}\theta \\ \cos\theta \end{pmatrix} \tag{4.9}$$

$$= \begin{pmatrix} \cos(2r\theta)\operatorname{sen}\theta + \operatorname{sen}(2r\theta)\cos\theta \\ -\operatorname{sen}(2r\theta)\operatorname{sen}\theta + \cos(2r\theta)\cos\theta \end{pmatrix} \tag{4.10}$$

$$= \begin{pmatrix} \operatorname{sen}((2r+1)\theta) \\ \cos((2r+1)\theta) \end{pmatrix} \tag{4.11}$$

$$= \operatorname{sen}((2r+1)\theta)\,|x^\star\rangle + \cos((2r+1)\theta)\,|x^\star_\perp\rangle \tag{4.12}$$

Logo, a probabilidade de medir o item marcado após r iterações é

$$P_{x^\star}(r) \;=\; \operatorname{sen}^2\Big((2r+1)\theta\Big). \tag{4.13}$$

4.2.4 Número de iterações e complexidade

Para maximizar $P_{x^\star}(r)$ escolhe-se $(2r+1)\theta \approx \dfrac{\pi}{2}$, isto é,

$$r \;\approx\; \frac{\pi}{4\theta} - \frac{1}{2}. \tag{4.14}$$

Como $\operatorname{sen}(\theta) = 1/\sqrt{N}$, para N grande temos $\theta \approx 1/\sqrt{N}$ e portanto

$$r \;=\; \left\lfloor \frac{\pi}{4}\sqrt{N} \right\rfloor. \tag{4.15}$$

Assim, o algoritmo encontra $x^\star$ com alta probabilidade usando $O(\sqrt{N})$ chamadas ao oráculo.

4.2.5 Busca por multiplas soluções

Descrevemos brevemente a variação do algoritmo para vários estados marcados. Maiores detalhes podem ser encontrados no livro [Nielsen and Chuang, 2010, seção 6.1]. Para o caso de M estados marcados, tomamos

$$|x^\star\rangle = \frac{1}{\sqrt{M}} \sum_{x:f(x)=1} |x\rangle \qquad |x^\star_\perp\rangle = \frac{1}{\sqrt{N-M}} \sum_{x:f(x)=0} |x\rangle\,, \tag{4.16}$$

assim,

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x\in\{0,1\}^n} |x\rangle \implies |s\rangle = \frac{M}{N}\,|x^\star\rangle + \frac{N-M}{N}\,|x^\star_\perp\rangle\,. \tag{4.17}$$

Tomando $\operatorname{sen}(\theta) = \sqrt{M/N}$, obtemos, com o mesmo raciocínio,

$$r \approx \left\lfloor \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rfloor, \tag{4.18}$$

ainda com rotação no subespaço de dimensão 2 gerado pelo vetor uniforme nos estados marcados e pelo vetor uniforme nos estados não-marcados.

4.2.6 Implementação em Qiskit

A implementação dada a seguir é uma adaptação da implementação fornecida pela IBM em IBM Quantum [2023a]. A figura 25 define a função oráculo que realiza a inversão de fase. A função assume que todos os estados marcados tem a mesma quantidade de qubits do primeiro estado da lista informada (linha 9). As linhas de 12 a 25 são responsáveis por pegar o estado marcado em formato de cadeia de caracteres, mudar a ordem dos caracteres para *little-endian* e, em seguida, aplicar o operador X em todos os qubits cujo índice da cadeia de caracteres revertida tem valor zero (que produz o efeito de inverter $|0\rangle$ para $|1\rangle$). A linha 21 aplica o operador Z controlado (MCZ) pelo número de qubits menos um (o qubit-alvo é alterado se todos os qubits de controle possuirem valor $|1\rangle$), sendo o qubit-alvo, o último qubit. Como o operador Z altera fase ($|1\rangle$ para $-|1\rangle$), a inversão inicial de todos os qubits com valor $|0\rangle$ para $|1\rangle$ e a aplicação de MCZ (via instrução `qc.compose(MCMTGate(ZGate(), ...)` permite a mudança de fase desejada, a aplicação do operador X novamente ao final permite que o estado final seja o mesmo informado, caso o estado do circuito não seja um estado marcado, ou o estado original com fase invertida.

```
1  import math
2
3  from qiskit import QuantumCircuit, transpile
4  from qiskit.circuit.library import grover_operator, MCMTGate, ZGate
5  from qiskit_aer import AerSimulator
6  from qiskit.visualization import plot_distribution
7
8  def grover_oracle(marked_states: list[str]) -> QuantumCircuit:
9      num_qubits = len(marked_states[0])
10
11     qc = QuantumCircuit(num_qubits)
12     for target in marked_states:
13         rev_target = target[::-1]
14         zero_inds = [
15             ind
16             for ind in range(num_qubits)
17             if rev_target.startswith("0", ind)
18         ]
19         if zero_inds:
20             qc.x(zero_inds)
21         qc.compose(MCMTGate(ZGate(), num_qubits - 1, 1), inplace=True)
22         if zero_inds:
23             qc.x(zero_inds)
24
25         qc.barrier()
26     return qc
```

Figura 25 – Função de criação do oráculo a partir dos estados marcados.

A figura 26 mostra o uso da função para o caso de dois estados marcados e quatro qubits. A figura 27 apresenta o circuito produzido pela função oráculo.

```
28 marked_states = ["0110", "1001"]
29 oracle = grover_oracle(marked_states)
30 oracle.draw(output="mpl", style="iqp")
```

Figura 26 – Criação do oráculo para os estados $|0110\rangle$ e $|1001\rangle$.

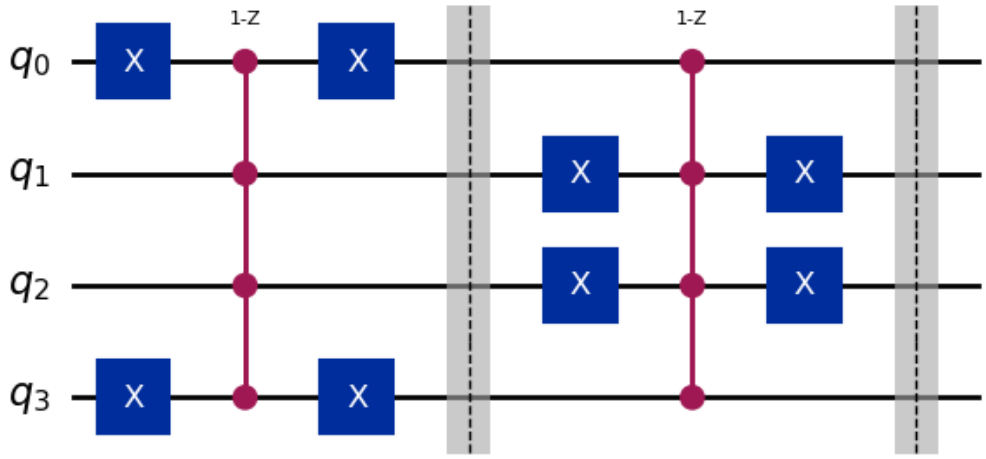

Figura 27 – Circuito quântico do oráculo para os estados marcados 0110 e 1001.

