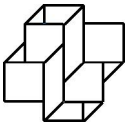

Laboratório Nacional de Computação Científica

Programa de Pós Graduação em Modelagem Computacional

Análise, simulações e aplicações algorítmicas de caminhadas quânticas

Por

Franklin de Lima Marquezino

PETRÓPOLIS, RJ - BRASIL

FEVEREIRO DE 2010

ANÁLISE, SIMULAÇÕES E APLICAÇÕES ALGORÍTMICAS DE CAMINHADAS QUÂNTICAS

Franklin de Lima Marquezino

TESE SUBMETIDA AO CORPO DOCENTE DO LABORATÓRIO NACIONAL DE COMPUTAÇÃO CIENTÍFICA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM MODELAGEM COMPUTACIONAL

Aprovada por:

Prof. Renato Portugal, D.Sc.
(Presidente)

Prof. Gilson Antonio Giraldi, D.Sc.

Prof. Raul Jose Donangelo, Ph.D.

Prof. Francisco Marcos de Assis, D.Sc.

PETRÓPOLIS, RJ - BRASIL
FEVEREIRO DE 2010

Marquezino, Franklin de Lima

M357a Análise, simulações e aplicações algorítmicas de caminhadas quânticas / Franklin de Lima Marquezino. Petropólis, RJ. : Laboratório Nacional de Computação Científica, 2010.

xix, 136p. : il.; 29 cm

Orientadores: Renato Portugal e Gonzalo Abal

Tese (Doutorado) – Laboratório Nacional de Computação Científica, 2010.

1. Computadores quânticos. 2. Algoritmos (Computação). 3. Computação quântica. 4. Caminhadas aleatórias. I. Portugal, Renato. II. Abal, Gonzalo. III. MCT/LNCC. IV. Título.

CDD – 004.1

"Por mais longa que seja a caminhada, o mais importante é dar o primeiro passo."
(Vinícius de Moraes)

Dedicatória

Ao bom e soberano Deus.

Soli Deo Gloria.

Agradecimentos

Muitas pessoas contribuíram direta ou indiretamente para o meu doutoramento. Agradeço primeiramente a Deus. Também agradeço aos meus pais, Edimar Vieira Marquezino e Leny Maria de Lima Marquezino, que desde cedo me ensinaram a priorizar a busca pelo conhecimento. Agradeço aos meus orientadores, Renato Portugal e Gonzalo Abal, pelo incentivo, pelos desafios, pela paciência e pela grande amizade que desenvolvemos ao longo dessa caminhada. Sou muito grato também ao Raul Donangelo pela confiança e pela amizade. Sua contribuição foi crucial para o desenvolvimento desta tese. Não poderia deixar de mencionar o grande amigo José Helayel, do CBPF, que contribuiu muito para minha formação como pesquisador. Os amigos do grupo de computação quântica, em especial o Demerson, a Amanda e o Carlos Magno, contribuíram com muitas discussões interessantes. Também sou grato aos meus colegas da pós-graduação, que me ajudaram muito durantes as disciplinas, além de terem me proporcionado um ambiente de trabalho muito alegre e inteligente.

Agradeço ao CNPq, pelo apoio financeiro. Agradeço também ao LNCC e seus funcionários, pela excelente estrutura oferecida e pelo ambiente de trabalho tão agradável. Agradeço à Universidad de la República, de Montevidéu, onde passei algumas semanas desenvolvendo parte do meu trabalho.

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

ANÁLISE, SIMULAÇÕES E APLICAÇÕES ALGORÍTMICAS DE CAMINHADAS QUÂNTICAS

Franklin de Lima Marquezino

Fevereiro , 2010

Orientador: Renato Portugal, D.Sc.

Co-orientador: Gonzalo Abal, D.Sc.

A computação quântica é um modelo computacional baseado nas leis da mecânica quântica, que pode ser utilizado para desenvolver algoritmos mais eficientes que seus correspondentes clássicos. O desenvolvimento de algoritmos quânticos eficientes, no entanto, é uma tarefa altamente desafiadora. Uma abordagem recente que vem se mostrando bem-sucedida é a utilização de caminhadas quânticas. Neste trabalho, estudamos a caminhada quântica no hipercubo, calculando analiticamente sua distribuição estacionária e analisando propriedades de seu *mixing time*, tanto na situação ideal como na situação com descoerência gerada por ligações interrompidas. Também estudamos a caminhada na malha bidimensional, calculando sua distribuição estacionária analiticamente e explorando a relação entre o *mixing time* e a complexidade do algoritmo de busca nesse grafo. Desenvolvemos uma ferramenta computacional para simulação numérica de caminhadas quânticas em malhas uni- e bidimensionais com diversas condições de contorno. Finalmente, estudamos alguns algoritmos de busca em grafos e analisamos numericamente o impacto que a descoerência exerce sobre seus desempenhos.

