

Laboratório Nacional de Computação Científica
Programa de Pós Graduação em Modelagem Computacional

**Um novo simulador de alta performance de caminhadas
quânticas**

Por
Aaron Bruno Leão

PETRÓPOLIS, RJ - BRASIL

NOVEMBRO DE 2015

UM NOVO SIMULADOR DE ALTA PERFORMANCE DE
CAMINHADAS QUÂNTICAS

Aaron Bruno Leão

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO LABORATÓRIO
NACIONAL DE COMPUTAÇÃO CIENTÍFICA COMO PARTE DOS REQUI-
SITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM
CIÊNCIAS EM MODELAGEM COMPUTACIONAL

Aprovada por:

Prof. Renato Portugal, D.Sc
(Presidente)

Prof. Franklin de Lima Marquezino, D.Sc

Prof. Gilson Antônio Giraldi, D.Sc.

PETRÓPOLIS, RJ - BRASIL
NOVEMBRO DE 2015

Leão, Aaron Bruno

XXXX um novo simulador de alta performance de caminhadas quânticas / Aaron Bruno Leão. Petrópolis, RJ. : Laboratório Nacional de Computação Científica, 2015.

xx, yy p. : il.; 29 cm

Orientador: Renato Portugal

Dissertação (M.Sc.) – Laboratório Nacional de Computação Científica, 2015.

1. Caminhadas quânticas. 2. Hiperwalk. 3. Simulação. 4. PAD. I. Portugal, Renato. II. LNCC/MCT. III. Título.

CDD XXX.XXX

Keep walking

A minha mãe

Agradecimentos

Aos meus familiares, a minha mãe Angela Maria Aires Teixeira e a minha irmã Jordana Aires Leão, sem vocês não haveria este trabalho.

Ao meu orientador Renato Portugal, pela paciência imensurável, compreensão, pelos empurrões motivacionais; por todos os ensinamentos e conselhos; por saber cobrar e relaxar; e o mais importante, por não ter desistido de mim.

Agradeço, também, aos pesquisadores da UDELAR, Alejandro Romanelli e Raul Donangelo, pelo excelente e frutífero tempo de muito conhecimento que passei no Uruguai, expandindo a minha visão da pesquisa.

Ao colega Pedro Carlos da Silva Lara pelo desenvolvimento da linguagem Neblina; por iluminar o caminho em momentos de dúvidas, sem você não teria o “Hiper” em “Hipewalk”.

Aos meus colegas de sala, Gastão Miranda, Leandro Tavares e aos meus colegas de grupo de pesquisa, Bruno Chagas, Tharso Fernandes e Daniel Gaspar; pessoal que sempre me ajudou a entender mais e mais do mundo onde estava aventurando.

A todos colegas da pós-graduação do LNCC com quem tive o prazer de entabular discussões que sempre ajudou ao longo desse tempo de mestrado.

Ao CNPq pelo apoio financeiro. À CAPES pelo programa CAPES-UDELAR.

Por último, mas não menos importante, agradeço a Sicilia Ferreira Ponce (quase de León) Pasini Judice pela sempre-ajuda que é ter um relacionamento em

um momento conturbado.

Resumo da Dissertação apresentada ao LNCC/MCT como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UM NOVO SIMULADOR DE ALTA PERFORMANCE DE CAMINHADAS QUÂNTICAS

Aaron Bruno Leão

Novembro , 2015

Orientador: Renato Portugal, D.Sc

O desenvolvimento de algoritmos quânticos não é uma tarefa trivial. Elementos como emaranhamento e paralelismo quântico, intrínsecos à computação quântica, dificultam esta tarefa. As caminhadas quânticas são ferramentas cruciais para o desenvolvimento de algoritmos, principalmente algoritmos de busca. Existem na literatura vários tipos de caminhadas: com lançamento de moeda, de Szegedy, utilizando tesselação (agrupamento de vértices) e a caminhada a tempo contínuo. Para extrair dados estatísticos de uma determinada caminhada quântica, necessitamos fazer sua simulação. Neste trabalho, desenvolvemos o simulador Hiperwalk, um novo simulador de caminhadas quânticas, em grafos de uma e duas dimensões para as caminhadas com moeda e sem moeda utilizando tesselação. O Hiperwalk permite ao usuário efetuar simulações de caminhadas quânticas em grafos utilizando processamento de alto desempenho, mesmo que o usuário não saiba programação paralela. O usuário pode empregar os dispositivos de paralelismo como CPU, GPGPU e co-processadores para acelerar o processo geral da caminhada.

Abstract of Dissertation presented to LNCC/MCT as a partial fulfillment of the requirements for the degree of Master of Sciences (M.Sc.)

A NEW HIGH PERFORMANCE SIMULATOR OF QUANTUM WALKS

Aaron Bruno Leão

November, 2015

Advisor: Renato Portugal, D.Sc

The development of quantum algorithms is not a easy task. Elements such as entanglement and quantum paralelism, intrinsics to quantum computation, difficult this task. Quantum walks are crucial tools for development of algorithms, mainly search algorithms. There are many types of quantum walks: with coin toss, Szegedy's, using tessellation (grouping of vertices) and the continuous-time quantum walk. To extract statistics data of a quantum walk, we need to perform its simulation. In this work, we developed the simulator Hiperwalk, a new simulator of quantum walks in graphs of one and two dimension for the quantum walk with a coin toss and coinless using tessellation. The Hiperwalk allows the user to perform simulations of quantum walks in graphs using high performance computing (HPC), even though the user does not knowing parallel programming. The user can employ the parallel devices such as CPU, GPGPU and accelerators cards to speedup the overall process of the walk.

Sumário

1	Introdução	1
2	Caminhadas Aleatórias	4
2.1	Introdução à probabilidade	5
2.2	Cadeia de Markov	7
3	Computação quântica	10
3.1	Computação quântica	11
4	Caminhadas Quânticas	18
4.1	Caminhada com moeda a tempo discreto (DTQW)	18
4.1.1	Grafos unidimensionais finitos	19
4.1.2	Grafos unidimensionais infinitos	22
4.1.3	Grafos bidimensionais finitos	24
4.1.4	Grafos bidimensionais infinitos	28
4.2	Caminhada sem moeda	29
4.2.1	Grafos unidimensionais finitos	30
4.2.2	Grafos unidimensionais infinitos	34
4.2.3	Grafos bidimensionais finitos	34

4.2.4	Grafos bidimensionais infinitos	36
4.3	Outras caminhadas quânticas	37
4.3.1	Caminhada quântica de Szegedy	37
4.3.2	Caminhada quântica a tempo contínuo	38
5	Simulações computacionais usando Hiperwalk	40
5.1	O simulador Hiperwalk	41
5.2	Caminhada com moeda a tempo discreto, DTQW	42
5.2.1	DTQW em grafos bidimensionais	51
5.3	Caminhada sem moeda	54
5.3.1	Caminhada sem moeda em grafos bidimensionais	61
5.4	Custom walk	65
6	Conclusão	68
	Referências Bibliográficas	70

Lista de Figuras

4.1	Grafo ciclo de 10 vértices. As arestas não possuem direção e peso um.	19
4.2	Grafo 2D 4×4 .	24
4.3	Duas tesselações de um grafo ciclo unidimensional, a verde cobrindo os vértices na forma $2x$ e a amarela os vértices $2x + 1$.	30
4.4	Duas tesselações de um grafo ciclo unidimensional.	33
4.5	Duas tesselações, uma azul nos vértices x e y e uma vermelha nos vértices $x + 1$ e $y + 1$, ambas tesselações possuem polígonos 2×2 .	36
5.1	Sequências dos passos tomados pelo Hiperwalk.	42
5.2	Caminhada de Hadamard com a moeda em superposição uniforme para $x = 50$.	46
5.3	Evolução da DTQW para 100 passos com estado inicial dado pela equação 5.5, usando a moeda de Hadamard.	48
5.4	Dados estatísticos da DTQW para 100 passos com estado inicial dado pela equação 5.5 usando a moeda de Hadamard.	49
5.5	Evolução da DTQW para 100 passos com estado inicial dado pela equação 5.5 usando a moeda de Hadamard.	50
5.6	Caminhada usando as moedas de Fourier, Grover e Hadamard na malha diagonal.	52

5.7	Caminhada usando as moedas de Fourier, Grover e Hadamard na malha natural.	53
5.8	Caminhada usando moeda de Grover na malha de tamanho 30×50	54
5.9	Evolução da caminhada <i>coinless</i> para 100 passos com estado inicial dado pela equação 5.8 no grafo linha.	56
5.10	Caminhada sem moeda para polígonos de 3 vértices de deslocamento de 1 vértice da segunda tesselação para a primeira.	57
5.11	Caminhada sem moeda para polígonos de 3 vértices de deslocamento de 2 vértices da segunda tesselação para a primeira.	58
5.12	Caminhada sem moeda para polígonos de 4 vértices de deslocamento de 1 vértice da segunda tesselação para a primeira.	58
5.13	Caminhada sem moeda para polígonos de 4 vértices de deslocamento de 2 vértices da segunda tesselação para a primeira.	58
5.14	Caminhada sem moeda para polígonos de 4 vértices de deslocamento de 3 vértices da segunda tesselação para a primeira.	59
5.15	Caminhada sem moeda para polígonos de 5 vértices de deslocamento de 1 vértice da segunda tesselação para a primeira.	59
5.16	Caminhada sem moeda para polígonos de 5 vértices de deslocamento de 2 vértices da segunda tesselação para a primeira.	59
5.17	Caminhada sem moeda para polígonos de 5 vértices de deslocamento de 3 vértices da segunda tesselação para a primeira.	60
5.18	Caminhada sem moeda para polígonos de 5 vértices de deslocamento de 4 vértices da segunda tesselação para a primeira.	60
5.19	Caminhada <i>coinless</i> ao longo do eixo X na <i>lattice</i>	62
5.20	Caminhada <i>coinless</i> ao longo do eixo Y na <i>lattice</i>	63

5.21 Caminhada <i>coinless</i> na <i>lattice</i>	64
5.22 Speedup do Hiperwalk em relação ao QWalk utilizando CPU e GPU.	67

Lista de Tabelas

2.1	Distribuição de probabilidade da caminhada do bêbado começando em $x = x_0$	9
-----	--	---

Lista de Siglas e Abreviaturas

- AKR: algoritmo de Ambainis-Kempe-Rivosh.
- CPU: *Central Processing Unit*, (Unidade de Processamento Central).
- CQ: Computação Quântica.
- DTQW: *Discrete time quantum walk*, (Caminhada quântica a tempo discreto).
- k-SAT: problema SAT restrito a cláusulas com no máximo k literais.
- GPGPU: *General Purpose Graphics Processing Unit.*, (Unidade de Processamento Gráfico de Propósito Geral).
- GPL: *General Public License*, (Licença Pública Geral).
- NP: *nondeterministic polynomial*. Classe das linguagens reconhecidas por uma máquina de Turing não-determinística em tempo polinomial.
- PAD: Processamento de Alto Desempenho.
- PDF: *Probability Density Function*, (Função de Distribuição de Probabilidade).
- qubit: *quantum bit*, (bit quântico).
- SAT: Problema da satisfatibilidade.
- VAD: Variável Aleatória Discreta.

Capítulo 1

Introdução

A simulação de objetos físicos sempre foi de grande interesse para a humanidade desde os tempos mais antigos. A criação de modelos físicos e matemáticos e suas posteriores simulações ajudam a entender o comportamento do objeto físico em estudo. Feynman (1982), em seu trabalho intitulado *Simulating Physics with Computers*, argumentou que a simulação de sistemas quânticos usando máquinas de Turing determinísticas não é possível de forma eficiente. Para isso, deveria ser utilizado um computador feito com elementos obedientes à lei da mecânica quântica. Deutsch (1985) formalizou a máquina de Turing quântica, iniciando a pesquisa em computação quântica. Foram apresentados os algoritmos de Deutsch e Jozsa (1992), Shor (1994) e Grover (1997), mostrando que a computação quântica possui elementos como o paralelismo quântico e emaranhamento que possibilitam ganhos computacionais, em relação à computação clássica. Esses ganhos podem ser exponenciais em alguns casos.

Na computação clássica, temos o problema da *Satisfatibilidade*, dado uma fórmula booleana $\phi(x_1, \dots, x_k)$ de k variáveis booleanas, existe uma instância das variáveis booleanas, x_1, \dots, x_k , tal que $\phi(x_1, \dots, x_k) = 1$? O problema da Satisfatibilidade é da classe **NP**, (SIPSER, 2012). Uma solução estocástica ao problema 2-SAT, satisfatibilidade de duas variáveis, utilizando caminhadas aleatórias foi apresentada por Papadimitriou (1991), que definiu uma caminhada aleatória

nas variáveis booleanas, mostrando uma solução em tempo polinomial. Aharonov, Davidovich e Zagury (1993) cunharam o termo *quantum random walk*. Formalizando, assim, a contrapartida quântica da caminhada aleatória clássica.

A partir do trabalho de Aharonov, Davidovich e Zagury (1993), tivemos várias formulações das caminhadas quânticas. Meyer (1996), apresentou um autômato celular em uma dimensão que pôde ser comparado a uma caminhada quântica sem moeda no grafo ciclo. A formulação a tempo contínuo foi introduzida por Farhi e Gutmann (1997). Posteriormente Aharonov et al. (2001) formularam de forma mais completa a caminhada quântica discreta, chamada de caminhada quântica com moeda, podendo ser vista como ferramenta para criação de algoritmos de busca em grafos. Patel, Raghunathan e Rungta (2005) apresentaram uma caminhada quântica sem moeda com o operador de evolução dividido em dois suboperadores, um para cada direção do caminhante. O modelo de Patel foi redescoberto por Falk (2013) com o uso de agrupamentos dos vértices do grafo em polígonos. Polígonos neste trabalho serão os agrupamentos dos vértices dos grafos de acordo a formar um polígono.

Neste trabalho fazemos uma revisão matemática para as caminhadas a tempo discreto com e sem moeda e suas formulações utilizando cadeias de Markov. Desenvolvemos um novo simulador de alta performance de caminhadas quânticas, chamado *Hiperwalk*. O Hiperwalk efetua caminhadas quânticas com moedas e sem moeda e foi desenvolvido utilizando a linguagem de programação *Python* e o interpretador Neblina para efetuar a interface com os dispositivos de paralelismo. O Hiperwalk e o *speedup* adquirido em relação a um simulador não paralelizável foram apresentados na conferência CCIS 2014 em Assunção no Paraguai (LARA; LEÃO; PORTUGAL, 2014). O simulador pode ser encontrado no site `qubit.lncc.br`, junto com o seu manual.

Este trabalho está esquematizado da seguinte maneira. No Capítulo 2, fazemos uma revisão da matemática para as caminhadas aleatórias. No Capítulo

3, apresentamos os postulados da computação quântica e citamos alguns algoritmos. No Capítulo 4, exibimos os modelos com moeda formalizado por Aharonov et al. (2001) e uma generalização da caminhada sem moeda de Falk para polígonos diferentes. No Capítulo 5, expomos alguns casos conhecidos da literatura para a caminhada com moeda e mostramos alguns exemplos da caminhada sem moeda com vários polígonos distintos. Nas simulações são mostrados os gráficos da média e do desvio padrão das caminhadas.

Capítulo 2

Caminhadas Aleatórias

Robert Brown, em 1827, ao observar o movimento de grãos de pólen em uma superfície aquática, percebeu que o sistema macroscópico, o pólen, está imerso em um sistema microscópico. Brown, então, estimou que o comportamento do sistema microscópico aos olhos macroscópico é aleatório. Nascia aí o **Movimento Browniano** que pode ser modelado matematicamente por uma caminhada aleatória.

A caminhada aleatória é o modelo matemático de como uma partícula se movimenta, tomando em cada instante de tempo uma decisão aleatória da direção do seu passo. A caminhada aleatória é utilizada em muitas áreas da ciência, como economia na dinâmica de preços e mercado financeiro (SCALAS, 2006); na difusão de processos hidrodinâmicos, mostrado por Scheidegger (1955); e, em computação, como uma solução aleatória ao problema k -SAT. Abaixo, mostramos de uma maneira informal e didática como a tomada de passos consecutivos, onde a direção é decidida aleatoriamente, é uma caminhada aleatória.

Imagine você, leitor, indo para casa, saindo de um bar depois de ter perdido um pouco de suas coordenações motoras. Tratando o bar como o marco zero, você começa a sua caminhada para a sua casa, supondo que ela esteja em linha reta neste percurso. Encostado nas paredes da rua você dá um passo para frente com uma probabilidade p e um passo para trás com probabilidade $1 - p$. Esta sua trajetória é conhecida na literatura como “caminhada do bêbado”, (MLODINOW,

2008).

Para adentrar um pouco mais no assunto das caminhadas aleatórias, devemos, inicialmente, introduzir alguns conceitos básicos, tais como, espaço amostral, variáveis aleatórias e função de distribuição de probabilidade (PDF) e outras grandezas estatísticas.

2.1 Introdução à probabilidade

Para apresentar os conteúdos de caminhadas aleatórias, precisamos introduzir os elementos iniciais de probabilidade, tais como, espaço amostral, variável aleatória entre outros.

Definição 2.1.1 (Espaço amostral) O conjunto de todos os resultados possíveis de um experimento probabilístico é chamado de espaço amostral, denotado pela letra Ω .

Por exemplo, se o leitor deixar um livro numa mesa, batendo vento sobre ele, em algum momento o livro estará aberto em uma determinada página. O conjunto formado por todas as páginas possíveis em que o livro pode estar aberto ao acaso é chamado de espaço amostral. A cardinalidade de um espaço amostral é a quantidade total de elementos do espaço amostral, podendo ser finita ou infinita. No exemplo do livro, a cardinalidade do espaço amostral é o número de páginas do livro, não contando com a capa e a contracapa.