Em seguida, a figura 28 apresenta a utilização do operador de Grover, cuja implementação já existe no Qiskit IBM Quantum [2025f], de modo análogo a seção 4.2.3, o operador de Grover é construído a partir do operador de difusão (implementação interna) em conjunto com o oráculo.

```
32 grover_op = grover_operator(oracle)
33 grover_op.decompose().draw(output="mpl", style="iqp")
```

Figura 28 – Criação do operador de Grover.

Determinamos o número de iterações de acordo com a equação (4.14). Podemos nos questionar sobre o fato de, por um lado, precisarmos de iterações e, por outro, como mencionado na seção 3.5, os circuitos precisarem ser acíclicos. As iterações em circuitos quânticos no Qiskit são implementadas previamente à execução via concateção. Em nossa implementação, como temos quatro qubits precisamos de duas iterações, concatenamos então o circuito do operador de Grover duas vezes. Diferente de muitos algoritmos, onde o aumento no número de iterações produz um aumento de precisão, no algoritmo de Grover, o aumento não garante melhora. A figura 29 apresenta a cálculo de iterações e a construção do circuito completo. Após determinar o número ótimo de iterações geramos a

superposição (linha 42), concatenamos o operador de Grover (linha 44) e então realizamos a medição de todos os qubits (linha 45). A figura 30 apresenta o diagrama do circuito.

```
35 N = 2**grover_op.num_qubits
36 M = len(marked_states)
37 optimal_num_iterations = math.floor(math.pi / (4 *
       math.asin(math.sqrt(M/N))))
38
39 print("iterations:", optimal_num_iterations) # iterations: 2
40
41 qc = QuantumCircuit(grover_op.num_qubits)
42 qc.h(range(grover_op.num_qubits))
43
44 qc.compose(grover_op.power(optimal_num_iterations), inplace=True)
45 qc.measure_all()
46
47 qc.draw(output="mpl", style="iqp")
```

Figura 29 – Definição do número ótimo de iterações e criação do circuito final.

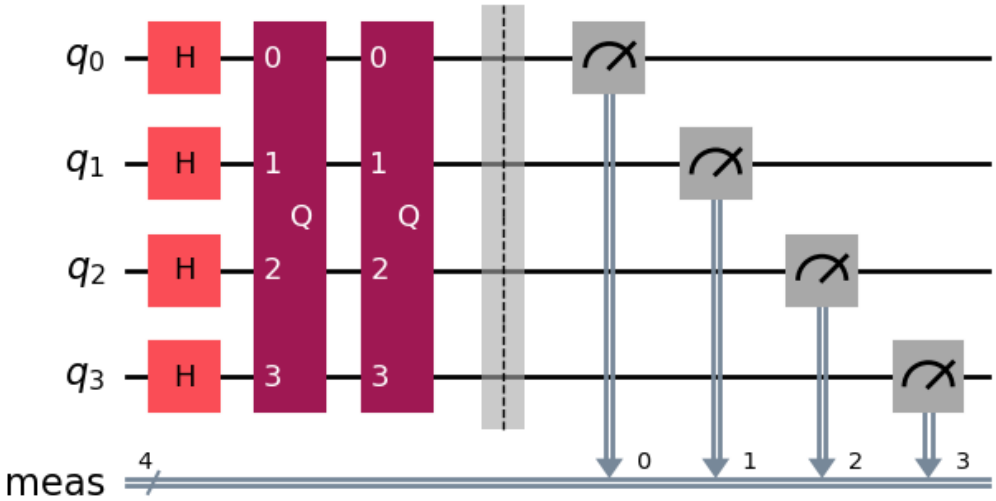

Figura 30 – Diagrama do circuito quântico para o algoritmo de Grover para quatro qubits.

Por fim, podemos realizar a simulação do circuito criado. A figura 31 apresenta o código utilizado para executar a simulação e a figura 32 apresenta o histograma gerado, mostrando que os estados marcados são medidos com maior probabilidade do que os demais.

```
sim_statevector = AerSimulator(method='statevector')
grover = transpile(qc, sim_statevector)

job_statevector = sim_statevector.run(grover, shots=10000)
dist = job_statevector.result().get_counts()

plot_distribution(dist)
```

Figura 31 – Execução do algoritmo de Grover em simulador.

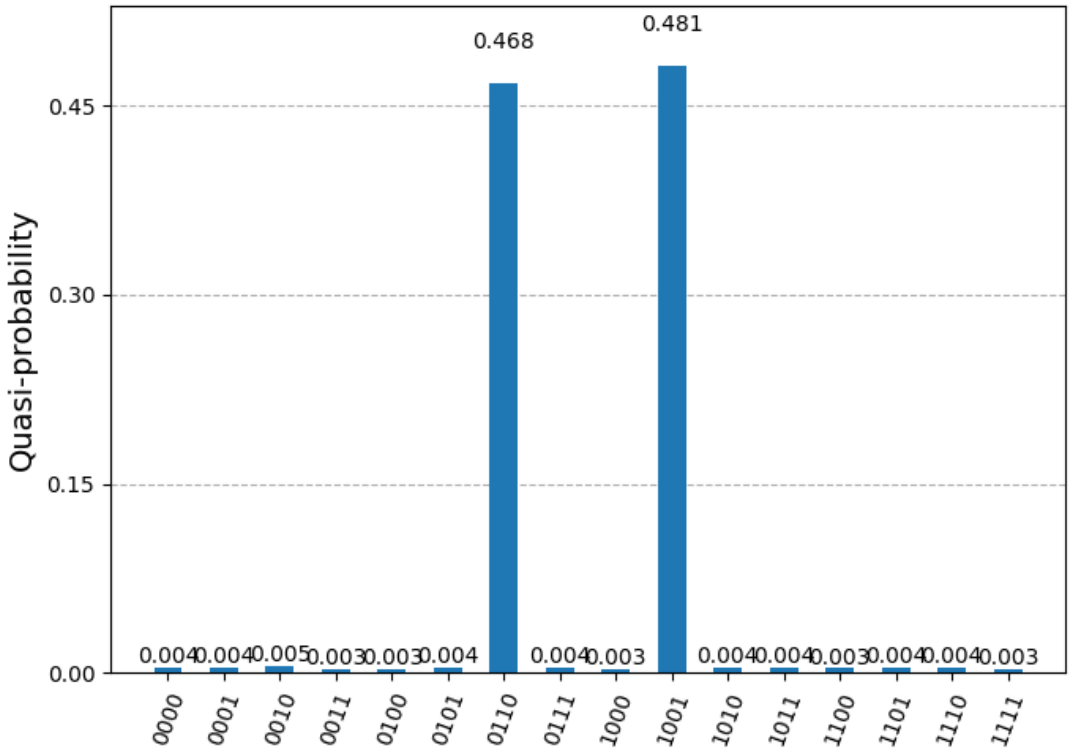

Figura 32 – Histograma de medição para o algoritmo de Grover.