Abstract of Thesis presented to LNCC/MCT as a partial fulfillment of the requirements for the degree of Doctor of Sciences (D.Sc.)

ANALYSIS, SIMULATIONS AND ALGORITHMIC APPLICATIONS OF QUANTUM WALKS

Franklin de Lima Marquezino

February, 2010

Advisor: Renato Portugal, D.Sc.

Co-advisor: Gonzalo Abal, D.Sc.

Quantum computing is a model of computation based on the laws of quantum mechanics, which can be used to develop faster algorithms. The development of efficient quantum algorithms, however, is a highly challenging task. A recent successful approach is the use of quantum walks. In this work, we have studied the quantum walk on the hypercube, obtaining the exact stationary distribution and analyzing properties of its mixing time both in the ideal and in the noisy set-ups, with noise generated by broken links. We have also studied the walk in a two-dimensional grid, where we have obtained its stationary distribution analytically and have explored the relation between mixing time and the complexity of the search algorithm for this graph. We have developed a computational tool for numerical simulation of quantum walks in one- and two-dimensional grids with several boundary conditions. Finally, we have studied some algorithms for search on graphs and have numerically analyzed the impact of decoherence over their performances.

Sumário

Lista de Figuras

Figura

Lista de Tabelas

Tabela

Lista de Siglas e Abreviaturas

- AKR: algoritmo de Ambainis-Kempe-Rivosh; busca de sítio marcado em malha bidimensional finita com condições de contorno periódicas.
- GNU GPL: *GNU General Public License*, licença para *software* livre.
- k-SAT: problema SAT restrito a cláusulas com no máximo k literais
- NAND: *not-and*; função Booleana binária que retorna "falso" quando ambas as entradas são "verdadeiro" e retorna "verdadeiro" caso contrário.
- NP: *nondeterministic polynomial*; classe das linguagens reconhecidas por uma máquina de Turing não-determinística polinomialmente limitada.
- P: *(deterministic) polynomial*; classe das linguagens reconhecidas por uma máquina de Turing determinística polinomialmente limitada.
- Q-bit: *quantum bit*, ou bit quântico; o mesmo que *qubit*.
- QWalk: Quantum Walk Simulator; simulador de caminhadas quânticas.
- SAT: problema da satisfabilidade.
- SKW: algoritmo de Shenvi-Kempe-Whaley; busca de vértice marcado no hipercubo.

Capítulo 1

Introdução

Costuma-se dizer que 1905 foi o *annus mirabilis* de Albert Einstein, quando este publicou quatro artigos revolucionários nos *Annalen der Physik*. Em um destes artigos ele estuda o movimento Browniano, que havia sido descoberto em 1827 pelo botânico inglês Robert Brown, quando este observava o movimento errático de grãos de pólen na superfície da água. O movimento Browniano pode ser modelado como o caso limite de uma caminhada aleatória.

A caminhada aleatória, também conhecida como passeio aleatório ou *random walk*, é a descrição matemática da trajetória de uma partícula que se move por meio de passos aleatórios sucessivos. Esse conceito já foi utilizado para modelar diversos problemas das mais variadas áreas do conhecimento. Em física é utilizado, por exemplo, no estudo de polímeros, servindo também para modelar o movimento de moléculas em líquidos e gases. Também existem aplicações de caminhadas aleatórias, por exemplo, na psicologia, na economia e na ciência da computação. Por trás de todas essas aplicações estão os processos difusivos em geral.

Um problema muito importante da computação, o k-SAT, possui soluções eficientes conhecidas para $k = 1$ e $k = 2$, porém é NP-completo para $k \geq 3$. Os melhores algoritmos para resolver esse problema para $k = 2$ e $k = 3$ são baseados em caminhadas aleatórias. Dentre os exemplos bem-sucedidos de aplicações computacionais das caminhadas aleatórias podemos ainda citar o problema de conectividade de grafos, da estimativa de volume de corpo convexo e de aproximação

do permanente de uma matriz. Mais detalhes sobre as caminhadas aleatórias podem ser encontrados nos livros de Motwani e Raghavan (1995) e Brémaud (1999).