Definição 2.1.2 (Variável Aleatória Discreta) Supondo que tenhamos um experimento com um espaço amostral Ω . Uma variável aleatória é um mapeamento do espaço amostral para os números reais, $X : \Omega \rightarrow \mathbb{R}$ que é definida em todos os elementos do espaço amostral. Uma variável aleatória que toma valores finitos ou um infinito enumerável é chamada de **variável aleatória discreta** (VAD).

No exemplo do livro ao vento, a variável aleatória seria a probabilidade de o livro estar aberto na página x com probabilidade $\mathbb{P}(x)$.

Definição 2.1.3 (Distribuição de probabilidade) Seja $X : \Omega \rightarrow \mathbb{R}$ uma variável aleatória, se associarmos para cada $x_i \in \Omega$ um número real positivo $\mathbb{P}(x_i)$, onde $\mathbb{P}(x_i) = \mathbb{P}(X = x_i)$. Os números $\mathbb{P}(x_i)$ são chamados de distribuição de probabilidade de X se, e somente se, $\mathbb{P}(x_i) \geq 0$ e $\sum_{x_i \in \Omega} \mathbb{P}(x_i) = 1, \forall_i$.

Voltando ao exemplo do livro aberto ao acaso pelo vento, descontando a capa e a contracapa, se o nosso livro em questão possui n páginas e cada página possui igual chance de ser aberta pelo vento. Como a nossa variável aleatória toma os valores nas páginas abertas, cada página possui probabilidade $\mathbb{P}(x_i) = \frac{1}{n}, 1 \leq i \leq n$. Essa função de distribuição é conhecida como distribuição uniforme.

Uma função de distribuição muito usada em problemas de modelagem e na natureza é a chamada distribuição normal ou gaussiana (VENAGAS-ANDRACA, 2008).

Definição 2.1.4 (Valor esperado e variância e desvio padrão) Seja X uma VAD, o seu valor esperado é definido como

$$E[X] = \mu = \sum_i x_i \mathbb{P}(x_i). \quad (2.1)$$

A **variância** de uma PDF é definida por

$$\begin{aligned} \sigma^2 &= V[X] \\ &= E[(X - \mu)^2] \\ &= \sum_i (x_i - \mu)^2 \mathbb{P}(x_i). \end{aligned} \quad (2.2)$$

A raiz quadrada da variância é chamado de **desvio padrão**,

$$\sigma = \sqrt{\sum_i (x_i - \mu)^2 \mathbb{P}(x_i)}. \quad (2.3)$$

2.2 Cadeia de Markov

Entende-se por processo estocástico um conjunto de variáveis aleatórias indexadas pelo tempo. Um processo estocástico sem memória, ou seja, a evolução depende apenas do estado atual e de nenhum outro, é chamado de Cadeia de Markov. Boas obras para os estudos desta teoria são Bremaud (1998) e Norris (1997).

Definição 2.2.1 (Cadeia de Markov) Seja $\{X_i, i \in \mathbb{N} \cup \{0\}\}$ um conjunto de variáveis aleatórias discretas e seja S um sistema definido no espaço de estados $\{s_j, j \in \mathbb{N} \cup \{0\}\}$, seja X_t como o estado de S no instante de tempo t , dizemos que S está no estado $s_{j'} \leftrightarrow X_t = s_{j'}$. A sequência de variáveis $\{X_i\}$ forma uma Cadeia de Markov com distribuição inicial λ e matriz de transição P , se em cada instante de tempo t' , S esteja no estado s_i e existe uma prefixada probabilidade p_{ij} de que, no instante seguinte, S estará no estado s_j e p_{ij} não depende em quais estados a cadeia estava em instantes de tempo $t < t'$. Em outras palavras,

$$\mathbb{P}(X_{t+1} = s_j | X_t = s_{j-1}) = \mathbb{P}(X_{t+1} = s_j | X_t = s_{j-1} \wedge \cdots \wedge X_0 = s_0) = p_{ij}(t). \quad (2.4)$$

Lema 2.2.2 Seja P uma matriz de transição de uma cadeia de Markov, a entrada p_{ij} da matriz P^t fornece a probabilidade de a cadeia de Markov começado no estado s_i estar no estado s_j após t passos. Portanto, $\sum_i p_{ij} = 1, \forall j$.

Lema 2.2.3 Seja P uma matriz de transição de uma cadeia de Markov com distribuição de probabilidade inicial λ , onde λ é um vetor linha, a probabilidade de o sistema estar no estado s_i após t passos é a i -ésima entrada do vetor linha $\lambda^{(n)}$

dado por $\lambda^{(n)} = \lambda P^n$, notando que o vetor coluna de probabilidade $\lambda^{T^n} = P^{T^n} \lambda^T$ e $\sum_i P_{ij}^T = 1, \forall_j$.

Podemos modelar a caminhada do bêbado como uma Cadeia de Markov da seguinte maneira: supondo que haja apenas 5 posições possíveis para o nosso caminhante estar, $S = \{s_0, s_1, s_2, s_3, s_4\}$, com s_0 sendo sua casa e s_4 sendo o bar. No instante de tempo 0, início da caminhada, o caminhante possui a seguinte distribuição de probabilidade inicial

$$\lambda = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

E sua matriz de transição é

$$P = \begin{bmatrix} 1 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 1 \end{bmatrix}.$$

O estado do nosso caminhante após t lançamentos de moeda é

$$\lambda^t = P^t \lambda. \quad (2.5)$$

Abaixo segue a distribuição de probabilidade do nosso caminhante.

$\begin{array}{c} \text{x} \\ \text{t} \end{array}$	x_{-4}	x_{-3}	x_{-2}	x_{-1}	x_0	x_1	x_2	x_3	x_4
0					1				
1				$\frac{1}{2}$		$\frac{1}{2}$			
2			$\frac{1}{4}$		$\frac{1}{2}$		$\frac{1}{4}$		
3		$\frac{1}{8}$		$\frac{3}{8}$		$\frac{3}{8}$		$\frac{1}{8}$	
4	$\frac{1}{16}$		$\frac{1}{4}$		$\frac{3}{8}$		$\frac{1}{4}$		$\frac{1}{16}$

Tabela 2.1: Distribuição de probabilidade da caminhada do bêbado começando em $x = x_0$.

Podemos verificar que a posição média da nossa caminhada é x_0 e que o desvio padrão, que é a informação de quanto o caminhantes se desvia da média, é dado por \sqrt{t} .

A seguir são introduzidos os conceitos fundamentais para a computação quântica.

Capítulo 3

Computação quântica

Em 1900, David Hilbert proferiu uma palestra em Paris e mostrou ao mundo 23 problemas matemáticos que estavam sem resolução até então. Dentre esses 23 problemas estava o 10º problema enunciado da seguinte maneira: existe um procedimento mecânico de caráter geral que pudesse ser aplicado na resolução de qualquer problema matemático de uma determinada classe?

Turing (1936), para responder esse problema, formalizou o conceito de procedimento mecânico e acabou desenvolvendo o que ficou conhecido como Máquina de Turing. Esta é uma máquina abstrata que compreenderia o problema como sendo uma entrada codificada em algum alfabeto (OLIVEIRA, 2010). Para mais informações sobre máquina de Turing uma boa referência é Sipser (2012).

Os computadores digitais são implementações físicas do modelo abstrato de Turing. Nesses computadores a informação é codificada e processada usando circuitos eletrônicos. Moore (1965) constatou que, a cada dezoito meses, a capacidade dos computadores era dobrada, este fato ficou conhecido como lei de Moore. A lei de Moore estipula que, em 2020, um bit de informação será representado por um único átomo, enquanto, em 1950, era representado por 10^{19} átomos (OLIVEIRA, 2010). A mecânica clássica não consegue prever a dinâmica de um átomo, apenas aproximação probabilística.

A computação quântica é uma abordagem computacional da mecânica quântica

tica. Através das leis da mecânica quântica pode-se codificar informação em um sistema quântico, processá-la usando portas quânticas e transmiti-la em canais quânticos. A seguir, mostraremos os postulados e critérios físico-matemáticos para efetuar computação com sistemas quânticos, muito bem mostrados em (NIELSEN; CHUANG, 2003).

3.1 Computação quântica

Neste trabalho denotaremos um espaço de Hilbert de dimensão n por \mathcal{H}^n .

Postulado 3.1.1 (Espaço de estados) Para todo sistema físico isolado, podemos associar um espaço de Hilbert \mathcal{H} , chamado de **espaço de estados** do sistema. O sistema físico é totalmente descrito por seu **vetor de estado**, um vetor unitário $|\psi\rangle \in \mathcal{H}$. A dimensão de \mathcal{H} é definido pelo grau de liberdade do sistema físico em questão.

Utilizaremos neste trabalho a notação de Dirac, onde $|\cdot\rangle$ é chamada de *ket*, um vetor coluna, e o seu dual por $\langle\cdot|$, um vetor linha. O Postulado 3.1.1 implica que combinação linear de vetores de estado é vetor de estado, isso é conhecido como princípio da **superposição**. Em suma, qualquer vetor $|\psi\rangle$ escrito como uma superposição dos estados da base $|e_i\rangle$ com $\alpha_i \in \mathbb{C}$ pode ser um vetor de estado.

$$|\psi\rangle = \sum_i \alpha_i |e_i\rangle. \quad (3.1)$$

Na computação clássica temos o bit. Ele é a menor unidade de informação que pode ser manipulada. O bit, matematicamente, é definido como uma variável booleana que toma valores no conjunto $\{0, 1\}$. A sua contrapartida quântica é o **qubit**. O qubit é um vetor unitário que está num espaço de Hilbert de dimensão 2, $|\psi\rangle \in \mathcal{H}^2$. Exemplos físicos de qubits podem ser um fóton que pode possuir duas polarizações distintas (vertical e horizontal) (OLIVEIRA, 2010). Matematicamente

um qubit é representado da seguinte forma

$$|\psi\rangle = \alpha|p\rangle + \beta|q\rangle, \quad (3.2)$$

onde $\alpha, \beta \in \mathbb{C}$, são chamadas de amplitudes, $|\alpha|^2 + |\beta|^2 = 1$ e $\{|p\rangle, |q\rangle\}$ é uma base que expande \mathcal{H}^2 . A base $\{|p\rangle, |q\rangle\} = \{|0\rangle, |1\rangle\}$ é chamada de base computacional, como demonstrada abaixo

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad (3.3)$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (3.4)$$

Logo um qubit na base computacional é

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \quad (3.5)$$

A equação (3.5) é a representação matemática da superposição dos estados $|0\rangle$ e $|1\rangle$ com amplitudes α e β . Se o sistema físico em questão tiver grau de liberdade igual a três temos um **qutrit** e grau de liberdade maior, temos um **qudit**.

Postulado 3.1.2 (Evolução unitária) A evolução de um estado quântico fechado $|\psi\rangle$ é descrita pelo operador unitário U . Denotando por $|\psi(t)\rangle$ o estado quântico no tempo t , temos

$$|\psi(t_1)\rangle = U|\psi(t_0)\rangle. \quad (3.6)$$

O Postulado 3.1.2 apenas define as propriedades matemáticas que o operador U deve possuir para se dar a evolução temporal. Um operador U é unitário se, e somente se, $UU^\dagger = U^\dagger U = I$. Como construir o operador para descrever a

dinâmica do sistema quântico não é dito pelo postulado, apenas como o operador deve ser descrito matematicamente.

Na computação clássica, todo processamento da informação deve ser dado por um algoritmo, que por sua vez, é decomposto em um circuito de portas lógicas. Os bits de informação são transmitidos pelas portas lógicas e, no final do circuito, a saída do algoritmo pode ser avaliada. No caso da computação quântica não é muito diferente, os qubits são processados por portas quânticas. Como veremos no Postulado 3.1.2, a evolução temporal de um sistema quântico fechado é dado por um operador unitário e sabe-se que o qubit está em um espaço de Hilbert de dimensão 2, portanto, não é difícil deduzir que uma porta quântica que processa um qubit é uma matriz unitária $U \in \mathcal{H}^{2 \times 2}$. Segue abaixo alguns exemplos das portas lógicas de um qubit mais conhecidas.

$$\sigma_X \equiv X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \quad (3.7)$$

$$\sigma_Y \equiv Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}. \quad (3.8)$$

$$\sigma_Z \equiv Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (3.9)$$

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (3.10)$$

Sendo as portas (3.7)-(3.9) conhecidas como portas de Pauli e a porta (3.10) é chamada de porta de Hadamard.

Postulado 3.1.3 (Composição de sistemas quânticos) O espaço de estado de um sistema quântico composto é o produto tensorial dos espaços de estados

componentes.

Por exemplo, se tivermos n sistemas quânticos denotados por $|\psi_1\rangle, \dots, |\psi_n\rangle$ interagindo em um sistema quântico maior, o estado quântico total é denotado por

$$|\Psi\rangle = |\psi_1\rangle \otimes \dots \otimes |\psi_n\rangle. \quad (3.11)$$

Denotando por $|\psi\rangle^{\otimes n}$ o produto tensorial do estado quântico $|\psi\rangle$ com ele mesmo, para $n = 1$, temos o próprio estado $|\psi\rangle$. Se tivermos um qubit na base computacional, assumindo sem perda de generalidade que seja $|\psi\rangle = |0\rangle$, portanto, $|\psi\rangle \otimes |\psi\rangle = |0\rangle \otimes |0\rangle$, sendo denotado por $|00\rangle$ ou apenas simplesmente por $|0\rangle$. Sendo que esta última notação, $|0\rangle \in \mathcal{H}^4$, é de fácil confusão se não estiver bem especificada a dimensão do espaço de Hilbert com que se esteja trabalhando.

Voltando ao Postulado 3.1.2, a evolução unitária de n qubits será dada por um operador unitário $U' \in \mathcal{H}^{2^n \times 2^n}$. Supondo que possua dois qubits no estado $|0\rangle$ e queira aplicar a porta de Hadamard em ambos, logo o sistema após a evolução será

$$H \otimes H |00\rangle = H|0\rangle \otimes H|0\rangle \quad (3.12)$$

$$= \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \right) \quad (3.13)$$

$$= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \quad (3.14)$$

$$= \frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle). \quad (3.15)$$

Este estado é a superposição com igual amplitude de todos os estados possíveis do sistema composto, um fenômeno interessante para o qual não há contrapartida clássica. Assim como a superposição, o **emaranhamento** quântico não possui uma contrapartida clássica. O emaranhamento ocorre quando há um sistema quântico de n qubits, com $n > 1$, que não pode ser fatorado em um tensorial

de n qubits, por exemplo.

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \quad (3.16)$$

O estado quântico fechado é um vetor complexo, portanto, não conseguimos interpretar o que está acontecendo com ele quando o manipulamos com uma operação unitária. Para saber como está uma determinada propriedade, ou observável, o experimentalista deve efetuar uma medição no sistema quântico. Essa medição destrói o estado quântico e suas correlações e não necessariamente é descrita por operadores unitários.

Postulado 3.1.4 (Medição quântica) As medições quânticas são descritas por determinados operadores de medições $\{M_m\}$. Esses operadores atuam sobre o espaço de estados do sistema. Os índices m se referem aos possíveis resultados da medição. Se o estado de um sistema quântico for $|\psi\rangle$ imediatamente antes da medição, a probabilidade de um resultado m ocorrer é dada por:

$$\mathbb{P}(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle. \quad (3.17)$$

E o estado após a medição será:

$$|\psi'\rangle = \frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}. \quad (3.18)$$

Os operadores de medição devem obedecer à relação de completude

$$\sum_m M_m^\dagger M_m = I. \quad (3.19)$$

A base computacional para \mathbb{C}^2 é $\{|0\rangle, |1\rangle\}$. Para um qubit, o observável da medição na base computacional é a matriz Pauli Z, (3.9). A decomposição espectral

da matriz Z é dada por:

$$Z = P_{+1} - P_{-1}. \quad (3.20)$$

com $P_{+1} = |0\rangle\langle 0|$ e $P_{-1} = |1\rangle\langle 1|$. As possibilidades de resultado da medição são ± 1 . Se tivermos um qubit dado segundo a equação (3.5), as probabilidades associadas com os resultados possíveis são

$$\mathbb{P}(+1) = |\alpha|^2, \quad (3.21)$$

$$\mathbb{P}(-1) = |\beta|^2 \quad (3.22)$$

Como estamos trabalhando com distribuições de probabilidade, então

$$\mathbb{P}(+1) + \mathbb{P}(-1) = |\alpha|^2 + |\beta|^2 = 1. \quad (3.23)$$

Isto é conhecido como medição na base computacional.

Em 1985, David Deutsch generalizou o trabalho de Turing (1936) para a computação quântica, criando a **Máquina de Turing Quântica** (DEUTSCH; JOZSA, 1992). Após a generalização da máquina de Turing, Deutsch junto com Richard Jozsa, em 1992, publicaram o primeiro algoritmo que realmente efetua um ganho ao clássico, usando uma propriedade que somente computadores quânticos possuem, o **paralelismo quântico**. Shor (1994), da AT&T Bell Labs, publicou o artigo que apresentava um algoritmo de fatoração de números inteiros, que extraia um ganho exponencial quanto aos algoritmos clássicos e, por último, Lov Grover, em 1995, da Bell Labs publicou um algoritmo de busca em uma base de dados quântica que possui um *speedup* quadrático em relação ao clássico. Apesar de esse ganho não ser tão grande quanto o mostrado pelo o algoritmo de Shor, o algoritmo de Grover pode ser aplicado em uma variedade maior de problemas. O algoritmo de Grover é baseado em **reflexões** no espaço de Hilbert do sistema, uma reflexão no estado ortogonal ao elemento marcado e outra reflexão no espaço expandido

pelo vetor diagonal, também chamado de média. Para mais informações e estudos detalhados sobre o algoritmo de Grover a referência indicada é Portugal (2013) e mais sobre algoritmos que possuem as reflexões no espaço de Hilbert em seu cerne são, Whitfield (2012a), Whitfield (2012b).