4.3 QAOA (Algoritmo Quântico de Otimização Aproximada)

4.3.1 Considerações iniciais

O conteúdo desta seção é baseado na descrição do QAOA fornecida pelo artigo Farhi et al. [2014]. Pretendemos dar uma descrição de problema de otimzação combinatória e justificativa informal do funcionamento do QAOA, ao final propomos uma implementação em Qiskit.

4.3.2 Problemas de Otimização Combinatória

Um problema de otimização combinatória pode ser especificado por uma família de m funções $C_\alpha : X = \{0,1\}^n \to \{0,1\}$, onde o objetivo é achar um z

$$z = z_1 z_2 \cdots z_n \in \{0,1\}^n, \tag{4.19}$$

uma sequência binária, que representa uma possível solução do problema, tal que a função objetivo

$$C(z) = \sum_{\alpha=1}^{m} C_\alpha(z), \tag{4.20}$$

tem valor maximal (z é dita solução ótima do problema). Geralmente, supomos que cada C_α pode ser construído por $O(1)$ portas, o que confere à função objetivo uma estrutura local. No contexto de otimização aproximada, o objetivo não é necessariamente encontrar a solução ótima, mas sim uma sequência z para a qual o valor de $C(z)$ seja próximo do valor maximal. O QAOA é um algoritmo quântico desenvolvido para esse cenário de otimização aproximada, explorando a estrutura local da função objetivo para construir circuitos quânticos de profundidade controlada (ver seção 3.5).

Podemos mapear a equação (4.20) em um espaço de Hilbert $\mathcal{H}$ das combinações lineares de elementos de $X = \{0,1\}^n$ com produto interno, no qual X é base ortonormal. Temos $\dim \mathcal{H} = 2^n$ (equação (1.14)) com vetores $|z\rangle$ da base computacional X e vemos $\mathcal{H}$ como

$$\mathcal{H} = H_1 \otimes \cdots \otimes H_n, \tag{4.21}$$

onde cada H_i é um espaço de Hilbert de dimensão 2, ou seja, um qubit. Assim, C pode ser visto como um operador diagonal (e também hermitiano) na base computacional

$$C = \sum_{z \in X} C(z) \, |z\rangle \langle z| \tag{4.22}$$

e, portanto, os $|z\rangle$ são autovetores de C cujos autovalores associados são $C(z)$:

$$C \, |z\rangle = C(z) \, |z\rangle \quad \text{para todo } |z\rangle \in X. \tag{4.23}$$

4.3.3 Operadores de custo e de mistura

Seja $C : \mathcal{H} \to \mathcal{H}$ o operador hermitiano da seção 4.3.2. Chamamos de *operador de custo*, o operador unitário (dado que C é hermitiano)

$$U(C, \gamma) = \prod_{\alpha=1}^{m} e^{-i\gamma C_\alpha} = e^{-i\gamma C}, \tag{4.24}$$

onde $\gamma \in [0, 2\pi)$. Em geral, o produto de operadores não comuta, no entanto, como cada C_α é diagonal e na base computacional, temos que cada termo do produtório comuta. Este operador é unitário e pode ser implementado por um circuito quântico de profundidade máxima $O(m)$ (uma camada para cada termo do produtório).

Considere $\mathcal{H}$ como definido na equação (4.21) e seja $X_j = I \otimes I \otimes \ldots \otimes I \otimes X \otimes I \otimes \ldots \otimes I$, ou seja, X_j é a aplicação do operador X de Pauli no qubit j, mantendo todos os demais qubits inalterados. Definimos

$$B = \sum_{j=1}^{n} X_j \tag{4.25}$$

e chamamos de *operador de mistura*, o operador unitário (pois cada X_j é hermitiano que comutam entre si)

$$U(B,\beta) = \prod_{j=1}^{n} e^{-i\beta X_j} = e^{-i\beta B}, \tag{4.26}$$

onde $\beta \in [0,\pi)$. Além disso, podemos ver que $U(B,\beta) = \bigotimes_{j=1}^{n} R_x(2\beta)$, conforme equação (3.10).

4.3.4 Inicialização

Dado $C : \mathcal{H} \to \mathcal{H}$, como na seção 4.3.2, queremos encontrar um autovetor cujo autovalor é autovalor maximal de C. Iniciamos o algoritmo no estado $|\psi\rangle = \bigotimes_{i=1}^{n} |0\rangle$, e aplicamos o operador de Hadamard (H) para obter a superposição

$$|s\rangle = H^{\otimes n}\left(\bigotimes_{i=1}^{n} |0\rangle\right) \qquad H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{4.27}$$

$$= |+\rangle_1 \otimes |+\rangle_2 \otimes \ldots \otimes |+\rangle_n \qquad |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{4.28}$$

$$= \frac{1}{\sqrt{2^n}} \sum_{z \in X} |z\rangle\,. \tag{4.29}$$

Os valores de γ e β nas equação (4.24) e (4.26) podem ser vistos como ângulos. Assim, dado um inteiro $p \geq 1$ podemos, a partir de $2p$ ângulos $\gamma_1, \gamma_2, \ldots, \gamma_p$ e $\beta_1, \beta_2, \ldots, \beta_p$, definir o estado quântico

$$|\gamma,\beta\rangle = U(B,\beta_p)U(C,\gamma_p)\ldots U(B,\beta_1)U(C,\gamma_1)\,|s\rangle = \prod_{l=p}^{1} e^{-i\beta_l B} e^{-i\gamma_l C}\,|s\rangle\,. \tag{4.30}$$

Note que a inversão de índice no produtório é intencional, indicando a ordem em que os operadores são aplicados, iniciando em $U(B,\beta_1)U(C,\gamma_1)$ até $U(B,\beta_p)U(C,\gamma_p)$. Para implementar $U(C,\gamma_i)$ precisamos de profundidade máxima $O(m)$ e para implementar $U(B,\beta_i)$ precisamos de $O(1)$ portas, portanto, temos que $|\gamma,\beta\rangle$ é produzido por um circuito com profundidade máxima $O(mp)$ (cada fator do produtório necessita de $O(m)$ portas).