Alguns trabalhos de Richard Feynman chamaram a atenção da comunidade científica para a possibilidade de aplicar sistemas físicos quânticos no desenvolvimento de um novo modelo computacional, possivelmente mais eficiente que o modelo clássico. Entretanto, antes mesmo de serem construídos os primeiros computadores quânticos com capacidade de processamento suficientemente elevada, é importante que existam algoritmos quânticos mais eficientes que seus correspondentes clássicos. O desenvolvimento desses algoritmos até hoje tem se demonstrado uma tarefa extremamente complexa. Portanto, é importante buscar caminhos mais eficazes para driblar o caráter contra-intuitivo da mecânica quântica e fomentar o surgimento de novos algoritmos quânticos com ganho de complexidade em relação aos clássicos.

Em 1993, pela primeira vez, é utilizado o termo "caminhada aleatória quântica", no artigo de Aharonov et al. (1993). A caminhada quântica é um conceito análogo ao de caminhada aleatória clássica. Há dois tipos de caminhadas quânticas: as contínuas no tempo e as discretas no tempo. O modelo contínuo no tempo foi desenvolvido por Farhi e Gutmann (1998). O modelo discreto no tempo foi desenvolvido inicialmente por Meyer (1996), por meio de seu estudo de autômatos celulares quânticos, e passou a ser visto como possível ferramenta computacional através de Aharonov et al. (2001). No modelo discreto, que será abordado nesta tese, consideramos a descrição do movimento de uma partícula quântica condicionado a um grau de liberdade adicional da própria partícula. Assim como as caminhadas aleatórias já foram empregadas com sucesso no desenvolvimento de diversos algoritmos, as caminhadas quânticas também constituem atualmente uma importante ferramenta para o desenvolvimento de algoritmos quânticos mais rápidos que seus correspondentes clássicos.

Os caminhantes quânticos possuem propriedades muito interessantes. Foi provado que eles se espalham quadraticamente mais rápido que seus correspondentes

tes clássicos (Ambainis et al., 2001) — a variância da posição de um caminhante quântico cresce com $\sigma^2 = O(t^2)$, enquanto a de um caminhante aleatório cresce com $\sigma^2 = O(t)$. De modo ainda mais surpreendente, foi demonstrado por Kempe (2003b) que o tempo de alcance[1] de uma caminhada quântica no hipercubo de dimensão n é exponencialmente menor que o de uma caminhada clássica no mesmo grafo. Certamente, essas e outras propriedades dos caminhantes quânticos já justificariam as pesquisas nessa área, do ponto de vista da física. No entanto, como as caminhadas clássicas têm sido empregadas com tanto êxito no desenvolvimento de algoritmos clássicos, surge naturalmente o questionamento sobre a possibilidade de aproveitar as caminhadas quânticas para o desenvolvimento de novos algoritmos quânticos eficientes.

De fato, muitos algoritmos quânticos já foram desenvolvidos com base em caminhadas quânticas. O algoritmo Shenvi-Kempe-Whaley (SKW) realiza uma busca por um vértice marcado em um hipercubo de dimensão n em tempo $O(\sqrt{2^n})$ (Shenvi et al., 2003b). Classicamente, seria necessário tempo $O(2^n)$. Mais tarde foi desenvolvido o algoritmo Ambainis-Kempe-Rivosh para busca de um vértice marcado em uma malha bidimensional finita, de dimensões $\sqrt{N} \times \sqrt{N}$, com condições de contorno periódicas (Ambainis et al., 2005). Esse algoritmo possui complexidade $O(\sqrt{N} \log N)$, em vez de $O(N)$ como no algoritmo clássico. O algoritmo de Tulsi (2008) resolve o mesmo problema que o algoritmo AKR em tempo $O(\sqrt{N \log N})$. Esses algoritmos podem ser descritos como instâncias do *algoritmo abstrato de busca*.

Alguns autores, usando um formalismo um pouco diferente, também encontraram algoritmos quânticos com ganho de complexidade em relação aos seus correspondentes clássicos. Dentre estes resultados, podemos citar o algoritmo para o problema de unicidade de elementos,[2] desenvolvido por Ambainis (2004), o algoritmo para encontrar subconjuntos, desenvolvido por Childs e Eisenberg (2005)

[1] *Hitting time*. Significa, *grosso modo*, o tempo médio que o caminhante gasta para ir de um vértice a outro de um grafo.

[2] *Element distinctness*. É o problema de determinar se todos os elementos de uma lista são distintos.

e o algoritmo para encontrar triângulos em grafos, desenvolvido por Magniez et al. (2007).