Capítulo 4

Caminhadas Quânticas

Muitos fatos da mecânica quântica, como superposição, paralelismo quântico e até mesmo teleporte de informação quântica, não são intuitivos. Isto prejudica a criação de novos algoritmos quânticos. Aharonov, Davidovich e Zagury (1993), cunharam o termo *Quantum Random Walks*, **Caminhada Aleatória Quântica**. O trabalho de Aharonov, Davidovich e Zagury (1993) usava um sistema quântico a mais, atrelado à partícula quântica que representava a moeda. E esse sistema quântico auxiliar deveria possuir a mesma dimensão que o grau de liberdade da partícula. Meyer (1996) desenvolveu um modelo de autômato celular quântico, onde o vetor de estado representa a *lattice* e a interação com a vizinhança de cada célula da *lattice* é descrita por um operador unitário.

4.1 Caminhada com moeda a tempo discreto (DTQW)

A generalização da caminhada quântica introduzida por Aharonov, Davidovich e Zagury (1993) foi feita por Aharonov et al. (2001) para grafos regulares. Esse modelo de caminhada quântica com moeda foi utilizado em algoritmos de busca em Shenvi, Kempe e Whaley (2003) e Ambainis, Kempe e Rivosh (2004).

4.1.1 Grafos unidimensionais finitos

Nesse modelo temos uma partícula quântica que está disposta em um grafo ciclo de N vértices, os vértices estão equidistante de uma unidade, como na Figura 4.1. Segundo o postulado 3.1.1 temos um espaço \mathcal{H}_p , p denotando posição, associado à partícula. Esse espaço \mathcal{H}_p possui a base canônica $\mathcal{B}_p = \{|i\rangle : 0 \leq i \leq N-1\}$. Como deseja-se deslocar a partícula de acordo com um eventual lance de “moeda”, deve-se adicionar ao sistema quântico da posição, um sistema associado à quiralidade da partícula. Como estamos tratando em apenas uma dimensão o espaço desta quiralidade, o que chamaremos de moeda, a partir de agora, é \mathcal{H}_2 e a base canônica para o seu espaço é $\mathcal{B}_c = \{|j\rangle : j \in \{0, 1\}\}$, a moeda da caminhada pode ser implementada como um qubit, vide equação (3.5). Pelo postulado 3.1.3, temos que o sistema completo do caminhante é $\mathcal{H} = \mathcal{H}_2 \otimes \mathcal{H}_p$. Como o caminhante é quântico, ele pode ser uma superposição e o seu estado genérico é dado por

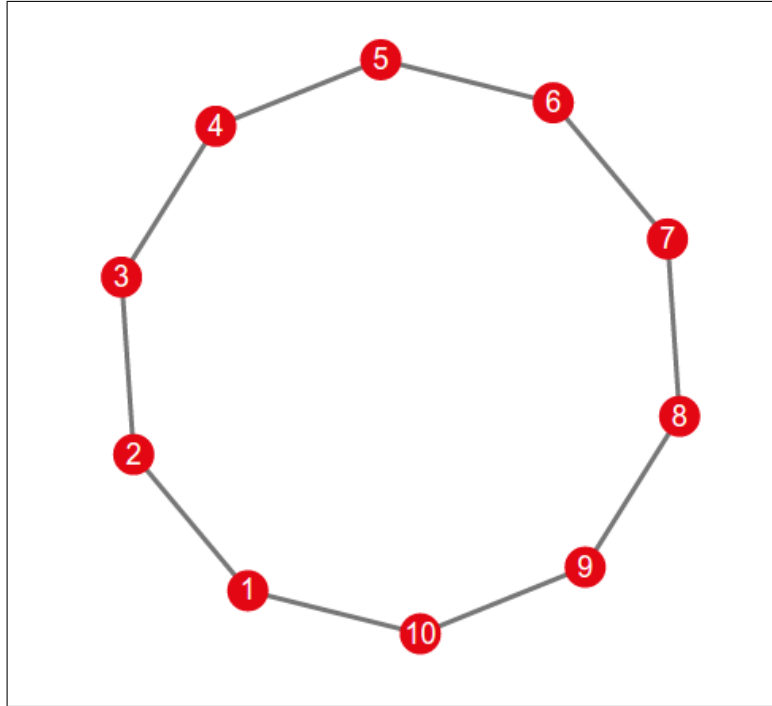


Figura 4.1: Grafo ciclo de 10 vértices. As arestas não possuem direção e peso um.

$$|\psi(t)\rangle = \sum_{c=0}^1 \sum_{x=0}^{N-1} \alpha_{c,x} |c\rangle |x\rangle, \quad (4.1)$$

com $\alpha_{c,x} \in \mathbb{C}$ e $\sum_{c,x} |\alpha_{c,x}|^2 = 1$.

Em uma caminhada aleatória, primeiro a moeda é lançada, medida, o seu resultado é verificado, e desloca a partícula para o lado resultante do lançamento da moeda. A evolução da caminhada quântica deve respeitar o postulado 3.1.2 e portanto ser unitária. Temos a construção do operador unitário da seguinte maneira:

Primeiramente para efetuar o lançamento da moeda, devemos definir um operador unitário $C \in \mathcal{L}(\mathcal{H}_2)$ que irá operar somente no espaço da moeda. Este operador pode ser qualquer operador desde que seja unitário. Um operador muito utilizado na literatura é um operador parametrizado por um ângulo $\theta \in [0, \pi]$, dado por

$$C(\theta) = \sigma_Z e^{-i\theta\sigma_Y}, \quad (4.2)$$

onde σ_Z e σ_Y são as matrizes de Pauli. Aplicando a exponencial das matrizes de Pauli obtêm-se

$$C(\theta) = \begin{pmatrix} \cos \theta & \sin \theta \\ \sin \theta & -\cos \theta \end{pmatrix}. \quad (4.3)$$

Observando que para $\theta = \frac{\pi}{4}$ têm-se o operador de Hadamard, equação 3.10.

Como o operador C é aplicado apenas no espaço da moeda, em cada passo do lançamento da moeda o espaço da posição é inalterado, portanto, é aplicado o operador identidade do espaço da posição, I_p , para que se mantenha coerente este passo do algoritmo.

Relembrando da seção 3.1.4 os operadores de medição da base computacional

do espaço da moeda são os projetores

$$\begin{aligned} R &= |0\rangle\langle 0|, \\ L &= |1\rangle\langle 1|. \end{aligned} \tag{4.4}$$

Deslocamos o caminhante de acordo com o resultado da medição, ou seja, aplicamos um operador de deslocamento nos elementos do espaço \mathcal{H}_p do caminhante. Definimos os operadores de deslocamento para a direita, S_+ , e para a esquerda, S_- respectivamente

$$\begin{aligned} S_+ &= \sum_{x=0}^{N-1} |x+1\rangle\langle x|, \\ S_- &= \sum_{x=0}^{N-1} |x-1\rangle\langle x|. \end{aligned} \tag{4.5}$$

Com $S_+, S_- \in \mathcal{L}(H_p)$. Para o caso de grafos finitos, o passo da caminhada quântica pode tomar valores maiores que o grafo e, portanto, na construção deste operador de deslocamento, deve-se estar ciente das condições de fronteiras tipo cíclicas, para que não haja nenhuma perda de amplitude e o operador se mantenha unitário.

Como desejamos que a partícula da posição seja deslocada para a direita quando a moeda for projetada para a direita e que seja deslocada para a esquerda quando a moeda for projetada para a esquerda, os dois sistemas quânticos, da moeda e da posição, estão se interagindo devemos relembrar do postulado 3.1.3 que nos diz que as correlações são dadas pelo produto tensorial entre as partes

$$S = R \otimes S_+ + L \otimes S_-. \tag{4.6}$$

O operador de deslocamento S pode ser escrito também usando a notação compacta do produto tensorial muito usada na literatura,

$$S = \sum_{c=0}^1 \sum_{x=0}^{N-1} |c, x + (-1)^c\rangle\langle c, x|. \tag{4.7}$$

O operador de evolução completo da caminhada é

$$U = S \circ (C \otimes I_p). \quad (4.8)$$

Se não houver nenhuma medição intermediária e a partir do lema 2.2.3, temos que o caminhante em qualquer instante de tempo é dado por

$$|\psi(t)\rangle = U^t |\psi(0)\rangle. \quad (4.9)$$

Sendo que em vez de termos uma distribuição de probabilidade, temos uma distribuição de amplitudes de probabilidade. Se após t passos de caminhada, efetuarmos uma medida na base canônica do subespaço da posição, a probabilidade de encontrarmos o caminhante na posição x no instante t é dada por

$$\mathbb{P}(x, t) = \sum_{c=0}^1 |\alpha_{c,x}(t)|^2, \quad (4.10)$$

onde $c \in \{0, 1\}$.

4.1.2 Grafos unidimensionais infinitos

A reta dos inteiros é o melhor exemplo de um grafo unidimensional infinito, o caminhante quântico, diferente da seção 4.1.1, não possui a possibilidade de causar somas de amplitudes. A base canônica do espaço da posição é $\mathcal{B}_p = \{|i\rangle : i \in \mathbb{Z}\}$. O estado genérico do caminhante será

$$|\psi(t)\rangle = \sum_{c=0}^1 \sum_{x=-\infty}^{\infty} \alpha_{c,x} |c\rangle |x\rangle, \quad (4.11)$$

com $\alpha_{c,x} \in \mathbb{C}$ e $\sum_{c,x} |\alpha_{c,x}|^2 = 1$. O operador de deslocamento é muito similar ao operador 4.7.

$$S = \sum_{c=0}^1 \sum_{x=-\infty}^{\infty} |c, x + (-1)^c\rangle \langle c, x|, \quad (4.12)$$

e o restante do operador unitário de evolução é idêntico ao operador (4.8), com o operador I_p e o operador moeda $C \in \mathcal{H}^{2 \times 2}$.

Um resultado muito conhecido na literatura é que a variância das caminhadas aleatórias varia linearmente no tempo, (KEMPE, 2008), ou seja, após t passos, temos

$$\sigma_{\text{clássico}}^2 \propto t, \quad (4.13)$$

e o desvio padrão

$$\sigma_{\text{clássico}} \propto \sqrt{t}, \quad (4.14)$$

enquanto a caminhada quântica possui

$$\sigma_{\text{quântico}}^2 \propto t^2, \quad (4.15)$$

e o desvio padrão

$$\sigma_{\text{quântico}} \propto t, \quad (4.16)$$

ou seja, o caminhante quântico se desloca no grafo unidimensional quadraticamente mais rápido em relação à sua contrapartida clássica. Vale ressaltar que esse tipo de caminhada preserva a paridade dos vértices do grafo, supondo que o caminhante esteja em $x = 0$ no início da caminhada, em tempos pares o caminhante ocupa vértices pares e em tempos ímpares ocupa vértices ímpares. Quando se define o operador de moeda C , como definido na equação (4.3), a caminhada quântica em uma dimensão pode ser escrita como o mapa descrito abaixo

$$\begin{aligned} a_x(t+1) &= a_{x+1}(t) \cos \theta + b_{x+1}(t) \sin \theta, \\ b_x(t+1) &= a_{x-1}(t) \sin \theta - b_{x-1}(t) \cos \theta. \end{aligned} \quad (4.17)$$

para mais informações sobre o mapa da caminhada unidimensional consultar a

referência Larrieu (2006).

4.1.3 Grafos bidimensionais finitos

Nesse caso, temos uma partícula quântica que se desloca em um grafo bidimensional onde cada vértice em x e y é equidistante por uma unidade, vide Figura 4.2. Como o grau de liberdade é 4, $x + 1$, $x - 1$, $y + 1$ e $y - 1$, a nossa moeda pode ser interpretada como 2 qubits, logo o seu espaço é $\mathcal{H}_c = \mathcal{H}_2 \otimes \mathcal{H}_2 = \mathcal{H}_4$. A base canônica para o espaço da moeda é $\mathcal{B}_C = \{|i, j\rangle : i, j \in \{0, 1\}\}$. O espaço da posição da partícula agora é composto pelo espaço \mathcal{H}_x e \mathcal{H}_y , $\mathcal{H}_p = \mathcal{H}_x \otimes \mathcal{H}_y$. Se a malha for um grafo tipo torus de dimensão $m \times n$, a base canônica para o espaço posição será $\mathcal{B}_p = \{|x, y\rangle : x \in \{0, m - 1\}, y \in \{0, n - 1\}\}$. O espaço do caminhante será

$$\mathcal{H} = \mathcal{H}_2 \otimes \mathcal{H}_2 \otimes \mathcal{H}_m \otimes \mathcal{H}_n. \quad (4.18)$$

O estado genérico do caminhante é dado por

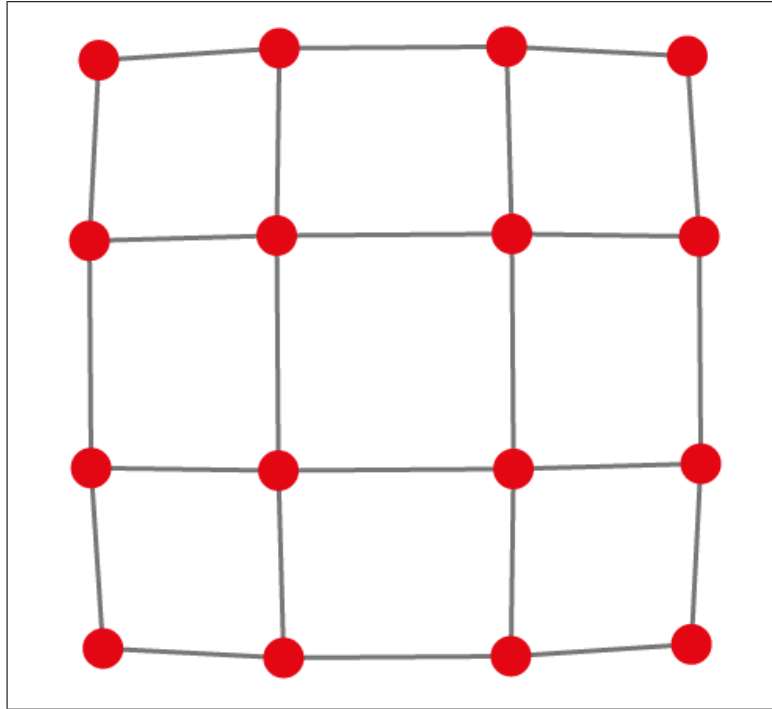


Figura 4.2: Grafo 2D 4×4.

$$\begin{aligned}
|\psi(t)\rangle &= \sum_{i=0}^1 \sum_{j=0}^1 \sum_{x=0}^m \sum_{y=0}^n \beta_{i,j,x,y}(t) |i\rangle |j\rangle |x\rangle |y\rangle \\
&= \sum_{i,j,x,y} \beta_{i,j,x,y}(t) |i, j, x, y\rangle.
\end{aligned} \tag{4.19}$$

com $\beta_{i,j,x,y}(t) \in \mathbb{C}$ e $\sum_{i,j,x,y} |\beta_{i,j,x,y}(t)|^2 = 1$.

O espaço da moeda é de dimensão 4, lembrando que a base do espaço da moeda pode ser reescrita usando uma notação mais compacta,

$$\begin{aligned}
\mathcal{B}_c &= \{|0, 0\rangle, |0, 1\rangle, |1, 0\rangle, |1, 1\rangle\} \\
&= \{|0\rangle, |1\rangle, |2\rangle, |3\rangle\}.
\end{aligned} \tag{4.20}$$

O operador moeda $C \in \mathcal{H}_4$ tem a sua definição livre, desde que mantenha unitário. Os operadores moedas mais conhecidos na literatura são:

(1) Hadamard

$$H_{j,k} = \frac{(-1)^{j \cdot k}}{\sqrt{N}}. \tag{4.21}$$

onde $j \cdot k$ é o produto interno da forma binária de j e k e N é a dimensão do espaço e N deve ser uma potência de 2. A sua forma matricial para $N = 4$ é

$$H_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}. \tag{4.22}$$

(2) Fourier

$$F_{j,k} = \frac{e^{(2\pi jk)/N}}{\sqrt{N}}. \tag{4.23}$$

Onde $i = \sqrt{-1}$ e N é a dimensão. O operador de Fourier para $N = 4$

$$F_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & 1 & i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}. \quad (4.24)$$

(3) Grover

$$G_{j,k} = \begin{cases} \frac{2}{N} - 1 & \text{se } j = k \\ \frac{2}{N} & \text{se } j \neq k \end{cases}. \quad (4.25)$$

e para $N = 4$

$$G_4 = \frac{1}{2} \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}. \quad (4.26)$$

O operador de deslocamento S pode ser definido da mesma forma que na seção anterior, associando cada projetor dos subespaços expandidos por cada elemento da base em uma direção do caminhante. Isso depende do tipo de malha. Se a malha for **diagonal**, temos

$$\begin{aligned} P_{x+1,y+1} &= |0\rangle\langle 0|, \\ P_{x+1,y-1} &= |1\rangle\langle 1|, \\ P_{x-1,y+1} &= |2\rangle\langle 2|, \\ P_{x-1,y-1} &= |3\rangle\langle 3|. \end{aligned} \quad (4.27)$$

O operador no espaço da posição também é dividido em quatro direções,

$$\begin{aligned}
S_{x+1,y+1} &= \sum_{\substack{x,y=0 \\ m,n}}^{m,n} |x+1, y+1\rangle\langle x, y|, \\
S_{x+1,y-1} &= \sum_{\substack{x,y=0 \\ m,n}}^{m,n} |x+1, y-1\rangle\langle x, y|, \\
S_{x-1,y+1} &= \sum_{\substack{x,y=0 \\ m,n}}^{m,n} |x-1, y+1\rangle\langle x, y|, \\
S_{x-1,y-1} &= \sum_{\substack{x,y=0 \\ m,n}}^{m,n} |x-1, y-1\rangle\langle x, y|.
\end{aligned} \tag{4.28}$$