4.3.5 Iterações

De acordo com a equação (2.32), a expectativa do operador C no estado $|\gamma,\beta\rangle$ é dada por

$$F_p(\gamma,\beta) = \langle\gamma,\beta|C|\gamma,\beta\rangle\,, \tag{4.31}$$

denotamos o máximo de F_p para os $2p$ ângulos por

$$M_p = \max_{\gamma,\beta} F_p(\gamma, \beta), \tag{4.32}$$

onde $|\gamma, \beta\rangle$ é unitário. Observe que

$$M_p \geq M_{p-1}, \tag{4.33}$$

pois, para duas sequências de $p-1$ ângulos $\gamma = \gamma_1, \ldots \gamma_{p-1}$, $\beta = \beta_1, \ldots, \beta_{p-1}$, denote por γ' e por β' as sequências de comprimento p que obtemos por adicionar ângulos $\gamma_p = \beta_p = 0$ na p-ésima posição. Temos que

$$F_p(\gamma', \beta') = F_{p-1}(\gamma, \beta), \tag{4.34}$$

deste modo, $M_p = M_{p-1}$. Por outro lado, note que o valor máximo absoluto de M_p é dado pelo lema 4.1.

Lema 4.1 (Princípio variacional para o operador custo)**.** *Seja C um operador hermitiano atuando em um espaço de Hilbert $\mathcal{H}$ de dimensão finita, com decomposição espectral*

$$C = \sum_{i=1}^{m} c_i |e_i\rangle \langle e_i|, \qquad c_1 \geq c_2 \geq \cdots \geq c_m \geq 0.$$

Então, para todo estado normalizado $|\psi\rangle \in \mathcal{H}$, vale

$$\langle\psi| C |\psi\rangle \leq c_1.$$

Além disso,

$$\max_{\||\psi\|=1} \langle\psi| C |\psi\rangle = c_1,$$

e o máximo é atingido se, e somente se, $|\psi\rangle$ pertence ao subespaço próprio associado ao maior autovalor c_1.

Demonstração. Seja $|\psi\rangle \in \mathcal{H}$ um estado normalizado, que pode ser escrito como

$$|\psi\rangle = \sum_{i=1}^{m} \psi_i |e_i\rangle, \qquad \sum_{i=1}^{m} |\psi_i|^2 = 1.$$

Então,

$$\langle\psi| C |\psi\rangle = \sum_{i=1}^{m} |\psi_i|^2 c_i.$$

Como $c_i \leq c_1$ para todo $i = 1, 2, \ldots, m$, segue que

$$\langle\psi| C |\psi\rangle \leq \sum_{i=1}^{m} |\psi_i|^2 c_1 = c_1.$$

Tomando $|\psi\rangle = |e_1\rangle$, obtemos

$$\langle\psi| C |\psi\rangle = c_1,$$

o que conclui a prova. ■

No QAOA, os estados permitidos são restritos aos estados unitários na forma $|\gamma, \beta\rangle$ conforme equação (4.30), assim, não necessariamente temos

$$|\gamma, \beta\rangle = |e_1\rangle\,, \quad \text{para algum } (\gamma, \beta). \tag{4.35}$$

No entanto, a medida que p aumenta, a família de estados $|\gamma, \beta\rangle$, se aproxima arbitrariamente do autovetor de maior autovalor de C (para maiores detalhes consulte [Farhi et al., 2014, seção VI]). Matematicamente,

$$\lim_{p\to\infty} M_p = \max_{z\in X} C(z). \tag{4.36}$$

Vamos justificar, informalmente, a equação (4.36). Inicialmente, vejamos um resultado preliminar [Nielsen and Chuang, 2010, seção 4.7.2].

Teorema 4.2 (Fórmula de Lie-Trotter)**.** *Sejam C e B dois operadores hermitianos, então para todo $\Delta t \in \mathbb{R}$,*

$$e^{\,i(B+C)\Delta t} = e^{\,iB\Delta t}\, e^{\,iC\Delta t} \;+\; O(\Delta t^2). \tag{4.37}$$

O termo $O(\Delta t^2)$ deve ser entendido no sentido da norma de operadores [Horn and Johnson, 2012, seção 5.6]. Mais precisamente, existe um real $c > 0$, dependente apenas dos operadores B e C, tal que

$$||e^{i(B+C)\Delta t} - e^{iB\Delta t}\, e^{iC\Delta t}|| \leq c\,\Delta t^2.$$

O QAOA pode ser visto como uma versão discreta de um outro algoritmo chamado QAA (algoritmo de evolução adiabática quântica), desenvolvido para encontrar soluções ótimas de problemas de otimização Farhi et al. [2002]. A descrição dada a seguir foi feita com base nas referências Olivares [2021], Todadri [2017] e Kottmann [2024].

Trabalhando em unidades tais que $\hbar = 1$, a evolução temporal de um sistema quântico governado por um hamiltoniano dependente do tempo $H(t)$, onde $t \in [0, T]$, é descrita pelo operador unitário

$$U(t_1, t_2) = \mathcal{T} \exp\!\left(-i \int_{t_1}^{t_2} H(t)\, dt\right), \tag{4.38}$$

onde $\mathcal{T}$ denota o operador de ordenação temporal e exp a exponencial de operador. Em particular, definimos

$$U(T) = U(0, T), \tag{4.39}$$

de modo que o estado do sistema no instante final é dado por

$$|\psi(T)\rangle = U(T)\,|\psi(0)\rangle\,. \tag{4.40}$$

No contexto do algoritmo de evolução adiabática quântica, consideramos o hamiltoniano de interpolação linear

$$H(t) = \left(1 - \frac{t}{T}\right) B + \frac{t}{T} C, \tag{4.41}$$

definido no intervalo $t \in [0, T]$, com $H(0) = B$ e $H(T) = C$. Assumindo que o estado inicial $|\psi(0)\rangle = |s\rangle$ é um autovetor associado ao maior autovalor de B, o teorema adiabático garante que, para T suficientemente grande e sob condições adequadas de lacuna espectral, o estado final $|\psi(T)\rangle$ aproxima-se de um autovetor associado ao maior autovalor de C.