Também foram desenvolvidos algoritmos usando caminhadas quânticas contínuas no tempo. O algoritmo de Childs et al. (2003) atravessa rapidamente um grafo bastante particular — consistindo de duas árvores binárias balanceadas, de altura n, com ligações aleatórias entre suas folhas — e é o primeiro a alcançar ganho de complexidade exponencial em relação ao melhor algoritmo clássico correspondente, utilizando para isso a técnica das caminhadas quânticas[3] . Antes disso, todos os algoritmos quânticos com ganho exponencial de complexidade usavam a transformada de Fourier quântica. Mais tarde, Farhi et al. (2008) desenvolveram um algoritmo baseado na caminhada quântica contínua no tempo e resolveram o problema da árvore NAND em tempo $O(\sqrt{N})$, enquanto o melhor algoritmo clássico conhecido é $O(N^{0.753\cdots})$. Maiores informações sobre o desenvolvimento histórico das caminhadas quânticas podem ser encontradas, por exemplo, nas excelentes revisões de Kempe (2003a) e Kendon (2007), bem como na tese de doutoramento de Oliveira (2007).

1.1 Questionamentos e contribuições

Ao longo deste texto, iremos apresentar nossas respostas a alguns questionamentos que fizemos sobre caminhadas quânticas.

Inicialmente, em nosso trabalho, nos questionamos sobre propriedades da caminhada quântica nas principais topologias, ou seja, em grafos com importantes aplicações algorítmicas e conceituais, como o hipercubo e a malha bidimensional.

Perguntamo-nos primeiramente qual seria a distribuição estacionária da caminhada quântica discreta no tempo, com moeda de Grover, no hipercubo. Sabendo que a distribuição estacionária da caminhada contínua no tempo não é uniforme (Moore e Russell, 2002) e sabendo que a caminhada contínua pode ser obtida a partir da discreta por meio de um processo limite (Strauch, 2006), es-

[3] Em um trabalho anterior, Childs et al. (2002) estudaram um caso parecido. No entanto, o algoritmo resultante não era mais rápido que qualquer algoritmo clássico correspondente.

perávamos que a distribuição estacionária também seria não-uniforme para o caso discreto no tempo. Uma vez tendo respondido essa pergunta, a sequência natural é perguntar-se como cresce o *mixing time* com a dimensão do hipercubo. Também nos perguntamos se haveria, no hipercubo descoerente, uma propriedade semelhante àquela reportada por Kendon e Tregenna (2003) para o ciclo, ou seja, se haveria um parâmetro crítico de descoerência para o qual o *mixing time* fosse mínimo. Estas perguntas foram respondidas em artigo publicado no periódico *Physical Review A* (Marquezino et al., 2008) e no Capítulo 3 desta tese.

Em relação à caminhada quântica na malha bidimensional finita com condições de contorno periódicas, questionamos inicialmente qual seria sua distribuição estacionária. Em seguida, perguntamo-nos se haveria alguma relação entre o *mixing time* da caminhada quântica e a complexidade computacional do algoritmo de busca espacial nesse grafo. Estas perguntas são respondidas no Capítulo 3 desta tese e em artigo em fase final de preparação.

Também nos perguntamos se o desenvolvimento de um simulador numérico para caminhadas quânticas poderia contribuir para o desenvolvimento da área, possibilitando aos pesquisadores de computação quântica estudar propriedades das caminhadas quânticas sem a necessidade de implementar códigos específicos. Nos propusemos a desenvolver um simulador livre e de código aberto, capaz de reproduzir os principais resultados da literatura de forma simples e rápida, além de viabilizar contribuições originais com poucas ou nenhuma modificação no código. Estas perguntas foram respondidas em artigo publicado no periódico *Computer Physics Communications* (Marquezino e Portugal, 2008) e no Capítulo 5 desta tese.

Tendo em vista a importância das aplicações algorítmicas dos caminhantes quânticos, também nos perguntamos qual seria o impacto da descoerência na complexidade de algoritmos de busca em grafos baseados em caminhadas quânticas. Portanto, estudamos numericamente o efeito de três diferentes modelos de ruído nos algoritmos AKR e SKW sem códigos de correção de erros. Estas perguntas

foram respondidas em artigo publicado nos anais do XXIX Congresso da Sociedade Brasileira de Computação (Abal et al., 2009) e no Capítulo 5 desta tese.