O operador de deslocamento do caminhante em malha diagonal é dado por

$$\begin{aligned}
S_D = & P_{x+1,y+1} \otimes S_{x+1,y+1} + \\
& P_{x+1,y-1} \otimes S_{x+1,y-1} + \\
& P_{x-1,y+1} \otimes S_{x-1,y+1} + \\
& P_{x-1,y-1} \otimes S_{x-1,y-1}
\end{aligned} \tag{4.29}$$

ou em sua forma reduzida

$$S_D = \sum_{i,j=0}^1 \sum_{x,y=0}^{m,n} |i, j\rangle\langle i, j| \otimes |x + (-1)^i, y + (-1)^j\rangle\langle x, y|. \tag{4.30}$$

Em uma malha diagonal, o caminhante caminha pelos vértices diagonais. Um outro tipo de malha muito usada é a chamada malha **natural**, onde a malha matemática coincide com a malha física. Os detalhes do operador deslocamento da malha natural estão abordados abaixo.

$$\begin{aligned}
P_{x,y+1} &= |0\rangle\langle 0|, \\
P_{x+1,y} &= |1\rangle\langle 1|, \\
P_{x-1,y} &= |2\rangle\langle 2|, \\
P_{x,y-1} &= |3\rangle\langle 3|.
\end{aligned} \tag{4.31}$$

O operador no espaço da posição também é dividido em quatro direções,

$$\begin{aligned}
S_{x,y+1} &= \sum_{\substack{x,y=0 \\ m,n}}^{m,n} |x, y+1\rangle \langle x, y|, \\
S_{x+1,y} &= \sum_{\substack{x,y=0 \\ m,n}}^{m,n} |x+1, y\rangle \langle x, y|, \\
S_{x-1,y} &= \sum_{\substack{x,y=0 \\ m,n}}^{m,n} |x-1, y\rangle \langle x, y|, \\
S_{x,y-1} &= \sum_{x,y=0}^{m,n} |x, y-1\rangle \langle x, y|.
\end{aligned} \tag{4.32}$$

O operador de deslocamento do caminhante é dado por

$$\begin{aligned}
S_N = & P_{x,y+1} \otimes S_{x,y+1} + \\
& P_{x+1,y} \otimes S_{x+1,y} + \\
& P_{x-1,y} \otimes S_{x-1,y} + \\
& P_{x,y-1} \otimes S_{x,y-1},
\end{aligned} \tag{4.33}$$

ou em sua forma reduzida

$$S_N = \sum_{i,j=0}^1 \sum_{x,y=0}^{m,n} |i, j\rangle \langle i, j| \otimes |x + (-1)^i(1 - \delta_{j,d}), y + (-1)^j(\delta_{j,d})\rangle \langle x, y|. \tag{4.34}$$

4.1.4 Grafos bidimensionais infinitos

Agora, supondo que o nosso caminhante esteja em uma *lattice* bidimensional infinita, a parte espacial do caminhante \mathcal{H}_p possui base canônica $\mathcal{B}_p = \{|x, y\rangle : x, y \in \mathbb{Z}\}$. Nesse tipo de caminhada, o caminhante não atingirá a fronteira. O estado genérico do caminhante quântico na malha bidimensional infinita no instante de tempo t , será

$$|\psi(t)\rangle = \sum_{i,j=0}^1 \sum_{x,y=-\infty}^{\infty} \beta_{i,j,x,y}(t) |i, j, x, y\rangle, \tag{4.35}$$

com $\beta_{i,j,x,y}(t) \in \mathbb{C}$ e $\sum_{i,j,x,y} |\beta_{i,j,x,y}(t)|^2 = 1$.

O operador de deslocamento também respeita as equações (4.30) e (4.34), para *lattices* diagonal e natural.

$$S_D = \sum_{i,j=0}^1 \sum_{x,y=0}^{m,n} |i,j\rangle\langle i,j| \otimes |x + (-1)^i, y + (-1)^j\rangle\langle x, y| \quad (4.36)$$

e

$$S_N = \sum_{i,j=0}^1 \sum_{x,y=-\infty}^{\infty} |i,j\rangle\langle i,j| \otimes |x + (-1)^i(1 - \delta_{j,d}), y + (-1)^j(\delta_{j,d})\rangle\langle x, y|. \quad (4.37)$$

Para mais informações sobre as caminhadas DTQW temos o trabalho de Marquezino (2010) que possui simulações de mais casos, usando o simulador *QWalk* e Portugal (2013) para informações sobre algoritmos quânticos de busca.

4.2 Caminhada sem moeda

Patel, Raghunathan e Rungta (2005) introduziram um conceito de caminhada quântica que não utiliza a moeda quântica, ou seja, o espaço de Hilbert do caminhante é expandido apenas pelo espaço da posição, \mathcal{H}_p . Esse modelo de caminhada foi redescoberto por Falk (2013), onde utiliza a técnica de *tessellation* adaptado em português, **tesselagem**¹.

Essa técnica consiste em efetuar o cobrimento de uma superfície plana utilizando polígonos regulares. Esses polígonos se ajustam à superfície de acordo a formar uma **pavimentação**², sem sobreposição e sem lacuna, como um mosaico, permitindo diferentes tipos de formas geométricas³.

¹ <http://pt.wikipedia.org/wiki/Tessela>

² Para mais informações sobre pavimentação de superfícies: <http://www.uff.br/cdme/ppr/ppr-html/ppr-br.html>

³ Para mais informações sobre *tessellation*, <http://en.wikipedia.org/wiki/Tessellation> e <https://www.mathsisfun.com/geometry/tessellation.html>

Como definido por Aharonov et al. (2001), em uma caminhada quântica genérica em um grafo, o operador unitário da evolução deve respeitar as propriedades do grafo fazendo com que o caminhante mova apenas para vértices vizinhos. Nesse modelo de caminhada, dividimos a *lattice* em conjuntos de vértices, delimitados por polígonos regulares. Esses conjuntos são chamados de pacotes. O caminhante, quando estiver em um determinado pacote, poderá se deslocar para qualquer vértice dentro desse cada pacote e ,a partir deste vértice, ir para qualquer outro vértice dentro desse pacote em um passo de tempo. Os polígonos, como afirmado anteriormente, não poderão possuir intersecção. Cada polígono irá formar uma combinação linear e essa combinação linear formará a base para um hiperplano. Se possuírem intersecções, as combinações lineares serão linearmente dependentes. Após a formação do hiperplano, podemos definir uma reflexão em torno dele e evoluir o nosso caminhante.

4.2.1 Grafos unidimensionais finitos

O modelo proposto por Falk (2013) pode ser generalizado para uma *lattice* de dimensão D . A Figura 4.3 mostra um exemplo de duas tesselações em um grafo ciclo.

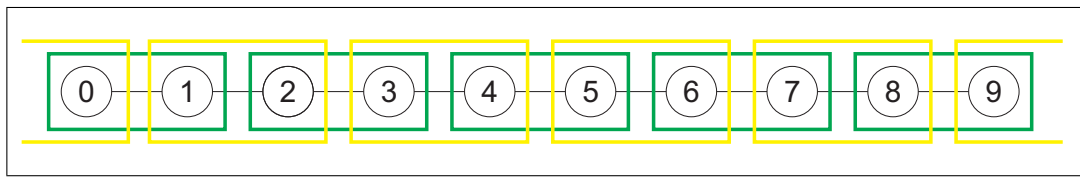


Figura 4.3: Duas tesselações de um grafo ciclo unidimensional, a verde cobrindo os vértices na forma $2x$ e a amarela os vértices $2x + 1$.

Definimos um polígono que englobe um conjunto de vértices do grafo. Correspondendo cada vértice do grafo em um elemento da base de \mathcal{H}_p , podemos gerar um hiperplano expandido pelas superposições dos vértices de cada polígono. Essas superposições são produzidas utilizando amplitudes pré-fixadas para cada vértice correspondente em polígonos distintos, ou seja, o n -ésimo vértice de cada polígono

da mesma tesselação terá a mesma amplitude. De maneira mais formal, definimos a base \mathcal{B}_i que expande os vértices do i -ésimo polígono

$$\mathcal{B}_i = \{|e_k\rangle, \dots, |e_l\rangle\}. \quad (4.38)$$

E seja $n = \#\mathcal{B}_i$, o número de vértices do grafo englobado pelos polígonos, podemos obtemos a superposição

$$|u_i\rangle = \sum_{j=0}^{n-1} \alpha_j |j'\rangle, \quad (4.39)$$

onde, $\alpha_j \in \mathbb{C}$, $\sum |\alpha_j|^2 = 1$, n é o número de vértices do i -ésimo polígono e $|j'\rangle \in \mathcal{B}_i$. Definimos os operadores que projetam um vetor $|\psi\rangle \in \mathcal{H}_p$ no subespaço expandido por $|u_i\rangle$,

$$\Pi_i = |u_i\rangle\langle u_i|, \quad (4.40)$$

notando que

$$\Pi_i^2 = \Pi_i \text{ e } \sum_{m=0}^N \Pi_m = I, \quad (4.41)$$

onde N é o número de polígonos da tesselação.

Após obtidos os projetores, podemos definir uma reflexão em torno do subespaço gerado por $|u_i\rangle$

$$R_i = 2\Pi_i - I, \quad (4.42)$$

notando que as reflexões são operações unitárias e hermitianas.

A ideia do algoritmo é produzir uma reflexão, sem viés, em torno de um hiperplano expandido por cada polígono da tesselação. Sem viés, quer dizer que cada polígono terá o mesmo *peso* na reflexão. Isso se dá por causa das mesmas amplitudes dos vértices equivalentes de cada polígono, representado pelo operador

abaixo

$$\mathcal{U} = 2 \sum_{m=0}^N \Pi_m - I. \quad (4.43)$$

Supondo, sem perda de generalidade, que tenhamos duas tesselações U_0 e U_1 , como na Figura 4.3, o resultado, será duas reflexões, \mathcal{U}_0 e \mathcal{U}_1 . A equação que define o caminhante em cada instante de tempo t é

$$|\psi(t)\rangle = \mathcal{U}^t |\psi(0)\rangle, \quad (4.44)$$

onde

$$\mathcal{U} = \mathcal{U}_1 \mathcal{U}_0. \quad (4.45)$$

Cada reflexão \mathcal{U}_0 e \mathcal{U}_1 é definida por

$$\mathcal{U}_0 = 2 \sum_i |u_i^0\rangle \langle u_i^0| - I, \quad (4.46)$$

$$\mathcal{U}_1 = 2 \sum_i |u_i^1\rangle \langle u_i^1| - I. \quad (4.47)$$

Como as duas tesselações possuem o mesmo número de polígonos, o índice i percorre os polígonos de cada tesselação. Por exemplo, na Figura 4.3, temos quatro polígonos, portanto, o conjunto $\{u_i^V : 0 \leq i \leq 3\}$ define os polígonos para a tesselação verde e o conjunto $\{u_i^A : 0 \leq i \leq 3\}$ define os polígonos para a tesselação amarela. Com u_0^V englobando os vértices 0 e 1, u_1^V , os vértices 2 e 3, u_2^V , os vértices 4 e 5 e u_3^V , os vértices 6 e 7 para a tesselação verde; e u_0^A englobando os vértices 1, 2, u_1^A , os vértices 3, 4, u_2^A , os vértices 5, 6 e u_3^A , os vértices 7, 0 para a tesselação amarela.

Supondo que em um grafo unidimensional temos duas tesselações, U_0 e U_1 , e que na tesselação U_0 temos polígonos englobando grupo de k_0 vértices e com

um espaçamento entre os polígonos de l_0 , denotando por $|u_i^0\rangle$ a superposição do i -ésimo polígono da tesselação U_0 , temos a seguinte forma genérica para $|u_i^0\rangle$

$$|u_i^0\rangle = \sum_{x=0}^{k_0-1} \alpha_x^0 |l_0 \cdot i + x\rangle. \quad (4.48)$$

Para a tesselação U_1 temos uma variável a mais, o deslocamento dos polígonos perante a tesselação U_0 , chamaremos de d_1 , logo poderemos escrever um polígono genérico da tesselação U_1 como

$$|u_i^1\rangle = \sum_{x=0}^{k_1-1} \alpha_x^1 |l_1 \cdot i + x + d_1\rangle, \quad (4.49)$$

onde $\alpha_x^{0,1} \in \mathbb{C}$, $\sum_x |\alpha_x^{0,1}|^2 = 1$, $l_{0,1} \geq k_{0,1}$ para manter os vetores independentes, caso contrário teríamos sobreposição de polígonos, ressaltando que na criação das tesselações não hajam lacunas. A Figura 4.4 apresenta alguns exemplos de tesselações.

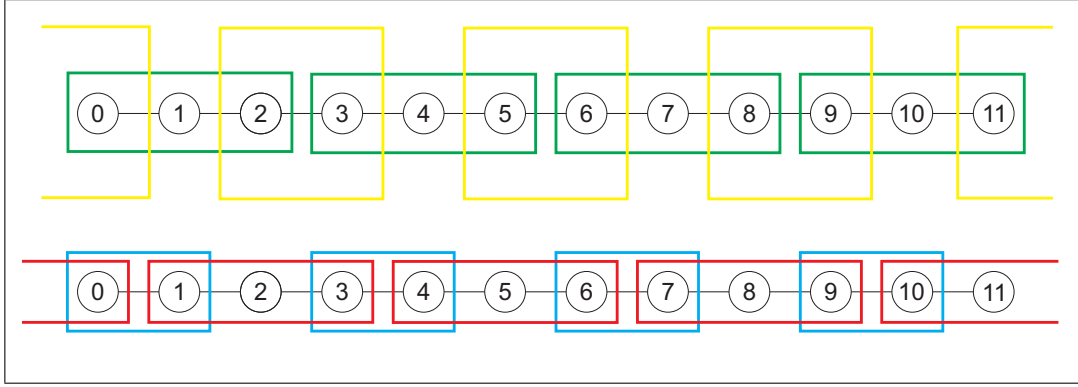


Figura 4.4: Duas tesselações de um grafo ciclo unidimensional.

Na Figura 4.4 na parte de cima, temos a tesselação verde cobrindo os vértices na forma $3x$ e a amarela os vértices $4x + 2$. Na parte debaixo a tesselação azul cobre os vértices $3x$ e a amarela os vértices $4x + 1$. Cada figura as tesselações possuem números diferentes de vértices

Em grafos finitos de qualquer dimensão deve-se estar ciente de que as tesselações devem englobar todo o grafo sem sobreposição e também não havendo vértices não englobados por algum pacote da tesselação.

4.2.2 Grafos unidimensionais infinitos

Como nesse tipo de grafo não há a possibilidade de sobreposição de amplitude do caminhante, usando a mesma tesselação da Figura 4.3 as equações (4.48) e (4.49) são reescritas como

$$|u_i^0\rangle = \sum_{x=-\infty}^{\infty} u_x^0 |l \cdot i + x\rangle \quad (4.50)$$

para a tesselação verde e

$$|u_i^1\rangle = \sum_{x=-\infty}^{\infty} u_x^1 |l \cdot i + x + d_1\rangle \quad (4.51)$$

para a tesselação amarela. O operador de evolução é o mesmo dado pela equação 4.45.

Para mais informações sobre as caminhadas quânticas em uma dimensão com e sem moeda, consultar a referência Portugal, Boettcher e Falkner (2015).

4.2.3 Grafos bidimensionais finitos

Neste tipo de grafo temos uma grid de $m \times n$. Na construção da tesselação devemos estar cientes de que a forma geométrica não deve extrapolar as fronteiras da grid, em outras palavras, se a tesselação possui um polígono da forma $m' \times n'$, portanto, tem-se $m = k_x m'$ e $n = k_y n'$ com $k_x, k_y \in \mathbb{Z}^+$.

Tendo em mente que neste tipo de grafo $\mathcal{H}_p = \mathcal{H}_x \otimes \mathcal{H}_y$ e não necessariamente temos uma malha quadrada, deveremos saber quantos polígonos teremos ao longo do eixo x e quantos ao longo do eixo y .

Supondo que tenhamos m' polígonos em x e n' polígonos em y , respectivamente, com k_x e k_y vértices, podemos reescrever as equações (4.48) e (4.49) como

$$|u_{i,j}\rangle = \sum_{x=0, y=0}^{k_x-1, k_y-1} \alpha_{x,y} |l_x \cdot i + x, l_y \cdot j + y\rangle, \quad (4.52)$$

onde $0 \leq i \leq m' - 1$; $0 \leq j \leq n' - 1$; k_x e k_y os número de vértice em x e y , respectivamente, l_x e l_y o deslocamento dos polígonos na tesselação, $\alpha_{x,y} \in \mathbb{C}$ e $\sum_{x,y} |\alpha_{x,y}|^2 = 1$.

Os projetores de cada polígono é descrito analogamente à equação (4.40)

$$\Pi_{i,j} = |u_{i,j}\rangle \langle u_{i,j}|. \quad (4.53)$$

Construímos as reflexões com os projetores da mesma forma ao caso $1D$

$$R = \sum_{i=0, j=0}^{m'-1, n'-1} 2\Pi_{i,j} - I. \quad (4.54)$$

Na Figura 4.5, temos duas tesselações, uma azul e uma vermelha. Esta última está deslocada de um vértice em x e um vértice em y da azul.