Para estabelecer uma conexão explícita com o QAOA, dividimos o intervalo temporal $[0, T]$ em p subintervalos de comprimento $\Delta t = T/p$, e definimos os pontos $t_l = l\Delta t$, com $l = 0, \ldots, p$. O operador de evolução total pode então ser escrito, usando a propriedade de composição, como

$$U(T) = \prod_{l=p-1}^{0} U(t_l, t_{l+1}), \tag{4.42}$$

onde a ordem do produto é essencial, pois, em geral, o produto de operadores não comuta. Em cada subintervalo, aproximamos o hamiltoniano por seu valor em t_l, obtendo

$$U(t_l, t_{l+1}) \approx e^{-i\Delta t\left((1-s_l)B + s_l C\right)}, \qquad s_l = \frac{l}{p}. \tag{4.43}$$

Aplicando a fórmula de Lie–Trotter, cada fator pode ser ainda aproximado por

$$e^{-i\Delta t\left((1-s_l)B + s_l C\right)} \;\approx\; e^{-i\beta_l B} e^{-i\gamma_l C}, \tag{4.44}$$

onde

$$\beta_l = \Delta t(1 - s_l), \qquad \gamma_l = \Delta t s_l. \tag{4.45}$$

Dessa forma, a evolução total pode ser escrita como

$$U(T)\,|s\rangle \;\approx\; \left(\prod_{l=p-1}^{0} e^{-i\beta_l B} e^{-i\gamma_l C}\right)|s\rangle \;=\; |\gamma, \beta\rangle\,, \tag{4.46}$$

o que evidencia que o QAOA pode ser interpretado como uma discretização da evolução adiabática contínua, com os parâmetros $\{\gamma_l, \beta_l\}$ codificando os passos temporais da interpolação. Deste modo, sempre é possível encontrar uma quantidade p de passos e ângulos γ e β pequenos que tornam $F_p(\gamma, \beta)$ tão próximo de M_p quanto se queira, em conjunto com a monotonicidade de M_p, validamos a equação (4.36). Para o argumento original consulte [Farhi et al., 2014, seção VI].

4.4 Implementação do QAOA em Qiskit

4.4.1 Definição do problema do corte máximo

Vamos implementar o QAOA para resolver o problema do Corte Máximo (MaxCut) em um grafo. É possível encontrar um tutorial específico para a implementação em Qiskit em IBM Quantum [2023b], no entanto, a implementação a seguir difere um pouco do tutorial.

Seja $G = (V, E)$ um grafo não direcionado, com um conjunto V de vértices e E de arestas. Um corte de G é uma partição do conjunto de vértices em dois subconjuntos disjuntos. Por exemplo, podemos escolher $A \subseteq V$ e tomando $\bar{A} = V \setminus A$, um corte de G pode ser visto como a escolha de um subconjunto A de V. O valor do corte é o número de arestas que tem uma extremidade em A e outra fora de A,

$$\text{Cut}(A) = |\{\{i, j\} \in E : i \in A, j \in \bar{A}\}|. \tag{4.47}$$

O problema do corte máximo consiste em encontrar conjunto $A \subseteq V$ cujo valor do corte é máximo, ou seja,

$$\text{MaxCut}(G) = \max_{A \subseteq V} \text{Cut}(A). \tag{4.48}$$

4.4.2 Modelagem

Seja $A \subseteq V$ um corte de um grafo como visto na seção 4.4.1. Associamos a cada vértice $i \in V$ uma variável binária

$$z_i \in \{0, 1\}, \qquad z_i = 1 \iff i \in A. \tag{4.49}$$

Assim, uma aresta $\{i, j\}$ contribui para o corte se e somente se $z_i \neq z_j$. Sendo $\oplus$ a operação de adição modulo 2, para $1 \leq i, j \leq n$, definimos a função $C_{ij} : X \to \{0, 1\}$ como

$$C_{ij}(z) = z_i \oplus z_j. \tag{4.50}$$

Note que se i, j é uma aresta do grafo, então $C_{ij}(z) = 1$ se, e somente se, a aresta contribui à quantidade $\text{Cut}(A)$ e temos que $C_{ij}(z) = 0$ caso contrário. Definimos a função objetivo do MaxCut como

$$C(z) = \sum_{\{i,j\} \in E} z_i \oplus z_j = \sum_{\{i,j\} \in E} C_{ij}(z). \tag{4.51}$$

Se estendermos C_{ij} para o espaço de Hilbert $\mathcal{H}$ de dimensão 2^n (como na seção 4.3.2), então, sendo Z o operador de Pauli, C_{ij} pode ser escrito como

$$C = \sum_{\{i,j\} \in E} \frac{1}{2}(I - Z_i Z_j), \quad Z_i Z_j = I \otimes \ldots \otimes Z \otimes \ldots \otimes Z \otimes \ldots \otimes I, \tag{4.52}$$

que é um operador diagonal na base computacional. A figura 33 apresenta código para criação dos operadores de custo, mistura e a construção do circuito do QAOA em Qiskit.

```python
from qiskit import QuantumCircuit, transpile
import numpy as np
from qiskit_aer import AerSimulator
from qiskit.visualization import plot_histogram
from scipy.optimize import differential_evolution

def cost_layer(qc: QuantumCircuit, gamma: float, edges: list):
    for i, j in edges:
        qc.cx(i, j)
        qc.rz(2*gamma, j)
        qc.cx(i, j)

def mixer_layer(qc: QuantumCircuit, beta: float):
    for q in range(qc.num_qubits):
        qc.rx(2*beta, q)

def qaoa_block(n_qubits: int, edges: list, gamas: list, betas: list):
    p = len(gamas)
    qc = QuantumCircuit(n_qubits)

    qc.h(range(n_qubits))
    qc.barrier()

    for l in range(p):
        gamma_l = gamas[l]
        beta_l = betas[l]
        cost_layer(qc, gamma_l, edges)
        qc.barrier()
        mixer_layer(qc, beta_l)
        qc.barrier()

    qc.measure_all()

    return qc
```

Figura 33 – Criação dos operadores de custo, mistura e circuito do QAOA.

Para o grafo da figura 35, a figura 34 apresenta o uso das funções construídas. O diagrama do circuito pode ser visto na figura 36, note que como estamos tomando $p = 1$ temos apenas um par $(U(C, \gamma), U(B, \beta))$ adicionado no circuito.

```python
n = 3
edges = [(0,1), (1,2)]
gammas = [np.pi/5]
betas = [np.pi/6]
p = len(gammas)

qc = qaoa_block(n, edges, gammas, betas)
qc.draw(output="mpl", style="iqp")
```

Figura 34 – Exemplo de criação de circuito QAOA.