1.2 Organização do trabalho

A tese está organizada da seguinte forma. No Capítulo 2 fazemos uma revisão sobre o modelo de caminhadas quânticas em tempo discreto, abordando diferentes topologias. No Capítulo 3 apresentamos alguns resultados originais de nossa pesquisa, referentes a análise de distribuição limite e do *mixing time* das caminhadas no hipercubo e na malha bidimensional finita com condições de contorno periódicas. No Capítulo 4 fazemos uma revisão sobre o algoritmo abstrato de busca, um formalismo que possibilita a definição e análise de algoritmos de busca em grafos baseados em caminhadas quânticas discretas no tempo. No Capítulo 5 apresentamos resultados também originais de nossa pesquisa, relacionados a simulação numérica de caminhadas quânticas e dos algoritmos quânticos de busca descoerentes. No Apêndice A, apresentamos uma revisão de conceitos fundamentais de álgebra linear, mecânica quântica e computação quântica. No Apêncide B, apresentamos uma análise do algoritmo abstrato de busca, tendo como principal referência o artigo de (Tulsi, 2008). No Apêndice C, abordamos algumas questões técnicas do simulador de caminhadas quânticas QWalk.

Capítulo 2

Caminhadas quânticas

Caminhadas quânticas generalizam o conceito de caminhada aleatória clássica. O modelo de caminhada aleatória clássica descreve o movimento de um caminhante condicionado ao resultado de uma variável aleatória, ou "moeda". No caso unidimensional, o caminhante realiza passos iguais à esquerda ou à direita com probabilidades p e $1-p$, respectivamente. Assim, ocupa sítios discretos distribuídos uniformemente sobre a linha, os quais podem ser representados por números inteiros $x = 0, \pm 1, \pm 2, \cdots$. Também é possível descrever uma caminhada aleatória em grafos mais gerais. Se o caminhante desloca-se sobre um grafo regular de grau d, a variável aleatória precisa assumir d valores diferentes, usualmente com a mesma probabilidade $1/d$. As arestas do grafo incidentes sobre um vértice v recebem rótulos de 1 a d. Se o caminhante está no vértice v e o resultado da variável aleatória é j, então o caminhante desloca-se para o vértice v' que se conecta a v pela aresta de rótulo j. Este procedimento é repetido diversas vezes e o resultado é uma caminhada aleatória sobre o grafo. Convém ressaltar que, ao usarmos o modelo de caminhada aleatórias no desenvolvimento de algoritmos, a posição do caminhante pode assumir outras interpretações. No algoritmo de Schöning, por exemplo, a posição é uma atribuição de variáveis que pode ou não satisfazer uma certa fórmula Booleana.

No modelo de caminhada quântica, o caminhante é representado por um vetor normalizado no espaço de Hilbert e seu movimento por um operador unitário.

Pode-se definir uma caminhada quântica em tempo discreto ou em tempo contínuo, sendo que neste trabalho iremos abordar somente o primeiro caso. Para determinar a direção do movimento do caminhante no modelo em tempo discreto é necessário considerar um ou mais graus de liberdade adicionais. Esses graus de liberdade desempenham papel análogo à moeda mencionada anteriormente.

A descoerência é um efeito colateral inevitável em qualquer implementação de um computador quântico, fazendo as características clássicas do sistema físico emergirem e as vantagens da computação quântica, portanto, serem perdidas. Diversos processos físicos, como imperfeições na aplicação de portas lógicas ou interações dos q-bits[1] com o ambiente, por exemplo, podem ser responsáveis por comprometer a evolução coerente da computação. Ao estudar caminhadas quânticas, especialmente quando se tem em mente aplicações algorítmicas das mesmas, é fundamental considerar o impacto de modelos de descoerência. A descoerência em caminhantes quânticos foi considerada previamente, por exemplo, nos trabalhos de Kendon e Tregenna (2003); Romanelli et al. (2005); Alagic e Russell (2005). Há também um importante trabalho de revisão por Kendon (2007).

Na Seção 2.1, descrevemos a caminhada quântica em uma malha unidimensional infinita. Na Seção 2.2, descrevemos uma caminhada com ligações interrompidas entre os vértices. Entre outras aplicações, o conceito de ligações interrompidas pode ser útil na elaboração de modelos físicos de descoerência. Na Seção 2.3, descrevemos a caminhada quântica em malha bidimensional infinita. Na Seção 2.4, discutimos acerca das caminhadas quânticas em malhas finitas com diferentes condições de contorno. Na Seção 2.5, descrevemos a caminhada quântica no hipercubo de dimensão n. Finalmente, na Seção 2.6, discutimos acerca da caminhada quântica na malha hexagonal.

[1] *Quantum bit*, ou bit quântico. Muitos autores, principalmente em língua inglesa, preferem o termo *qubit*. Neste trabalho utilizaremos o termo q-bit, por ser o mais comum nos principais textos da área em língua portuguesa, além de ter uma pronúncia mais natural para leitores lusófonos. Para uma descrição mais detalhada, veja o Apêndice A.