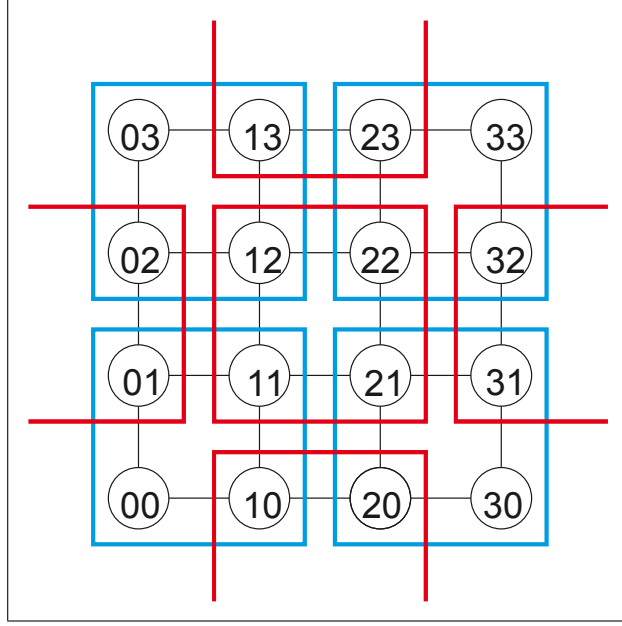


Figura 4.5: Duas tesselações, uma azul nos vértices x e y e uma vermelha nos vértices $x + 1$ e $y + 1$, ambas tesselações possuem polígonos 2×2 .

O estado genérico do caminhante no instante de tempo t será dado por

$$|\psi(t)\rangle = \sum_{x=0,y=0}^{m,n} \alpha_{x,y}(t) |x, y\rangle, \quad (4.55)$$

com $\alpha_{x,y}(t) \in \mathbb{C}$ e $\sum_{x,y} |\alpha_{x,y}(t)|^2 = 1$.

A seguir, generalizaremos a caminhada sem moeda em grafos bidimensionais infinitos.

4.2.4 Grafos bidimensionais infinitos

As tesselações neste tipo de grafo são mais gerais, pois podem receber polígonos quadriláteros de grandes dimensões, não ficando limitadas pelo tamanho da *lattice*. A equação (4.56) pode ser reescrita como

$$|u_{i,j}\rangle = \sum_{x=0,y=0}^{k_x-1,k_y-1} \alpha_{x,y} |l_x \cdot i + x, l_y \cdot j + y\rangle, \quad (4.56)$$

onde $-\infty \leq i, j \leq \infty$; k_x e k_y os número de vértice em x e y , respectivamente, l_x e l_y o deslocamento dos polígonos na tesselação, $\alpha_{x,y} \in \mathbb{C}$ e $\sum_{x,y} |\alpha_{x,y}|^2 = 1$. Os projetores e as reflexões são análogas às equações (4.53) e (4.54).

4.3 Outras caminhadas quânticas

Existem na literatura mais duas caminhadas quânticas: a caminhada quântica de Szegedy e caminhada a tempo contínuo.

4.3.1 Caminhada quântica de Szegedy

Seja G um grafo bipartido formado pelos conjuntos de vértices X e Y e suas arestas definidas pelas matrizes estocásticas P e Q que definem a probabilidade de transição entre os vértices de X e Y , respectivamente. Criando um espaço de Hilbert $\mathcal{H}^{|X||Y|}$ expandido por $\{|x\rangle|y\rangle : x \in X, y \in Y\}$. Podemos definir dois subespaços expandidos por

$$|\phi_x\rangle = |x\rangle \otimes \left(\sum_{y \in Y} \sqrt{p_{xy}} |y\rangle \right) = \sqrt{p_{xy}} |x\rangle |y\rangle, \quad (4.57)$$

$$|\sigma_y\rangle = \left(\sum_{x \in X} \sqrt{q_{yx}} |x\rangle \right) \otimes |y\rangle = \sqrt{q_{yx}} |x\rangle |y\rangle, \quad (4.58)$$

respectivamente, onde p_{xy} é a probabilidade de sair do vértice x e ir para o vértice y e q_{yx} é a probabilidade de sair do vértice y e ir para o x . Definindo duas reflexões em $\mathcal{H}^{|X||Y|}$

$$R_a = 2 \sum_{x \in X} |\psi_x\rangle \langle \phi_x| - I, \quad (4.59)$$

$$R_b = 2 \sum_{y \in Y} |\sigma_y\rangle \langle \sigma_y| - I, \quad (4.60)$$

portanto, temos o operador de evolução $U = R_b R_a$. O caminhante no instante t será

$$|\psi(t)\rangle = U^t |\psi(0)\rangle. \quad (4.61)$$

A caminhada de Szegedy foi inicialmente formalizada por Szegedy (2004). Para informações mais detalhadas a uma boa referência é Santos (2006), onde foi feito uma análise da caminhada de Szegedy e aplicação em busca de elementos marcados.

4.3.2 Caminhada quântica a tempo contínuo

A caminhada quântica a tempo contínuo foi formulada por Farhi e Gutmann (1997), tendo em seu cerne o formalismo das cadeias de Markov a tempo contínuo. Uma cadeia de Markov a tempo contínuo pode ser formalizada da seguinte forma: seja λ a distribuição de probabilidade inicial de um sistema e P a matriz de transição deste sistema então,

$$\lambda^t = e^{-Pt} \lambda \quad (4.62)$$

é o estado do sistema no instante de tempo t .

A ideia principal da caminhada é: se o caminhante está em um determinado nó de um grafo em um determinado tempo o caminhante move para um nó adjacente com uma probabilidade por unidade de tempo γ , onde γ é fixado. Isso significa que no tempo Δ_t onde $\gamma \Delta_t \ll 1$ a probabilidade de mover para um determinado nó é $\gamma \Delta_t \ll 1$. Podemos construir a matriz de transição do grafo como abaixo

$$H_{ij} = \begin{cases} 2\gamma, & \text{se } i=j; \\ -\gamma, & \text{se } i \neq j \text{ e adjacentes;} \\ 0, & \text{se } i \neq j \text{ não adjacentes} \end{cases} \quad (4.63)$$

O estado do caminhante no instante t é

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle. \quad (4.64)$$

No próximo capítulo, mostraremos as simulações das caminhadas com e sem moeda baseada em tesselações, usando processamento de alto desempenho.

Capítulo 5

Simulações computacionais usando Hiperwalk

Para efetuar um estudo analítico de uma caminhada quântica, devemos possuir um conjunto de equações que caracterizam a caminhada ao longo do tempo, chamada de mapa da caminhada quântica, equações (4.17) (ROMANELLI, 2009). Para muitos casos, esse mapa não é fácil de encontrar. Uma solução para esse problema é efetuar uma análise numérica com base em simulação computacional, onde, variando os parâmetros de entrada, tais como, estado quântico, operador de moeda, se caso houver, tempo da caminhada, pode-se inferir algumas propriedades analíticas da caminhada. Marquezino e Portugal (2008) apresentaram um simulador de caminhadas quânticas a tempo discreto, baseado no mapa desse tipo de caminhada. Neste capítulo, apresentamos um simulador de alta performance baseado na multiplicação matriz-matriz e matriz-vetor chamado, **Hiperwalk**, desenvolvido para ser um simulador capaz de efetuar caminhadas quânticas a tempo discreto, com e sem moeda e utiliza processamento de alto desempenho para efetuar as operações de multiplicação supracitadas.

5.1 O simulador Hiperwalk

O simulador foi desenvolvido em linguagem Python e Neblina¹. Hiperwalk é disponibilizado sob a licença de *software* GNU GPL² e pode ser utilizado para caminhadas quânticas com e sem moeda.

As caminhadas do simulador são baseadas na multiplicação matrix-matrix e matrix-vetor usual. Como em alguns casos o tamanho das matrizes cresce em razão do número de passos, em grafos infinitos, por exemplo, para fazer uma simulação eficiente, usamos uma plataforma de paralelismo em memória compartilhada chamada Neblina. O simulador escreve em arquivos no disco as matrizes de forma esparsa e repassa ao Neblina. Por sua vez, o Neblina efetua cada produto interno da i -ésima linha da matriz com o vetor em um núcleo de processamento.

O Hiperwalk possibilita várias caminhadas distintas, utilizando como interface de entrada um simples arquivo texto e como saída arquivos de texto com as informações das amplitudes do caminhante, a distribuição de probabilidade. O simulador possibilita a geração de gráficos utilizando o *Gnuplot*³, animação da caminhada ao longo do tempo com programas nativos do Linux. Atualmente, conta com a possibilidade de confirmar alguns resultados de caminhadas quânticas muito bem conhecidas na literatura, tal como, a caminhada de Hadamard. O funcionamento do simulador é totalmente dependente da instalação do interpretador Neblina. As seguintes etapas em seu processo de simulação:

- 1) Análise sintática ou *parsing* do arquivo de entrada.
- 2) Construção do estado quântico inicial, $|\psi(0)\rangle$ e escrita dos operadores unitários.
- 3) Chamada ao Neblina, repassando o estado quântico, operadores unitários,

¹ Neblina é utilizada para efetuar a simulação em alto desempenho e é voltado para a plataforma *Linux*, mais informações em <http://www.lncc.br/~pcslara/neblina/>

² www.gnu.org/licenses/licenses.pt-br.html

³ Mais informações em <http://www.gnuplot.info/>

tempo de simulação e outros parâmetros para efetuar a simulação da caminhada usando PAD.

4) Geração de gráficos, usando o *Gnuplot*.

O simulador produz gráficos da média e do desvio padrão para cada tempo, juntamente com a regressão linear obtida pelo método de mínimos quadrados, nas variáveis tempo e média ou tempo e desvio padrão. Estes gráficos são gerados pelo *Gnuplot*.

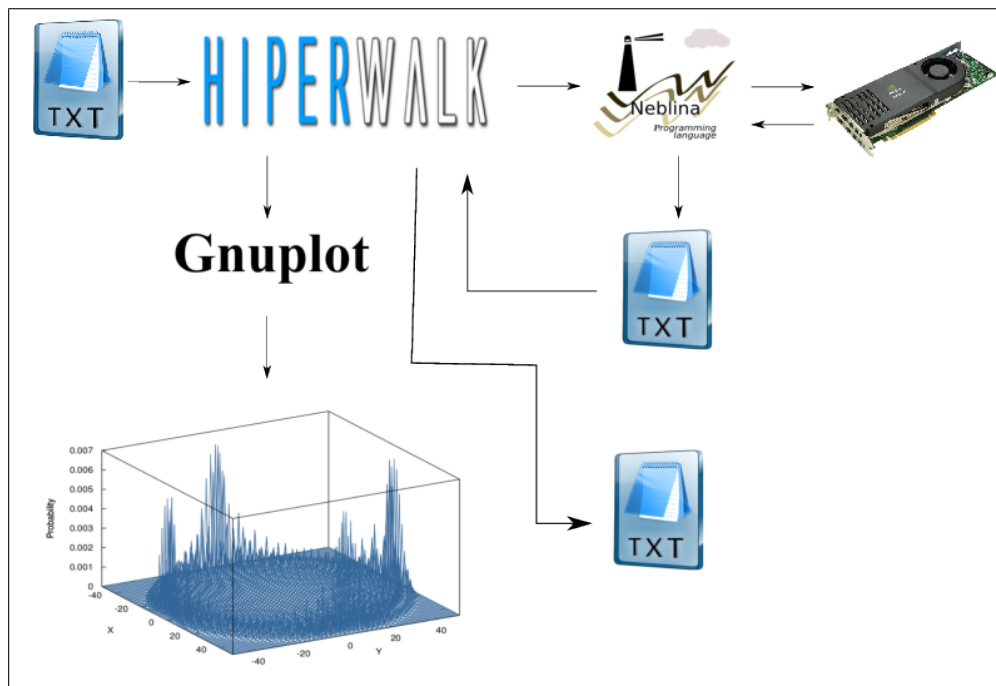


Figura 5.1: Sequências dos passos tomados pelo Hiperwalk.

A Figura 5.1 descreve as ações tomadas pelo Hiperwalk.

5.2 Caminhada com moeda a tempo discreto, DTQW

Para efetuar qualquer simulação deve-se gerar um arquivo em ASCII, utilizando o editor de texto da preferência do leitor, com os parâmetros primordiais como tipo de caminhada, tipo de grafo, quantidade de passos, operador moeda e

estado inicial. Todos os comandos devem ser inseridos em caixa alta.

Supondo que se deseja efetuar a seguinte caminhada quântica

$$|\psi(t)\rangle = (S \circ (H_2 \otimes I))^{100} |\psi(0)\rangle \quad (5.1)$$

na linha, e com estado inicial

$$|\psi(0)\rangle = \frac{|0\rangle - i|1\rangle}{\sqrt{2}} |50\rangle \quad (5.2)$$

o arquivo de entrada deverá conter os comandos definidos no código abaixo.

```
WALK DTQW
DIRECTORY DIR_DTQW1D
GRAPH LINE
STEPS 100
BEGINSTATE
0.70710678118654746 0 0 50
0 -0.70710678118654746 1 50
ENDSTATE
BEGINCOIN
HADAMARD 2
ENDCOIN
PLOTS TRUE
HARDWAREID 0
```

Cada comando deve ser inserido em linha distinta. O comando **WALK** define o tipo de caminhada no caso da caminhada de Hadamard é **DTQW**.

O comando **DIRECTORY** define em qual pasta será efetuada a caminhada, no exemplo, será criada uma pasta no local de execução chamada **DIR_DTQW1D**. Na omissão deste valor, o *default* do simulador é criar uma pasta chamada **HIPERWALK_DEFAULT_FOLDER** no local de execução do programa *hiperwalk*. **DIRECTORY** também aceita nome completo de diretórios. Todos os arquivos temporários gerados da simulação serão gravados no diretório parâmetro desse comando com o nome de **HIPERWALK_TEMP_<NOME_DO_ARQUIVO>** e serão apagados no final da simulação.

O tipo de grafo que se dará a caminhada é definido pelo comando **GRAPH**, se

será na linha ou no ciclo. Contemplando os grafos **CYCLE N**, **LINE**, **TORUS M N** e **LATTICE**, onde **M N** define o tamanho do grafo.

O comando **HARDWAREID 0** indica ao Neblina qual dispositivo de paralelismo utilizar. No *Linux* o usuário pode inserir o comando

```
$ neblina -l
```

para listar os dispositivos com suporte ao *OpenCL* que o Neblina pode utilizar.

STEPS informa quantos passos será efetuada a caminhada. Esse comando afeta diretamente o tempo de execução da simulação e também, em alguns casos, o consumo de memória. Para o caso de caminhadas em grafos infinitos (linha e *lattice*) se **STEPS** for grande, o caminhante se afastará mais da origem, ou das origens, assim, será necessário a construção de um estado quântico e operadores unitários de dimensão alta. Em outras palavras, o comando **STEPS** pode aumentar não só o tempo de execução do simulador, mas também o consumo de memória.

Já o estado quântico deve ser inserido entre os comandos **BEGINSTATE** e **ENDSTATE**. O estado quântico deve ser dado da seguinte forma:

$$|\psi\rangle = \sum_i \alpha_i |c_i\rangle |p_i\rangle \quad (5.3)$$

Com $\alpha_i \in \mathbb{C}$, $|c_i\rangle$ é o estado da moeda e $|p_i\rangle$ o estado da posição. Cada termo da soma deverá ser inserido em linhas distintas entre os comandos **BEGINSTATE** e **ENDSTATE** da seguinte forma:

$$\text{re}(\alpha_i) \text{ im}(\alpha_i) \ c_i \ p_i$$

Por exemplo, se deseja-se

$$|\psi(0)\rangle = \frac{1}{\sqrt{2}}(|0\rangle|50\rangle - i|1\rangle|50\rangle) \quad (5.4)$$

entre os blocos **BEGINSTATE** e **ENDSTATE** deverá ter a forma abaixo:

```
BEGINSTATE
0.70710678 0 0 50
0 -0.70710678 1 50
ENDSTATE
```

O comando `BEGINCOIN` indica ao simulador qual será a moeda utilizada na caminhada e seu tamanho, podendo ser `HADAMARD`, `GROVER`, `FOURIER`, todas devem ser seguidas do seu tamanho. Se desejado, poderá inserir uma moeda customizada entre os comandos `BEGINCOIN` e `ENDCOIN`. Abaixo temos os dois meios de indicar a moeda de Hadamard de dimensão 2×2 .

```
BEGINSTATE
0.70710678 0 0.70710678 0
0.70710678 0 -0.70710678 0
ENDSTATE
```

ou

```
BEGINSTATE
HADAMARD 2
ENDSTATE
```

A caminhada, utilizando a moeda de Hadamard com a condição inicial dada pela equação (5.4) e com 100 passos de caminhada, possui a PDF final descrita na Figura 5.2.

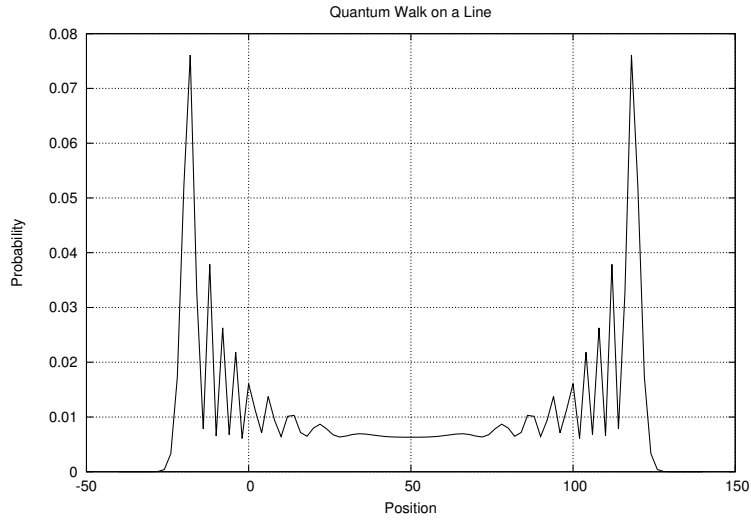


Figura 5.2: Caminhada de Hadamard com a moeda em superposição uniforme para $x = 50$.

O comando `PLOTS TRUE` informa ao simulador que serão gerados gráficos da distribuição de probabilidades final chamado *final_distribution.eps* e também dos dados estatísticos, do desvio padrão chamado de *standard_deviation.eps* e média chamado *mean.eps* pelo *gnuplot*. Para os gráficos dos dados estatísticos, o *gnuplot* efetua um ajuste polinomial de grau um, utilizando mínimos quadrados no conjunto de dados do tempo e da variável estatística desejada, portanto, é um gráfico de uma reta para o desvio padrão e outra reta para a média.