O código para execução do algoritmo pode ser vista na figura 37. A definição dos valores de $\gamma = \pi/5$ e $\beta = \pi/6$ na figura 34 não pode ser arbitrária. A execução do algoritmo

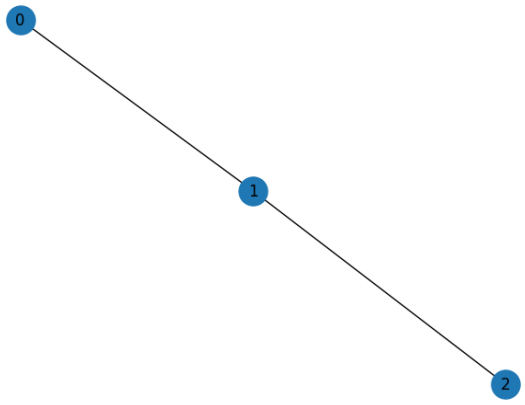

Figura 35 – Grafo para uso no QAOA.

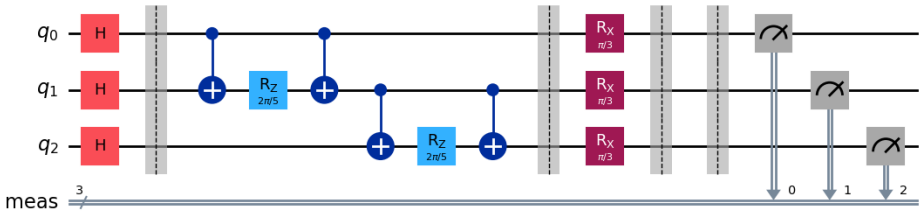

Figura 36 – Circuito do QAOA.

com tais valores, produz o resultado da figura 38, que indentifica, incorretamente, que o melhor corte seria $A = V$ ou $A = \emptyset$.

```
45 sim = AerSimulator()
46 tqc = transpile(qc, sim)
47 result = sim.run(tqc, shots=1000).result()
48 counts = result.get_counts()
49
50 plot_histogram(counts, title='Resultados de Medida')
```

Figura 37 – Execução do QAOA sem otimização de parâmetros.

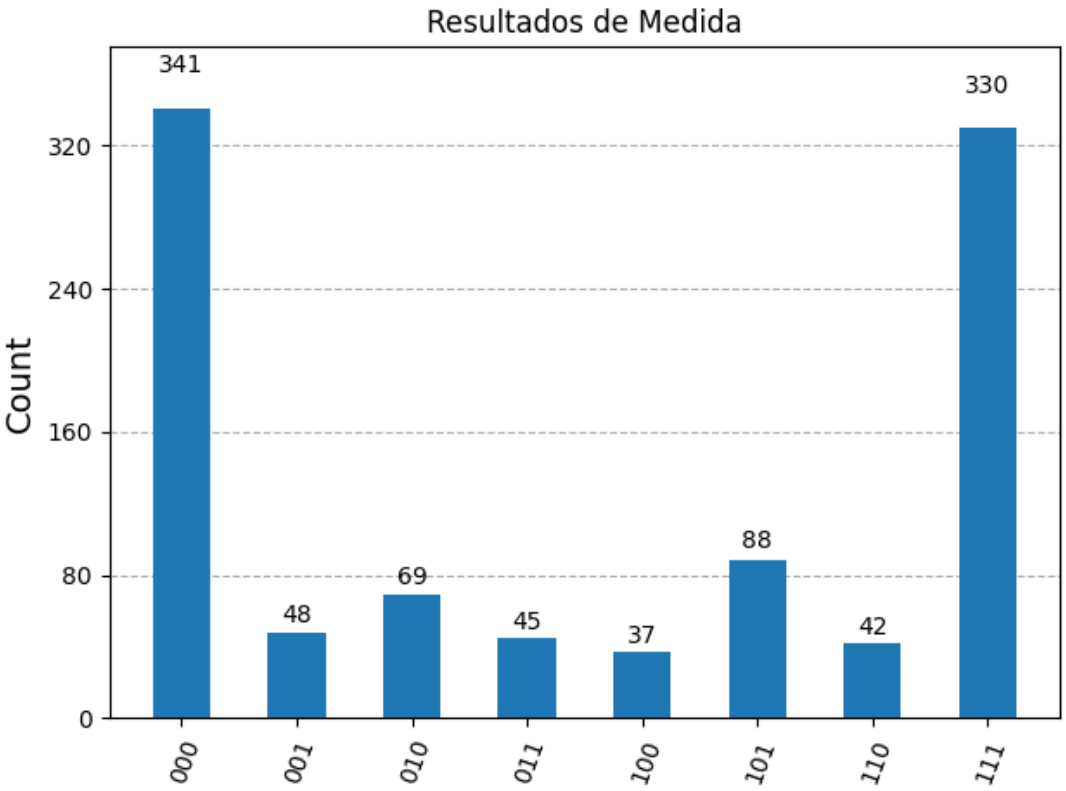

Figura 38 – Histograma de medida sem otimização de parâmetros.

Segundo Farhi et al. [2014] a principal dificuldade prática do QAOA está na escolha adequada dos $2p$ parâmetros $\{\gamma_1, \ldots, \gamma_p, \beta_1, \ldots, \beta_p\}$ que maximizam $F_p(\gamma, \beta)$. Para p fixo e independente do tamanho da instância, uma primeira estratégia consiste em realizar uma busca clássica sobre uma grade fina no conjunto compacto $[0, 2\pi]^p \times [0, \pi]^p$, isso permite utilizar algoritmos clássicos de otimização contínua como o gradiente descendente. Além disso, quando o grau do grafo é limitado, F_p pode ser decomposta como uma soma ponderada de contribuições locais associadas a subgrafos de tamanho finito, permitindo que os ângulos ótimos sejam determinados por um pré-processamento clássico A figura 39, apresenta o código para otimização de parâmetros e a figura 40 mostra a execução e os valores obtidos. A função `calc_cut` retorna o valor negativo do corte, isso é necessário, pois a função de otimização `differential_evolution` do pacote `scipy` realiza otimização minimizando o valor da função objetivo.

```python
def cut_value(z: list, edges: list):
    cut_value = 0
    for (i, j) in edges:
        if z[i] != z[j]:
            cut_value += 1

    return cut_value

def expected_cut(counts, edges: list):
    shots = sum(counts.values())

    exp_cut = 0
    for z, z_count in counts.items():
        exp_cut += cut_value(z, edges) * z_count/shots

    return exp_cut

def calc_cut(n_qubits: int, edges: list, gammas: list, betas: list):
    qc = qaoa_block(n_qubits, edges, gammas, betas)
    tqc = transpile(qc, sim)
    counts = sim.run(tqc, shots=1000).result().get_counts()

    return -expected_cut(counts, edges)

def qaoa_bounds(p: int):
    bounds = []
    for _ in range(p):
        bounds.append((0, 2*np.pi))
    for _ in range(p):
        bounds.append((0, np.pi))
    return bounds
```