2.1 Caminhada unidimensional

Vamos começar definindo o modelo de caminhada quântica na reta. Seja $\mathcal{H}_P$ o espaço gerado por todas as possíveis posições da partícula. No caso que estamos tratando nesta seção o espaço $\mathcal{H}_P$ possui dimensão infinita, de modo que também podemos denotá-lo $\mathcal{H}_\infty$. A base canônica para este espaço-posição é $\mathcal{B}_P = \{|x\rangle : x \in \mathbb{Z}\}$. Para definir a caminhada quântica em tempo discreto, ainda é necessário introduzir um grau de liberdade adicional, chamado de quiralidade ou de estado moeda. É importante ressaltar que a evolução da moeda na caminhada quântica é determinística, e não aleatória como no caso da caminhada clássica. Este comportamento é consequência do segundo postulado da mecânica quântica, e é uma das principais diferenças entre a caminhada clássica e a quântica. Seja $\mathcal{H}_2$ o espaço gerado pelos possíveis estados da moeda, que irão determinar se o caminhante move-se para a esquerda ou para a direita. Nesse caso, o caminhante pode ser uma partícula quântica de spin 1/2, por exemplo. A base canônica para o espaço-moeda é $\mathcal{B}_C = \{|j\rangle : j \in \{0, 1\}\}$. O espaço de Hilbert considerado na caminhada unidimensional é, portanto, $\mathcal{H}_2 \otimes \mathcal{H}_\infty$.

O estado genérico do caminhante quântico na reta, no instante t, é dado por

$$|\Psi(t)\rangle = \sum_{j=0}^{1} \sum_{x=-\infty}^{\infty} \psi_{j,x}(t) \, |j\rangle \, |x\rangle \, , \tag{2.1}$$

com $\psi_{j,x}(t) \in \mathbb{C}$ e $\sum_j \sum_x |\psi_{j,x}(t)|^2 = 1$. Aqui está implícita a notação compacta para produto tensorial (ou produto de Kronecker), de modo que $|j\rangle \, |x\rangle \equiv |j\rangle \otimes |x\rangle$. Os detalhes encontram-se no Apêndice A.

A fim de respeitar os postulados da mecânica quântica (Apêndice A), a dinâmica do caminhante quântico deve ser totalmente descrita por meio de operadores unitários. O primeiro passo em uma caminhada quântica é uma operação unitária no espaço-moeda, análogo a um "lançamento de moeda" ou à verificação do resultado de uma variável aleatória no modelo clássico. Convém ressaltar, no entanto, que a leitura de uma variável aleatória é um processo irreversível, enquanto a evo-

lução do estado quântico da moeda é um processo reversível. O operador moeda é dado por

$$C = \sum_{j,k=0}^{1} C_{j,k} \left|j\right\rangle \left\langle k\right| . \tag{2.2}$$

O operador moeda pode ser definido livremente, desde que se mantenha unitário. Na reta, porém, utiliza-se usualmente o operador de Hadamard,

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} . \tag{2.3}$$

De fato, foi mostrado por Nayak e Vishwanath (2000) que qualquer caminhada quântica na reta pode ser estudada através do caminhante de Hadamard.

Em seguida, o movimento do caminhante é condicionado ao resultado da moeda. Na reta, podemos descrevê-lo através do operador unitário

$$S = \sum_{j=0}^{1} \sum_{x=-\infty}^{+\infty} \left|j\right\rangle \left\langle j\right| \otimes \left|x + (-1)^j\right\rangle \left\langle x\right| . \tag{2.4}$$

Portanto, o operador de evolução para um passo da caminhada quântica é dado por

$$U = S \circ (C \otimes I_P), \tag{2.5}$$

em que I_P é o operador identidade no subespaço-posição. Para sabermos o estado do caminhante em qualquer instante de tempo, basta que façamos

$$\begin{aligned} \left|\Psi(t+1)\right\rangle =& U \left|\Psi(t)\right\rangle \\ =& U^{t+1} \left|\Psi(0)\right\rangle , \end{aligned} \tag{2.6}$$

em que $\left|\Psi(0)\right\rangle$ é o estado inicial da caminhada. Também podemos aplicar o operador U diretamente no estado da Equação (2.1), a fim de obter a equação de evolução