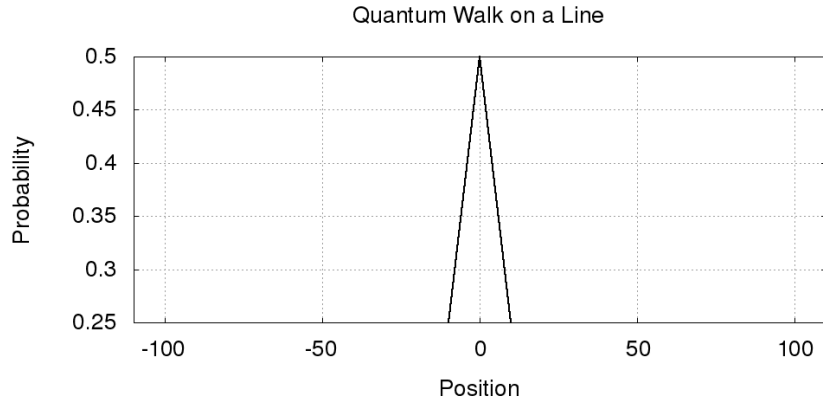
No final de toda simulação, o Hiperwalk gera os seguintes arquivos:

- *final_distribution.dat*, que contém as informações da posição e probabilidade.
- *final_state.dat*, contém as informações da posição, moeda e amplitudes.
- *statistics.dat*, para o caso da caminhada quântica em grafos 1D este arquivo possui quatro colunas. Para o tempo, média, segundo momento e desvio padrão respectivamente.

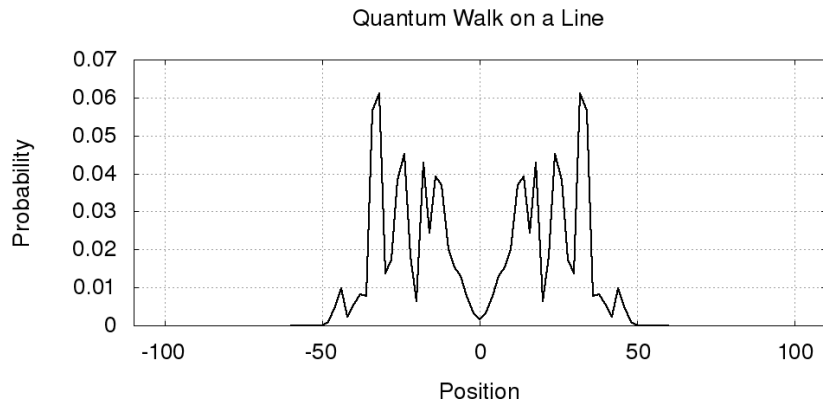
Podemos efetuar uma DTQW no grafo linha, usando a moeda de Hadamard

e com um estado inicial em superposição de mais de uma posição no grafo linha, por exemplo

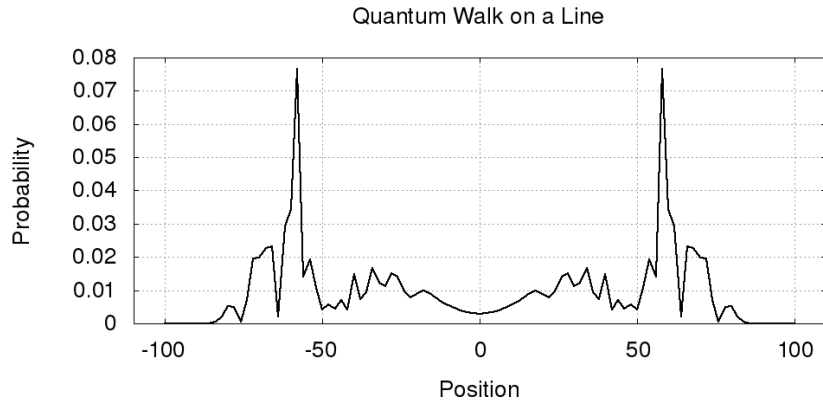
$$|\psi(0)\rangle = \frac{1}{2} (|0\rangle|-10\rangle + i|1\rangle|10\rangle - i|0\rangle|0\rangle + |1\rangle|0\rangle). \quad (5.5)$$



(a) $|\psi(0)\rangle$, estado do caminhante em $t = 0$.



(b) $|\psi(50)\rangle$, estado do caminhante em $t = 50$.

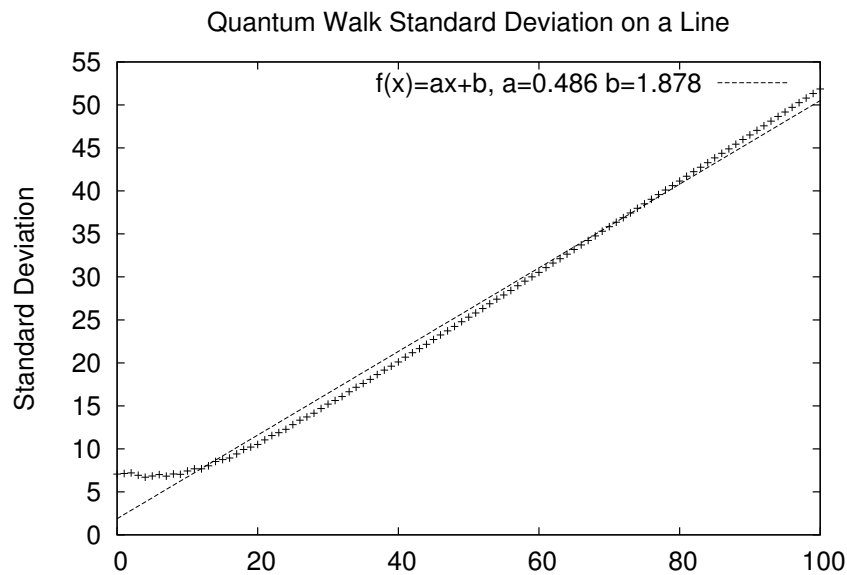


(c) $|\psi(100)\rangle$, estado do caminhante em $t = 100$.

Figura 5.3: Evolução da DTQW para 100 passos com estado inicial dado pela equação 5.5, usando a moeda de Hadamard.

Como a caminhada descrita na Figura 5.3 é simétrica com a origem do mo-

vimento, a média é 0. Abaixo podemos ver a equação linear que define o desvio padrão dessa caminhada.



(a) Gráfico do desvio padrão.

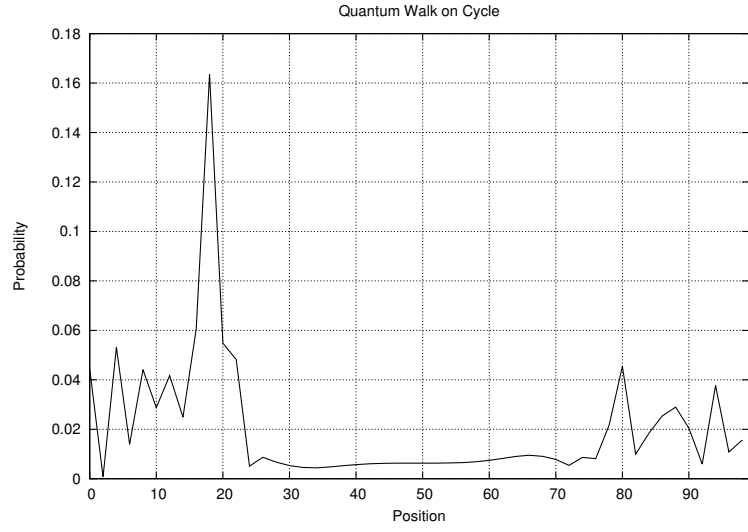
Figura 5.4: Dados estatísticos da DTQW para 100 passos com estado inicial dado pela equação 5.5 usando a moeda de Hadamard.

Para o grafo ciclo de N vértices, adaptamos a entrada do programa da seguinte forma

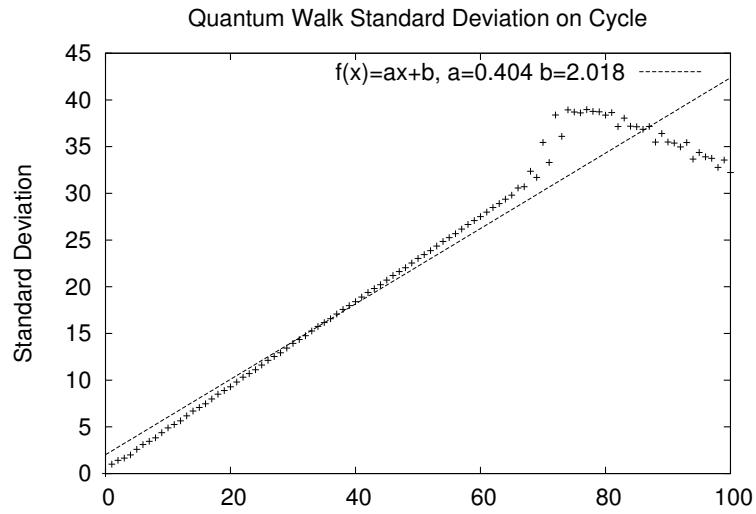
```
WALK DTQW
DIRECTORY DIR_DTQW1D
GRAPH CYCLE 100
STEPS 100
BEGINSTATE
1 0 0 50
ENDSTATE
BEGINCOIN
HADAMARD 2
ENDCOIN
PLOTS TRUE
```

No caso do arquivo de entrada acima as amplitudes do caminhante se sobrepõem por causa do número de passos, tamanho do grafo e codição inicial do

caminhante. Na Figura 5.5, as imagens da PDF e dos primeiro e segundo momentos.



(a) $|\psi(100)\rangle$



(b) Gráfico do desvio padrão.

Figura 5.5: Evolução da DTQW para 100 passos com estado inicial dado pela equação 5.5 usando a moeda de Hadamard.

O desvio padrão da Figura 5.5 se encontra com as amostras de pontos distantes da reta por causa da ciclicidade do grafo.

Já para grafos bidimensionais podemos analisar as três moedas mais comuns

na literatura, Fourier, Hadamard e Grover.

5.2.1 DTQW em grafos bidimensionais

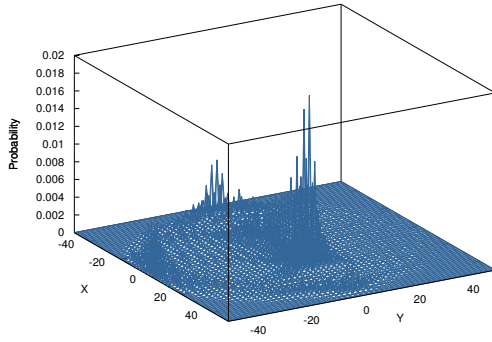
Para grafos bidimensionais, temos dois tipos de malhas, natural e diagonal como explicado na Seção 4.1.3. Os exemplos da DTQW, nesses tipos de grafos, serão exemplificados nos dois tipos de malha. A condição inicial do caminhante será dada pelas equações abaixo

$$|\psi(0)\rangle = \frac{1}{2} (|0, 0, 0\rangle - |1, 0, 0\rangle - |2, 0, 0\rangle + |3, 0, 0\rangle) \quad (5.6)$$

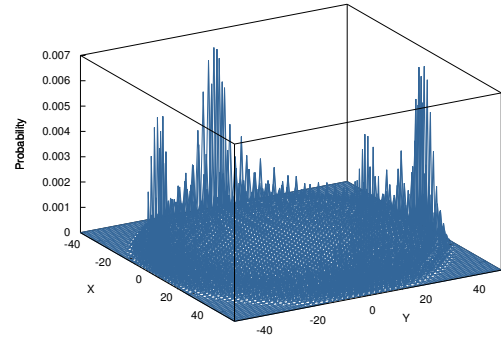
para as caminhadas, usando a moeda de Fourier e de Grover e

$$|\psi(0)\rangle = \frac{1}{2} (|0, 0, 0\rangle + i|1, 0, 0\rangle + i|2, 0, 0\rangle + |3, 0, 0\rangle). \quad (5.7)$$

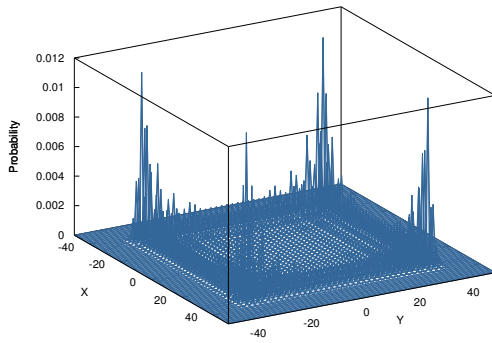
para a caminhada, usando a moeda de Hadamard para ambas malhas, diagonal e natural. Abaixo, temos os gráficos para 50 passos da caminhada.



(a) $|\psi(50)\rangle$ para a moeda de Fourier

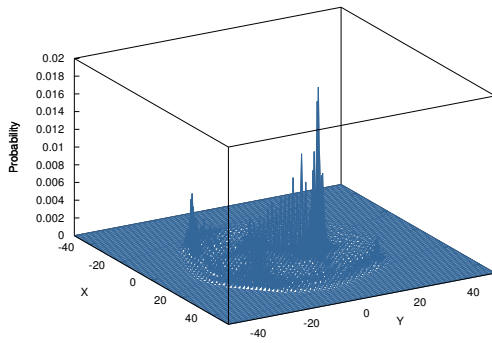


(b) $|\psi(50)\rangle$ para a moeda de Grover.

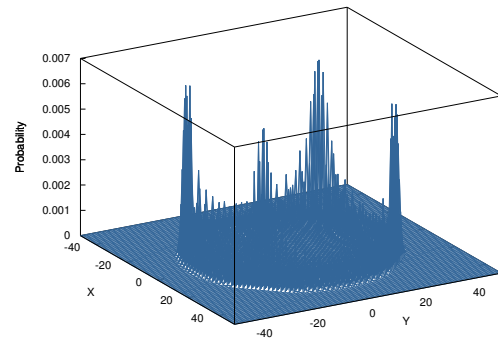


(c) $|\psi(50)\rangle$ para a moeda de Hadamard

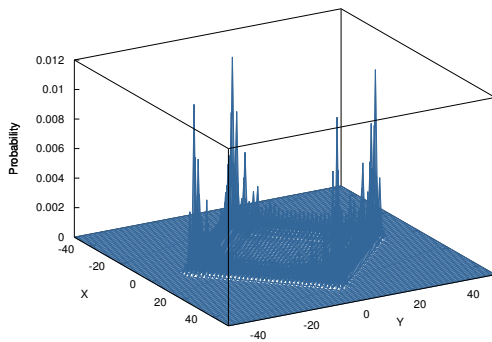
Figura 5.6: Caminhada usando as moedas de Fourier, Grover e Hadamard na malha diagonal.



(a) $|\psi(50)\rangle$ para a moeda de Fourier



(b) $|\psi(50)\rangle$ para a moeda de Grover.



(c) $|\psi(50)\rangle$ para a moeda de Hadamard

Figura 5.7: Caminhada usando as moedas de Fourier, Grover e Hadamard na malha natural.

Abaixo temos o exemplo do arquivo de entrada para a caminhada usando a moeda de Grover.

```
WALK DTQW
DIRECTORY DIR_DTQW2D
STEPS 50
GRAPH LATTICE
LATTYPE DIAGONAL
BEGINCOIN
GROVER 4
ENDCOIN
BEGINSTATE
0.5 0 0 0 0
-0.5 0 1 0 0
-0.5 0 2 0 0
```

```
0.5 0 3 0 0
ENDSTATE
```

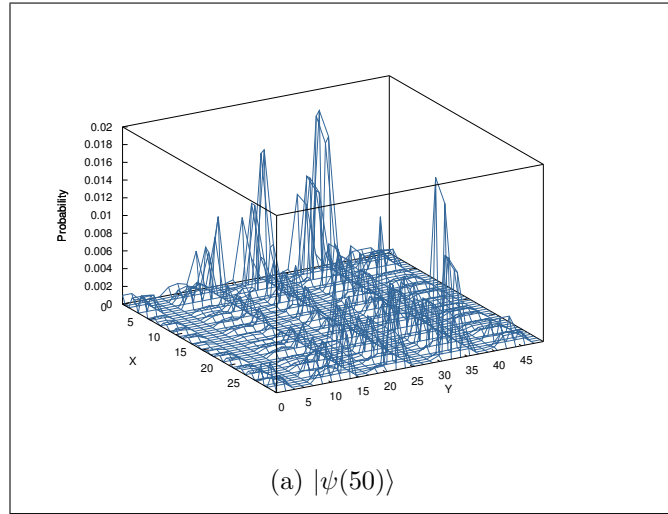


Figura 5.8: Caminhada usando moeda de Grover na malha de tamanho 30×50 .

Com uma simples substituição, podemos efetuar uma caminhada no grafo tipo torus retangular de dimensões $m \times n$. Para isso, basta substituir a linha GRAPH LATTICE por GRAPH TORUS $m \ n$.

A seguir, mostraremos algumas simulações usando a caminhada sem moeda.