Figura 39 – Otimização de parâmetros para execução do QAOA.

```python
bounds = qaoa_bounds(p)

result = differential_evolution(
    func=lambda params: calc_cut(
        n_qubits=qc.num_qubits,
        edges=edges,
        gammas=params[:p].tolist(),
        betas=params[p:].tolist()
    ),
    bounds=bounds,
    maxiter=50,
    popsize=10,
    tol=1e-2,
    polish=True
)

best_params = {"val": -result.fun, "params": result.x}
print(f"val: {best_params["val"]}, params:{best_params["params"]}")
# val: 1.672, params:[0.57105229 2.814569  ]
```

Figura 40 – Calculo dos valores otimizados.

4.4.3 Execução final

Com os parâmetros otimizados podemos executar o algoritmo. As figuras 41 e 42, apresentam, respectivamente, o código de execução e o histograma gerado. Veja que os estados $|010\rangle$ e $|101\rangle$ são apresentados corretamente como solução para o corte máximo.

```
104 params = best_params["params"]
105 qc = qaoa_block(n, edges, params[0:p], params[p:])
106
107 sim = AerSimulator()
108 tqc = transpile(qc, sim)
109 result = sim.run(tqc, shots=1000).result()
110 counts = result.get_counts()
111
112 plot_histogram(counts, title='Resultados de Medida')
```

Figura 41 – Execução do QAOA com parâmetros otimizados.

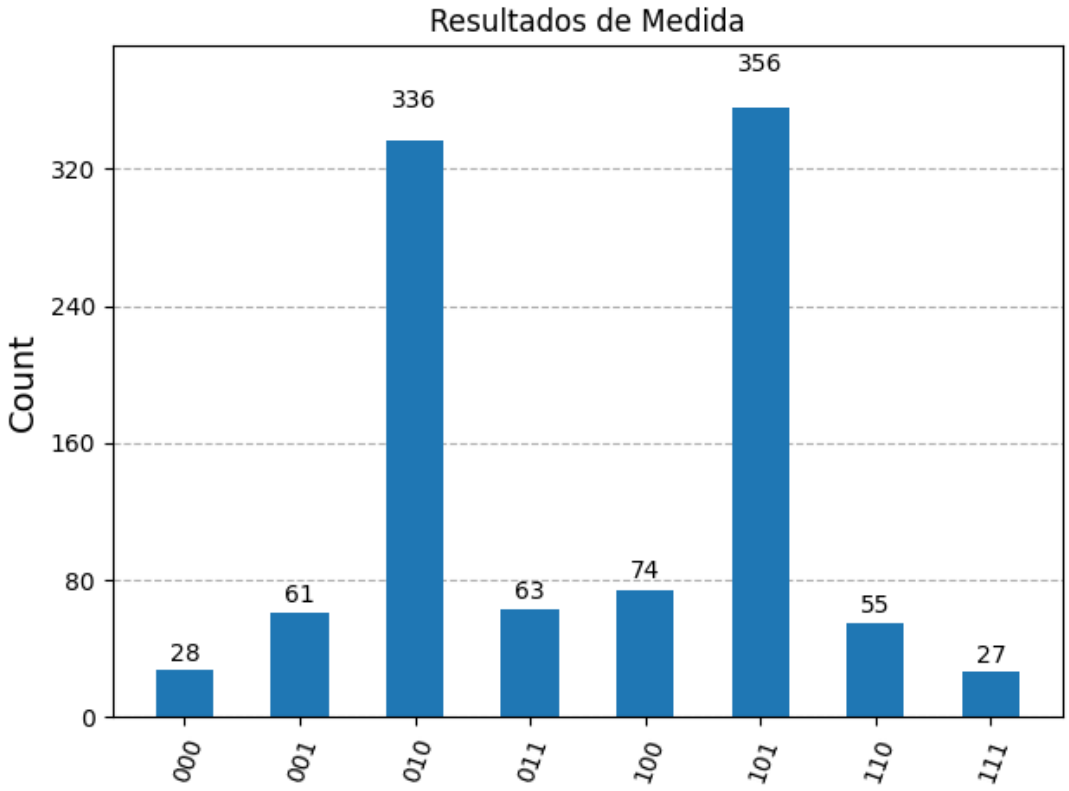

Figura 42 – Histograma de medida com o resultado correto para o corte máximo.

5 Conclusão

Neste trabalho foi proposto um estudo dos conceitos utilizados na computação quântica a partir da base matemática fornecida no curso de Especialização em Matemática ofertado pela UFMG. A computação quântica possui conexões bem amplas com a física, matemática e computação, sendo uma área desafiadora (e ao mesmo tempo intrigante) devido a maior profundidade dos conceitos utilizados e de certa forma, devido a "estranheza"criada pelas propriedades quânticas como superposição, emaranhamento entre outras.

Neste trabalho, primeiramente abordamos a computação quântica adequando o conhecimento matemático de espaços vetoriais, operadores lineares, produtos tensoriais às notações e conceitos da computação quântica, para esta parte as disciplinas do curso de Especialização em Matemática da UFMG foram essenciais, fornecendo uma base bastante sólida para o tema. O livro Quantum Computation and Quantum Information de Nielsen & Chuang Nielsen and Chuang [2010] foi fundamental durante todo o estudo sendo um grande guia para os tópicos e para entendimento das notações utilizadas.

Em seguida, foi feita a apresentação dos postulados da mecânica quântica, os quais em conjunto com os conceitos matemáticos anteriores são fundamentais para o entendimento e manipulação de sistemas quânticos ou mais diretamente circuitos quânticos. Após a revisão dos postulados comparamos os conhecimentos sobre circuitos clássicos com o que a literatura e as empresas tem feito para implementar versões quânticas de circuitos, vimos que toda porta lógica quântica precisa ser reversivel e que empresas como IBM possuem bibliotecas (tal como o Qiskit) para construção, visualização, otimização, simulação e execução (em processadores quânticos reais) de circuitos quânticos. Tais bibliotecas aproveitam de muitas ferramentas clássicas e linguagens de programação existentes (tal como Python) para construir circuitos e simular ou se conectar com sistemas que executarão de fato o circuito em um processador quântico.