$$\psi_{j,x}(t+1) = \sum_{k=0}^{1} C_{j,k} \psi_{k,x-(-1)^j}(t). \tag{2.7}$$

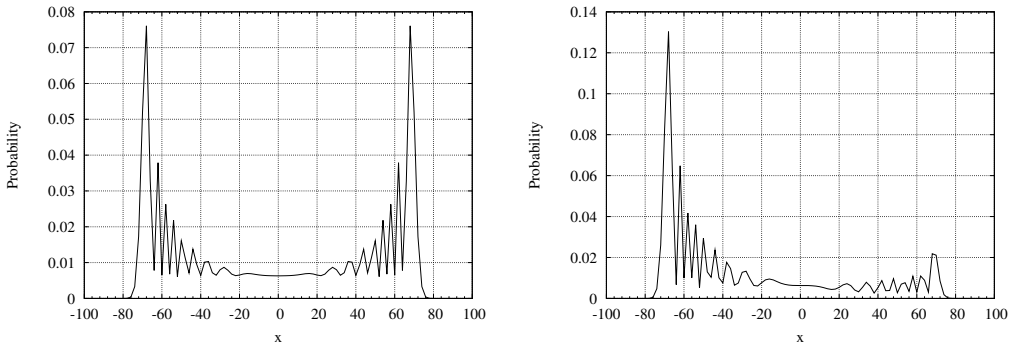

Figura 2.1: Distribuição de probabilidade do caminhante de Hadamard na reta após $t = 100$ passos, com duas condições iniciais diferentes.

Após t passos, caso efetuemos uma medição na base canônica do subespaço-posição, a probabilidade de encontrarmos o caminhante na posição x é dada por

$$P(x,t) = \sum_{j=0}^{1} |\psi_{j,x}(t)|^2, \tag{2.8}$$

de acordo com o quarto postulado da mecânica quântica (Apêndice A). Na Figura 2.1, temos a distribuição de probabilidades do caminhante de Hadamard após $t = 100$ passos, obtida numericamente através do simulador QWalk (Marquezino e Portugal, 2008). É importante ressaltar que a caminhada quântica respeita a paridade dos vértices — partindo de $x = 0$, o caminhante ocupa sítios pares em tempos pares e sítios ímpares em tempos ímpares. Portanto, a fim de facilitar a visualização, os gráficos da Figura 2.1 omitem os zeros dos sítios ímpares. O gráfico à esquerda corresponde ao resultado para a condição inicial

$$|\Psi(0)\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle) \otimes |0\rangle\,, \tag{2.9}$$

enquanto o gráfico à direita representa o resultado para a condição inicial

$$|\Psi(0)\rangle = |1\rangle \otimes |0\rangle\,. \tag{2.10}$$

Discutiremos acerca do simulador QWalk em detalhes no Capítulo 5.

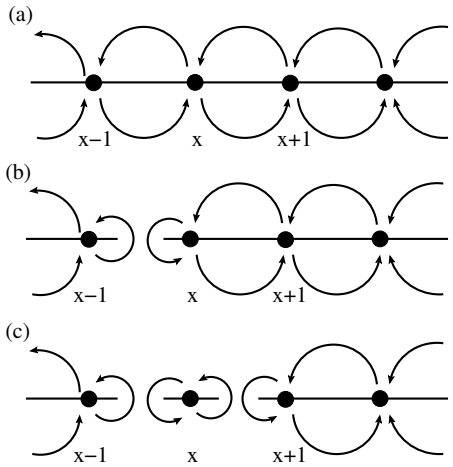

Figura 2.2: Possíveis situações de ligações interrompidas.

2.2 Ligações interrompidas

Podemos considerar a possibilidade de, no instante t, uma ou mais ligações entre vértices da linha estarem interrompidas. A técnica das ligações interrompidas para caminhadas quânticas foi desenvolvida originalmente por Romanelli et al. (2005) e posteriormente generalizada para o caso bidimensional por Oliveira et al. (2006a). As possíveis situações de ligações interrompidas na reta estão representadas na Figura 2.2.

No caso sem ligação interrompida — Figura 2.2(a) — temos a seguinte evolução:

$$\begin{aligned} \psi_{0,x} =& C_{00}\psi_{0,x+1} + C_{01}\psi_{1,x+1} \\ \psi_{1,x} =& C_{10}\psi_{0,x-1} + C_{11}\psi_{1,x-1}, \end{aligned} \tag{2.11}$$

obtida pela Equação (2.7). É importante observar que a evolução unitária implica em conservação do fluxo de probabilidade.