5.3 Caminhada sem moeda

Uma primeira simulação, que pode ser efetuada da caminhada sem moeda, é uma similar à caminhada de Hadamard com moeda. Para isso, vamos montar o estado quântico inicial em uma superposição das posições $x = 0$ e $x = 1$ do grafo linha, seguindo a equação abaixo

$$|\psi(0)\rangle = \frac{1}{\sqrt{2}} (|0\rangle + i|1\rangle). \quad (5.8)$$

Comandos no arquivo de entrada

```
BEGINSTATE
```

```
0.70710678 0 0
0 0.70710678 1
ENDSTATE
```

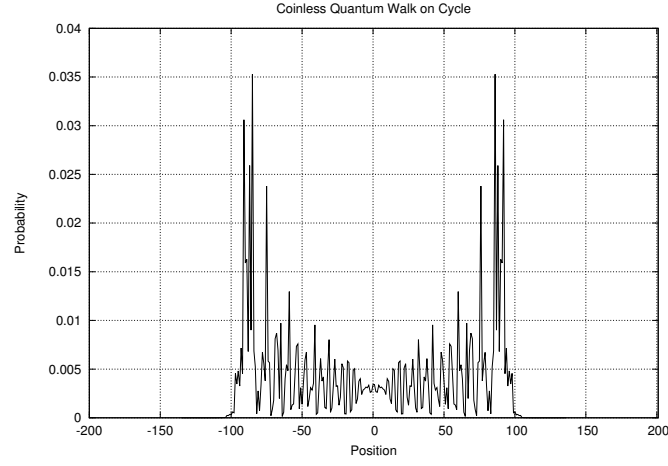
Serão duas tesselações cujos polígonos serão apenas de dois vértices do grafo. Com uma superposição específica dos vértices, diferente da distribuição uniforme dentro dos polígonos, ou seja, para o caso de dois vértices, uma distribuição uniforme seria as amplitudes $\frac{1}{\sqrt{2}}$ para cada vértice, pois nesta configuração o vetor de estado continuaria estacionado na posição de origem. Definimos o tipo de polígonos das tesselações e o deslocamento da segunda tesselação em relação à primeira pelos comandos

```
POLYGONS 2
DISPLACEMENT 1
```

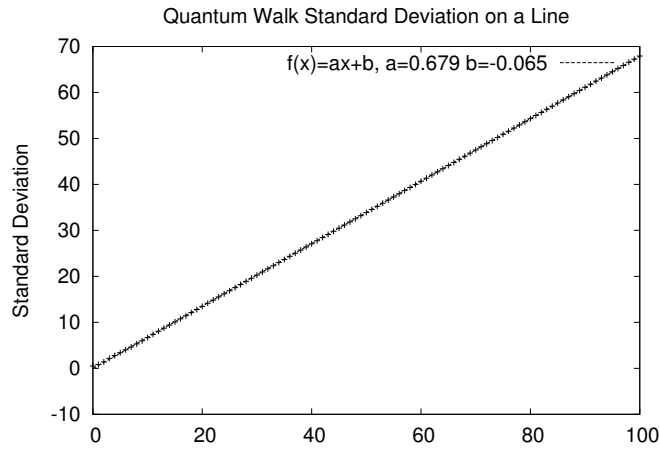
respectivamente. Para finalizar, informaremos quais são as amplitudes de cada vértice de cada pacote. Para isso, utilizaremos o comando `BEGINTESELLATION` e `ENDTESELLATION`.

```
BEGINTESELLATION
0.913375856139019 0 0.407117360747872 0
0.970592781760616 0 0.240727339523764 0
ENDTESELLATION
```

Na primeira linha dentro do comando, temos dois números complexos. O primeiro para o primeiro vértice dos polígonos que definem a primeira tesselação e o segundo para o segundo vértice do polígono. Na segunda linha, temos o mesmo para a segunda tesselação.



(a) $|\psi(100)\rangle$



(b) Gráfico do desvio padrão.

Figura 5.9: Evolução da caminhada *coinless* para 100 passos com estado inicial dado pela equação 5.8 no grafo linha.

O comando `POLYGONS 2` define a quantidade de vértices em cada polígono. Já `DISPLACEMENT 1` indica quantos vértices serão deslocados da segunda tesselação em relação à primeira. Para se definir as amplitudes de cada tesselação, temos o comando `BEGINTESELLATION` e `ENDTESELLATION`, onde cada linha entre os dois comandos define as amplitudes para cada tesselação, a primeira linha para a primeira tesselação e a segunda linha para a segunda tesselação. Temos nesse exemplo amplitudes arbitrárias para a primeira tesselação dada da forma

$$\text{re}(\alpha_1) \text{ im}(\alpha_1) \text{ re}(\alpha_2) \text{ im}(\alpha_2)$$

com $\alpha_{1,2} \in \mathbb{C}$, definindo as amplitudes dos vértices 1 e 2 de cada polígono.

Podemos construir no grafo linha tesselações de tamanhos arbitrários, vamos exemplificar alguns casos de tesselações com polígonos de tamanhos 3, 4 e 5, e variando o deslocamento da segunda tesselação de um vértice até a dimensão do polígono menos 1. Utilizando sempre a condição dada na equação 5.8 e uma distribuição normal de amplitudes na tesselações, ou seja, para polígonos de três vértices temos $\frac{1}{\sqrt{3}}$ para cada vértice, $\frac{1}{2}$ para polígonos de quatro vértices e $\frac{1}{\sqrt{5}}$ para polígonos de 5 vértices. Nas simulações abaixo utilizamos 100 passos.

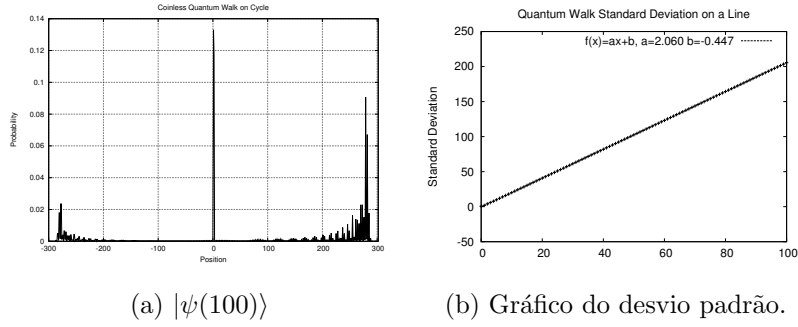
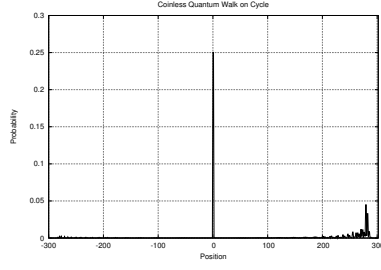
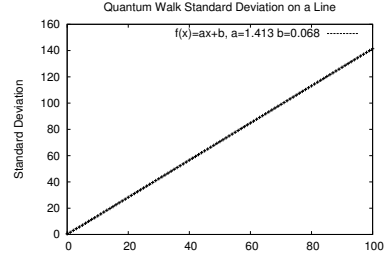


Figura 5.10: Caminhada sem moeda para polígonos de 3 vértices de deslocamento de 1 vértice da segunda tesselação para a primeira.

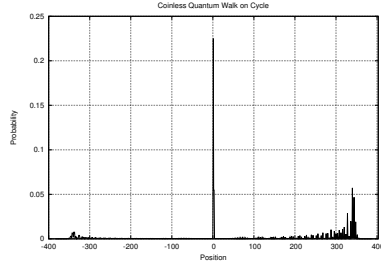


(a) $|\psi(100)\rangle$

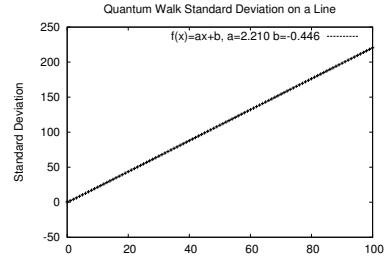


(b) Gráfico do desvio padrão.

Figura 5.11: Caminhada sem moeda para polígonos de 3 vértices de deslocamento de 2 vértices da segunda tesselação para a primeira.

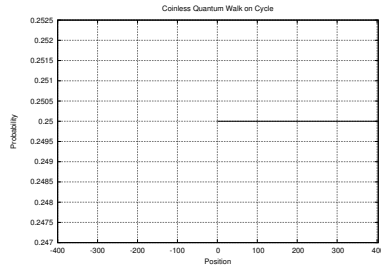


(a) $|\psi(100)\rangle$

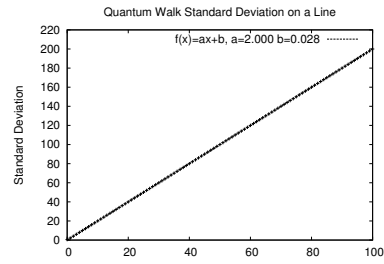


(b) Gráfico do desvio padrão.

Figura 5.12: Caminhada sem moeda para polígonos de 4 vértices de deslocamento de 1 vértice da segunda tesselação para a primeira.

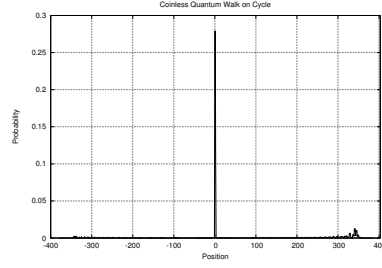


(a) $|\psi(100)\rangle$

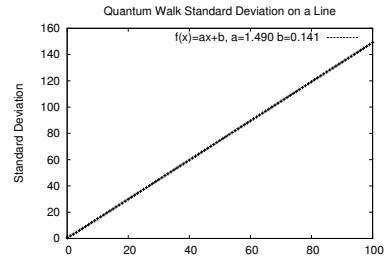


(b) Gráfico do desvio padrão.

Figura 5.13: Caminhada sem moeda para polígonos de 4 vértices de deslocamento de 2 vértices da segunda tesselação para a primeira.

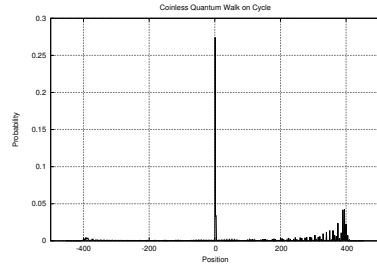


(a) $|\psi(100)\rangle$

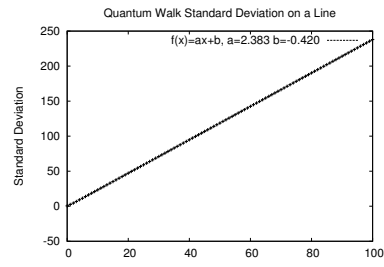


(b) Gráfico do desvio padrão.

Figura 5.14: Caminhada sem moeda para polígonos de 4 vértices de deslocamento de 3 vértices da segunda tesselação para a primeira.

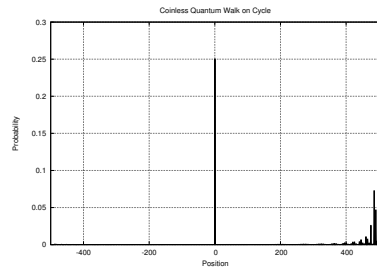


(a) $|\psi(100)\rangle$

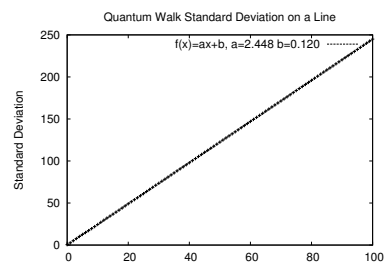


(b) Gráfico do desvio padrão.

Figura 5.15: Caminhada sem moeda para polígonos de 5 vértices de deslocamento de 1 vértice da segunda tesselação para a primeira.

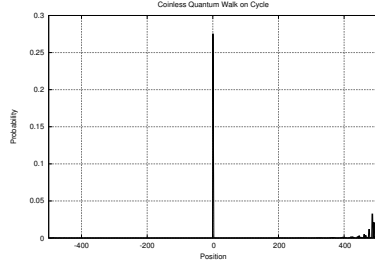


(a) $|\psi(100)\rangle$

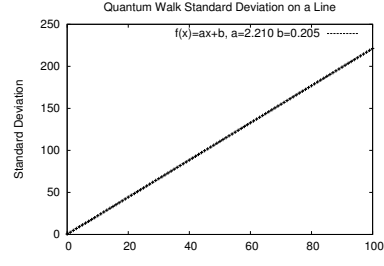


(b) Gráfico do desvio padrão.

Figura 5.16: Caminhada sem moeda para polígonos de 5 vértices de deslocamento de 2 vértices da segunda tesselação para a primeira.

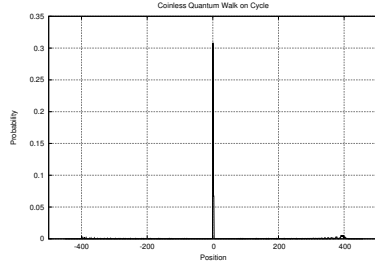


(a) $|\psi(100)\rangle$

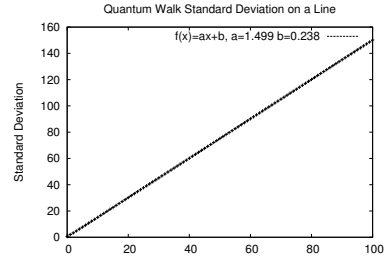


(b) Gráfico do desvio padrão.

Figura 5.17: Caminhada sem moeda para polígonos de 5 vértices de deslocamento de 3 vértices da segunda tesselação para a primeira.



(a) $|\psi(100)\rangle$



(b) Gráfico do desvio padrão.

Figura 5.18: Caminhada sem moeda para polígonos de 5 vértices de deslocamento de 4 vértices da segunda tesselação para a primeira.

Nas figuras 5.10-5.18, com exceção de 5.13, podemos observar o comportamento da distribuição de amplitudes mantendo constante na origem do movimento. Isso se dá devido ao estado inicial em superposição de algum autovetor do operador de evolução associado ao autovalor de valor 1. E outra observação interessante é do desvio padrão tendo o maior coeficiente linear na Figura 5.16. Isso ocorre por causa do tamanho do polígono e do deslocamento entre as tesselações.

5.3.1 Caminhada sem moeda em grafos bidimensionais

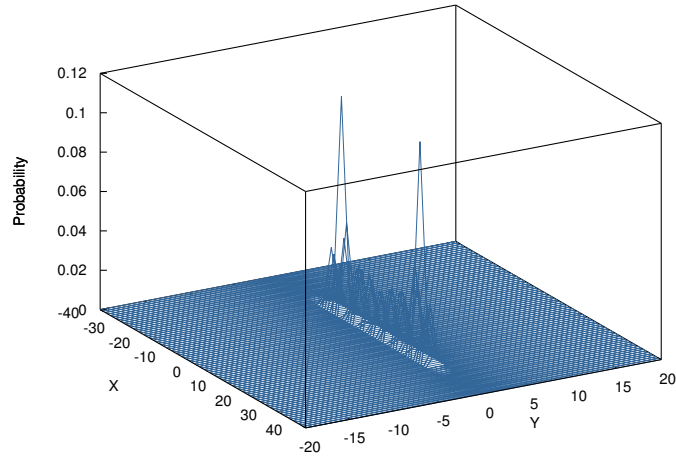
Exploraremos aqui algumas formas geométricas mais distintas para a *lattice*. Começaremos com uma simulação de polígonos 1D na *lattice*. Usando o estado inicial

$$|\psi(0)\rangle = \frac{1}{\sqrt{2}}(|0,0\rangle - i|1,0\rangle) \quad (5.9)$$

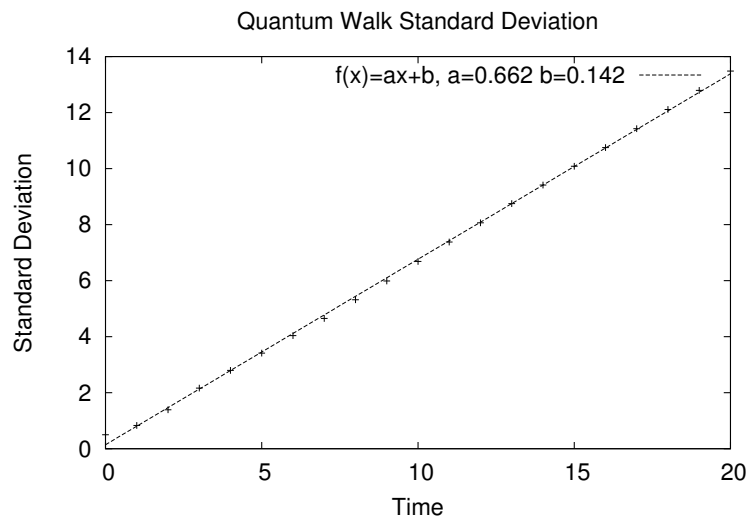
e polígonos segundo os parâmetros do arquivo de entrada abaixo

```
STEPS 20
GRAPH LATTICE
POLYGONS 2 1
DISPLACEMENT 1 0
BEGINTESELLATION
0.913375856139019 0 0.407117360747872 0
0.970592781760616 0 0.240727339523764 0
ENDTESELLATION
```

cujos gráficos abaixo são as representações dos dados estatísticos.



(a) $|\psi(20)\rangle$



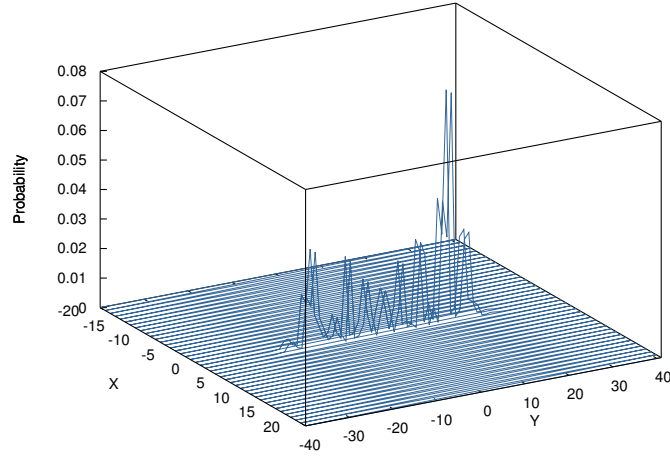
(b) Gráfico do desvio padrão.

Figura 5.19: Caminhada *coinless* ao longo do eixo X na *lattice*.