Por fim, fizemos um estudo de dois algoritmos quânticos. O objetivo de estudá-los foi o de aplicar o conhecimento na construção de alguns algoritmos bastante conhecidos, tentando, na medida do possível, justificar matematicamente seu comportamento. O QAOA em especial, apesar de ter uma implementação simples em Qiskit, parte de conceitos bastante avançados a respeito de simulação de sistemas quânticos em tempo contínuo e adequação para execução em número fixo de passos. Para este último algoritmo foi dada uma justificativa mais informal de seu comportamento.

Referências

Scott Aaronson. Limits on efficient computation in the physical world. *arXiv preprint arXiv:quant-ph/0605218*, 2006. URL https://arxiv.org/abs/quant-ph/0605218. Citado na página 40.

S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. ISBN 9781139477369. URL https://books.google.com.br/books?id=nGvI7cOuOOQC. Citado na página 17.

Anindita Banerjee and Anirban Pathak. New designs of reversible sequential devices. *arXiv preprint arXiv:0908.1620*, 2009. URL https://arxiv.org/abs/0908.1620. Citado na página 40.

Alonzo Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2):345–363, 1936. doi: 10.2307/2371045. Citado na página 15.

F. U. Coelho and Mary Lilian Lourenço. *Curso de Álgebra Linear, Um Vol. 34*. EDUSP, 2001. ISBN 9788531405945. URL https://books.google.com.br/books?id=aiLoWF-9AaoC. Citado na página 17.

Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151–158, New York, NY, 1971. ACM. Citado na página 16.

Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. Quantum adiabatic evolution algorithms versus simulated annealing. jan 2002. doi: 10.48550/arXiv.quant-ph/0201031. URL https://arxiv.org/abs/quant-ph/0201031. 16 pp., 10 EPS figures, Report no. MIT-CTP-3228. Citado na página 63.

Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014. URL https://arxiv.org/abs/1411.4028. Citado 5 vezes nas páginas 53, 59, 63, 64 e 68.

Jan Faye. Copenhagen interpretation of quantum mechanics. Stanford Encyclopedia of Philosophy, 2022. URL https://plato.stanford.edu/entries/qm-copenhagen/. Citado na página 15.

Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 212–219, New York, NY, USA, 1996. Association for Computing Machinery. ISBN

0897917855. doi: 10.1145/237814.237866. URL https://doi.org/10.1145/237814.237866. Citado na página 53.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, UK, 2 edition, 2012. Ver Seção 5.6 "Matrix Norms", em especial a Subseção 5.6.2 "Operator Norms". Citado na página 63.

IBM Quantum. Grover's algorithm, 2023a. URL https://quantum.cloud.ibm.com/docs/en/tutorials/grovers-algorithm. Accessed: 2025-03-18. Citado na página 56.

IBM Quantum. Quantum approximate optimization algorithm, 2023b. URL https://quantum.cloud.ibm.com/docs/en/tutorials/quantum-approximate-optimization-algorithm. Citado na página 64.

IBM Quantum. Pricing | ibm quantum computing, 2025a. URL https://www.ibm.com/quantum/pricing. Acesso em: 22 jun. 2025. Citado na página 48.

IBM Quantum. Circuit library. https://quantum.cloud.ibm.com/docs/en/guides/circuit-library, 2025b. Accessed: 2026-01-07. Citado na página 47.

IBM Quantum. Visão geral dos guias — ibm quantum documentation, 2025c. URL https://quantum.cloud.ibm.com/docs/pt/guides. Acesso em: 16 Dezembro 2025. Citado na página 33.

IBM Quantum. Overview of operator classes. https://quantum.cloud.ibm.com/docs/en/guides/operators-overview, 2025d. Accessed: 2026-01-07. Citado na página 47.

IBM Quantum. Introdução ao qiskit, 2025e. URL https://quantum.cloud.ibm.com/docs/pt/guides/tools-intro. Acesso em: 16 Dezembro 2025. Citado na página 46.

IBM Quantum. *GroverOperator — Qiskit API Documentation*. IBM, 2025f. URL https://quantum.cloud.ibm.com/docs/en/api/qiskit/qiskit.circuit.library.GroverOperator. Acesso em: 08 jan. 2026. Citado na página 57.

IBM Quantum Platform. Install qiskit, 2025. URL https://quantum.cloud.ibm.com/docs/en/guides/install-qiskit. Acesso em: 22 jun. 2025. Citado na página 48.

Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Wei, Ewout Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme, and Abhinav Kandala. Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618: 500–505, 06 2023. doi: 10.1038/s41586-023-06096-3. Citado na página 48.

Korbinian Kottmann. What is the time-ordered exponential and why you should stop using it, sep 2024. URL https://pennylane.ai/blog/2024/09/time_ordered_exponential. PennyLane Blog post. Citado na página 63.

Leonid A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9(3):265–266, 1973. Citado na página 16.

M. Morris Mano and Michael D. Ciletti. *Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog*, chapter 3, pages 90–97. Pearson, Boston, 5 edition, 2013. Citado na página 37.

M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. ISBN 9781139495486. URL https://books.google.com.br/books?id=-s4DEy7o-a0C. Citado 8 vezes nas páginas 17, 25, 33, 38, 40, 55, 63 e 71.

Stefano Olivares. *Lecture Notes on Quantum Computing*. University of Milan, 2021. URL https://sites.unimi.it/olivares/wp-content/uploads/2021/08/lectures_qc_olivares_v5.0.pdf. Lecture notes (PDF). Versão 5.0, 9 de agosto de 2021. Citado na página 63.

Qiskit Community. Qiskit: An open-source framework for quantum computing, March 2017. URL https://github.com/Qiskit/qiskit. Disponível em: https://www.ibm.com/quantum/qiskit. Citado na página 46.

S. Roman. *Advanced Linear Algebra*. Graduate Texts in Mathematics. Springer New York, 2007. ISBN 9780387728315. URL https://books.google.com.br/books?id=bSyQr-wUys8C. Citado na página 23.

Senthil Todadri. Lecture 8: Quantum theory i. 2017. URL https://ocw.mit.edu/courses/8-321-quantum-theory-i-fall-2017/579cb61c4eaa494b9adb90830d921099_MIT8_321F17_lec8.pdf. MIT OpenCourseWare. Citado na página 63.

Alan M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2):230–265, 1936. doi: 10.1112/plms/s2-42.1.230. Citado na página 15.