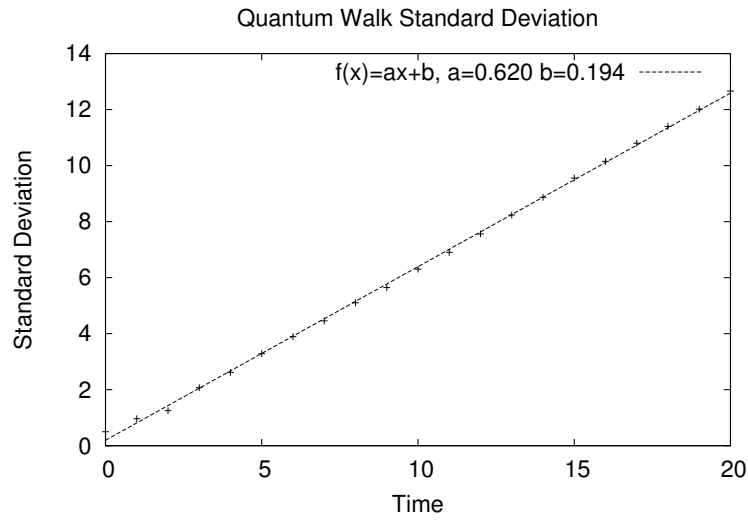
Fazendo agora um polígono 1×2 e deslocando de um vértice em Y , mudando apenas os parâmetros abaixo no arquivo de entrada

```
POLYGONS 1 2
DISPLACEMENT 0 1
```

Essa caminhada produz os seguintes gráficos de saída



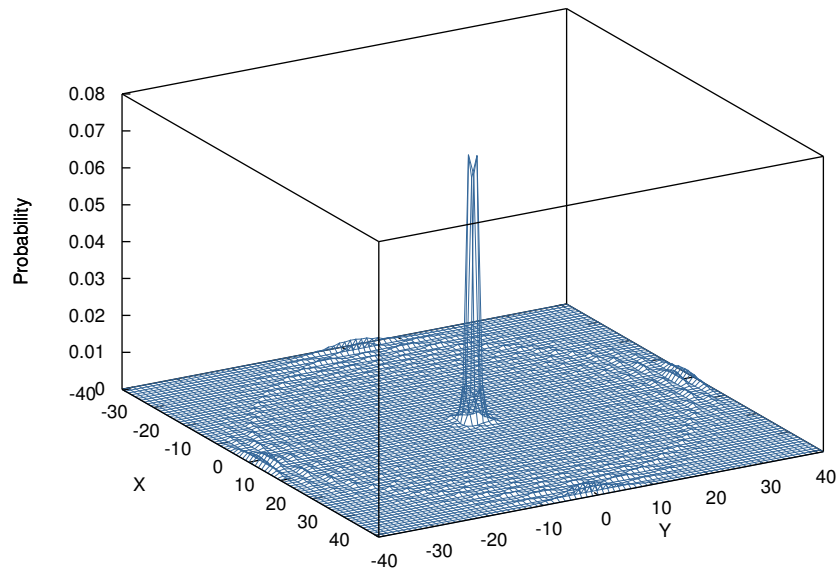
(a) $|\psi(20)\rangle$



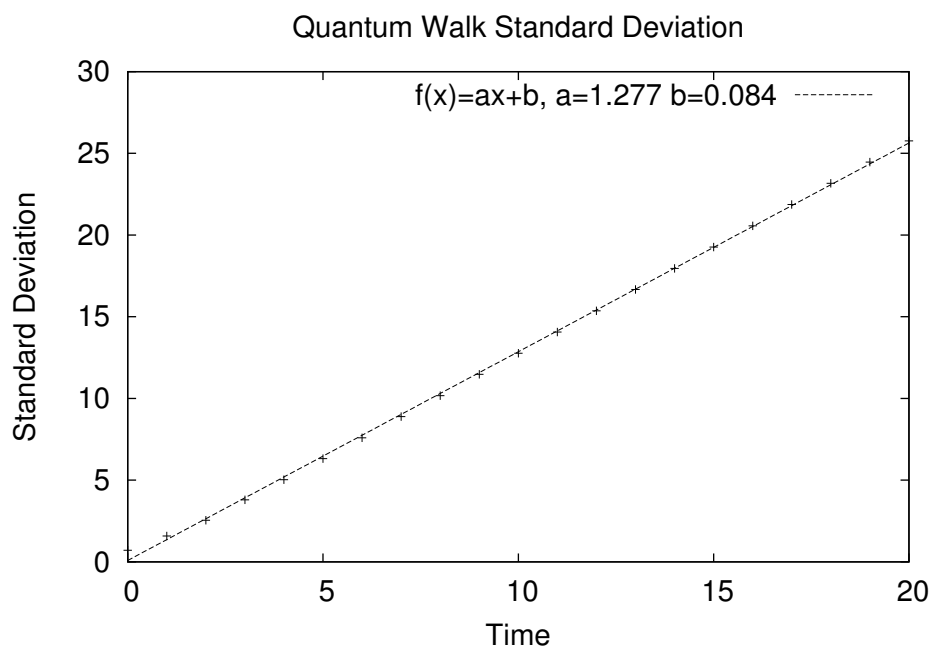
(b) Gráfico do desvio padrão.

Figura 5.20: Caminhada *coinless* ao longo do eixo Y na *lattice*.

Com polígono 2×2 e usando a distribuição uniforme das amplitudes nos vértice e deslocamento de 1 vértice em X e 1 vértice em Y , temos a seguinte distribuição de probabilidade para 20 passos da caminhada *coinless*.



(a) $|\psi(20)\rangle$



(b) Gráfico do desvio padrão.

Figura 5.21: Caminhada *coinless* na *lattice*.

Abaixo segue os parâmetros para a caminhada da Figura 5.21

POLYGONS 2 2
DISPLACEMENT 1 1


```

BEGINSTATE
0.70710678118654746 0 0 0
0.70710678118654746 0 1 1
ENDSTATE
BEGINTESELLATION
0.5 0 0.5 0 0.5 0 0.5 0
0.5 0 0.5 0 0.5 0 0.5 0
ENDTESSELLATION

```

Observamos que, na Figura 5.21, temos dois picos de probabilidades sendo mantidas na origem do movimento em $|0, 0\rangle$ e $|1, 1\rangle$. Este fenômeno é chamado de *localidade quântica* e é resultante do fato de o estado inicial ser uma superposição dos autovetores do operador de evolução associados ao autovalor 1.

5.4 Custom walk

O Hiperwalk também possui a opção de efetuar uma caminhada customizada pelo usuário chamada de **CUSTOM WALK**. Nesse tipo de caminhada, o Hiperwalk recebe do arquivo de entrada o nome do arquivo do vetor de estado e os nomes dos operadores que efetuarão a evolução e repassa-os ao Neblina. Em cada arquivo dos operadores, em cada linha, deverão apenas ser escritos os índices da matriz cujos elementos não são nulos. Os índices começam de 1, por exemplo, para operar a matriz de identidade 2×2 descrita nos arquivos *U0.dat* e σ_X descrita no arquivo *U1.dat* no vetor descrito pelo arquivo *psi0.dat* o arquivo de entrada do Hiperwalk deverá ser:

```

WALK CUSTOM
DIRECTORY DIR_CUSTOM
STEPS 10
INITIALSTATE psi0.dat
UNITARY U0.dat U1.dat
HARDWAREID 0
ALLSTATES 1

```

O comando **ALLSTATES n** salva a função de onda em todos os passos múltiplos de n . E o arquivo *U0.dat* deverá ser da forma

```

1 1 1 0
2 2 1 0

```

e o arquivo *U1.dat* deverá ser

```

1 2 1 0
2 1 1 0

```

O arquivo do vetor de estado possuirá apenas as amplitudes iniciais, incluindo as amplitudes nulas. Se $|\psi(0)\rangle = |0\rangle$, então o arquivo *psi0.dat* deverá ser

```

1 0
0 0

```

Com esses poucos parâmetros, podemos efetuar uma análise de caminhadas quânticas em grafos mais complexos como árvores, fractais etc.

Abaixo mostraremos o *speedup* produzido pelo Hiperwalk em relação ao simulador, também conhecido na literatura **QWalk**. O QWalk não utiliza paralelismo em sua estrutura para efetuar a caminhada. As execuções de ambos simuladores tomaram como dado de entrada o *Hadamar walk* com o caminhante na origem, $|\psi(0)\rangle = \left(\frac{|0\rangle - i|i\rangle}{\sqrt{2}}\right) |0\rangle$.

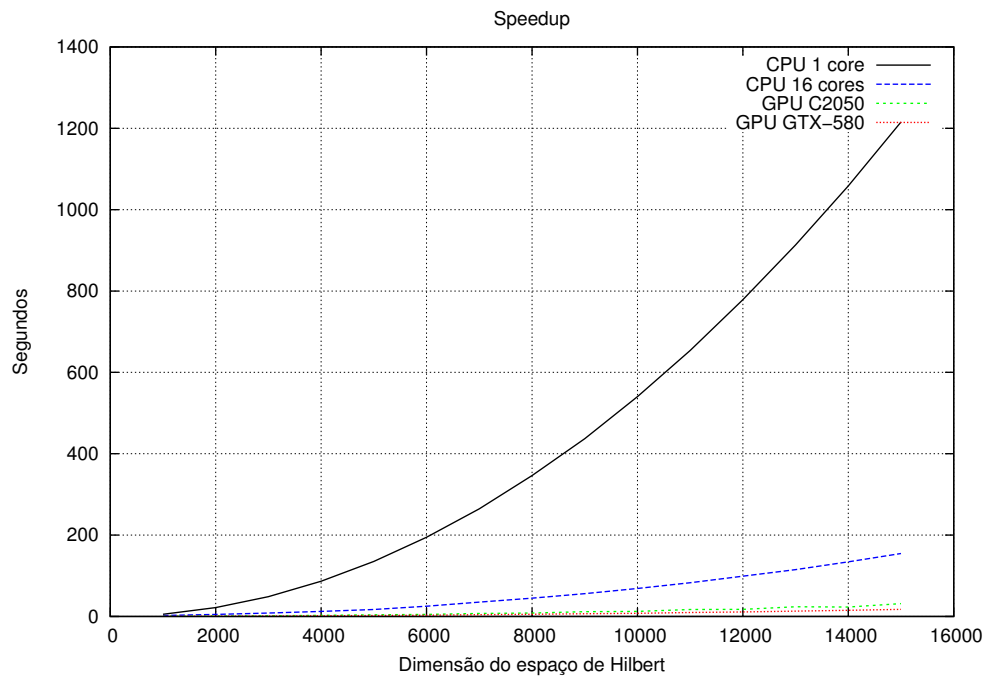


Figura 5.22: Speedup do Hiperwalk em relação ao QWalk utilizando CPU e GPU.

A linha preta da Figura 5.22 está relacionada ao QWalk e as outras ao Hiperwalk. O tempo foi calculado utilizando o comando *time* do Linux.

Capítulo 6

Conclusão

Nesse trabalho, dissertamos acerca das caminhadas quânticas com moeda DTQW e também sem moeda baseado em tesselação, que está sendo objeto de muita pesquisa, atualmente, por não possuir resultados analíticos para tipos distintos de tesselações. Apresentamos, também, um novo simulador de caminhadas quânticas que utiliza processamento de alto desempenho para efetuar as multiplicações matriz-matriz e matriz-vetor, o Hiperwalk. A construção desse simulador foi de tal forma que possibilita simulações de forma bem genérica dos estados quânticos e dos operadores unitários. As saídas estatísticas do simulador ajudam aos usuários retirar informações como *mixing time* e *hitting time*.

Na data da produção deste texto, a caminhada de Szegedy está sendo finalizada e acoplada de acordo com os padrões do Hiperwalk, assim como, a caminhada a tempo contínuo.

Como trabalhos futuros, temos as simulações com mais de duas tesselações para pavimentação do grafo. Esse tipo de tesselação é interessante por poder utilizar diversas formas de agrupamento de vértices para efetuar a caminhada.

A simulação de caminhadas quânticas com mais de uma partícula foi implementado por Ahlbrecht et al. (2012), na caminhada DTQW e utilizando dois caminhantes que se interagem sob uma fase aplicada nos vértices em que as partículas se encontraram. Nesse tipo de caminhada, o espaço de Hilbert é de dimensão

quadraticamente maior que a dimensão do espaço de Hilbert da caminhada de única partícula. A adição da implementação de estados iniciais genéricos, superposição das partículas e fases distintas, é interessante, pois, pode-se aferir algumas propriedades como centro de massa e distância relativa.

Referências Bibliográficas

AHARONOV, D. et al. Quantum walks on graphs. **Proc. 33th annual ACM symposium on Theory of computing, ACM**, p. 50–59, 2001.

AHARONOV, Y.; DAVIDOVICH, L.; ZAGURY, N. Quantum random walks. **Phys. Rev. A**, v. 48, p. 1687, 1993.

AHLBRECHT, A. et al. Molecular binding in interacting quantum walks. **New Journal of Physics**, v. 14, 2012.

AMBAINIS, A.; KEMPE, J.; RIVOSH, A. Coins make quantum walks faster. **Proc. 16th ACM-SIAM SODA**, p. 1099–1108, 2004.

BREMAUD, P. **Markov Chains: Gibbs Fields, Monte Carlo Simulation and Queues**. 1. ed. [S.l.]: Springer, 1998.

DEUTSCH, D. Quantum theory, the church-turing principle and the universal quantum computer. **Proceedings of the Royal Society of London**, v. 400, p. 99–117, 1985.

DEUTSCH, D.; JOZSA, R. Rapid solution of problems by quantum computation. **Proceedings of the Royal Society of London**, v. 439, p. 553–558, 1992.

FALK, M. D. Quantum search on the spatial grid. 2013. Disponível em: <arXiv:quant-ph/1303.4127>.

FARHI, E.; GUTMANN, S. Quantum Computation and Decision Trees. p. 28, 1997. ISSN 1050-2947. Disponível em: <arXiv:quant-ph/9706062>.

- FEYNMAN, R. P. Simulating physics with computers. **International Journal of Theoretical Physics**, v. 21, 1982.
- GROVER, L. Quantum mechanics helps in searching for a needle in a haystack. **Bell Labs**, 1997.
- KEMPE, J. Quantum random walks - an introductory overview. **Phys. Rev. A**, v. 44, 2008.
- LARA, P. C. S.; LEÃO, A.; PORTUGAL, R. Simulation of quantum walks using hpc. **Conference of Computational Interdisciplinary Science**, 2014.
- LARRIEU, A. A. **Caminata Cuántica en la Línea en Tiempo Discreto**. Dissertação (Mestrado) — Universidad de la Republica Facultad de Ciencias, 2006.
- MARQUEZINO, F. de L. **Análise, simulação e aplicações algorítmicas de caminhadas quânticas**. Tese (Doutorado) — Laboratório Nacional de Computação Científica, 2010.
- MARQUEZINO, F. L.; PORTUGAL, R. The qwalk simulator of quantum walks. **Computer Physics Communications**, v. 179, p. 359–369, 2008.
- MEYER, D. A. From quantum cellular automata to quantum lattice gases. **quant-ph/9604003**, 1996.
- MLODINOW, L. **O andar do bêbado : Como o acaso determina nossas vidas**. 1. ed. [S.l.]: Zahar, 2008.
- MOORE, G. E. Cramming more components onto integrated circuits. **Electronics**, v. 38, 1965.
- NIELSEN, M. A.; CHUANG, I. **Quantum computing and quantum information**. 1. ed. [S.l.]: Cambridge, 2003.
- NORRIS, J. R. **Markov Chains**. 1. ed. [S.l.]: Cambridge, 1997.

- OLIVEIRA, I. S. **Física moderna para iniciados, interessados e aficionados**. 1. ed. [S.l.]: Livraria da Física, 2010.
- PAPADIMITRIOU, C. On selecting a satisfying truth assignment. **Proceedings 32nd Annual Symposium of Foundations of Computer Science**, 1991. ISSN 02725428.
- PATEL, A.; RAGHUNATHAN, K. S.; RUNGTA, P. Quantum random walks do not need a coin toss. **Phys. Rev. A**, v. 71, 2005.
- PORTUGAL, R. **Quantum Walks and Search Algorithm**. 1. ed. [S.l.]: Springer, 2013.
- PORTUGAL, R.; BOETTCHER, S.; FALKNER, S. One-dimensional coinless quantum walks. **Phys. Rev. A** **91**, 052319, 2015.
- ROMANELLI, A. Driving quantum-walk spreading with the coin operator. **Phys. Rev. A**, v. 80, 2009.
- SANTOS, R. A. M. **Cadeias de Markov Quânticas**. Dissertação (Mestrado) — Laboratório Nacional de Computação Científica, 2006.
- SCALAS, E. The application of continuous-time random walks in finance and economics. **Elsevier**, v. 362, p. 225–239, 2006.
- SCHEIDEGGER, A. E. Statistical hydrodynamics in porous media. **Gennaio-Aprile**, v. 30, p. 17–26, 1955.
- SHENVI, N.; KEMPE, J.; WHALEY, K. B. Quantum random-walk search algorithm. **Phys. Rev. A**, v. 67, 2003.
- SHOR, P. Algorithms for quantum computation: Discrete logarithm and factoring. **AT T Bell Labs**, 1994.
- SIPSER, M. **Theory of Computation**. 3. ed. [S.l.]: Cengage Learning, 2012.

SZEGEDY, M. Quantum speed-up of markov chain based algorithms. **Proceedings of the 45th Symposium on Foundations of Computer Science**, 2004.

TURING, A. M. On computable numbers, with an application to the Entscheidungsproblem. v. 42, p. 230–265, 1936.

VENAGAS-ANDRACA, S. E. **Quantum walks for computer scientists**. 1. ed. [S.l.]: Morgan & Claypool, 2008.

WHITFIELD, J. D. Reflections in hilbert space i: Grover’s search. **Selected Lectures**, 2012.

WHITFIELD, J. D. Reflections in hilbert space iv: Quantum walks via szegedy’s scheme. **Selected Lectures**, 2012.