Se a ligação à esquerda do sítio x estiver interrompida, como na Figura 2.2(b), a primeira componente da amplitude ψ em x recebe o fluxo de probabilidade de $x+1$. No entanto, o fluxo de probabilidade saindo da primeira componente de ψ em x é direcionado para a segunda componente de ψ no mesmo sítio, de modo que

passamos a ter a evolução

$$\begin{aligned}\psi_{0,x} =& C_{00}\psi_{0,x+1} + C_{01}\psi_{1,x+1}\\ \psi_{1,x} =& C_{00}\psi_{0,x} + C_{01}\psi_{1,x}.\end{aligned} \tag{2.12}$$

Se a ligação interrompida estiver à direita do sítio x, o processo é análogo.

Se ambas as ligações, à esquerda e à direita do sítio x, estiverem interrompidas, como na Figura 2.2(c), a operação moeda é seguida de uma troca de quiralidade. Assim, passamos a ter a evolução

$$\begin{aligned}\psi_{0,x} =& C_{10}\psi_{0,x} + C_{11}\psi_{1,x}\\ \psi_{1,x} =& C_{00}\psi_{0,x} + C_{01}\psi_{1,x}.\end{aligned} \tag{2.13}$$

A fim de apresentar uma descrição mais sucinta destas observações, vamos agora definir a função

$$\mathcal{L}(j,x) = \begin{cases} (-1)^j, & \text{se ligação para } x+(-1)^j \text{ está fechada.}\\ 0, & \text{caso contrário,}\end{cases} \tag{2.14}$$

para representar todas as possíveis situações de ligações interrompidas em uma dimensão. Pela simetria da reta, temos que $\mathcal{L}(1-j, x+(-1)^j) = 0$ sempre que $\mathcal{L}(j,x) = 0$. Ou seja, se a ligação à direita do vértice x está quebrada, então a ligação à esquerda do vértice $x+1$ também está quebrada.

Tendo definido a função $\mathcal{L}(j,x)$, podemos reescrever o operador deslocamento como

$$S_{bl} = \sum_{j=0}^{1}\sum_{x=-\infty}^{+\infty} |j+\mathcal{L}(j,x)\rangle\langle 1-j| \otimes |x+\mathcal{L}(j,x)\rangle\langle x|, \tag{2.15}$$

de modo a incluir a possibilidade de ligações interrompidas.

Aplicando o operador modificado $U_{bl} = S_{bl}\circ(C\otimes I_P)$ diretamente no estado

da Equação (2.1), obtemos a equação de evolução

$$\psi_{1-j,x}(t+1) = \sum_{k=0}^{1} C_{j+\mathcal{L}(j,x),k}\psi_{k,x+\mathcal{L}(j,x)}(t), \tag{2.16}$$

que descreve a evolução temporal das amplitudes do caminhante quântico. Não é difícil verificar que esta equação abrange todos os casos de evolução com ligações interrompidas descritos anteriormente.

A técnica de ligações interrompidas para caminhadas quânticas pode ser relevante em realizações experimentais de computadores quânticos baseados em cadeias de spin-1/2 de Ising (Oliveira, 2007), além de servir para modelar um tipo de descoerência uniforme, como veremos mais adiante nesse trabalho. Também é possível generalizar o conceito de ligação interrompida para caminhadas em outros grafos, como veremos a seguir, para a malha bidimensional e para o hipercubo.

2.3 Caminhada bidimensional

O espaço de Hilbert considerado na caminhada bidimensional é $\mathcal{H}_2 \otimes \mathcal{H}_2 \otimes \mathcal{H}_\infty$, em que $\mathcal{H}_2 \otimes \mathcal{H}_2$ é o subespaço de Hilbert associado à moeda e $\mathcal{H}_\infty$ é o subespaço de Hilbert associado à posição do caminhante sobre a malha. A base canônica para o subespaço-moeda é $\mathcal{B}_C = \{|j,k\rangle : j,k \in \{0,1\}\}$ e a base canônica para o subespaço-posição é $\mathcal{B}_P = \{|x,y\rangle : x,y \in \mathbb{Z}\}$.

O estado genérico do caminhante quântico na malha bidimensional infinita, no instante t, é dado por

$$|\Psi(t)\rangle = \sum_{j,k=0}^{1} \sum_{x,y=-\infty}^{\infty} \psi_{j,k;x,y}(t) \, |j,k\rangle \, |x,y\rangle \, , \tag{2.17}$$

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

A evolução do sistema ao longo do tempo é dada por um operador unitário $U = S \circ (C \otimes I_P)$, em que S é o operador deslocamento, I_P é o operador identidade

no subespaço-posição e

$$C = \sum_{j,k;j',k'} C_{j,k;j',k'} \left|j,k\right\rangle \left\langle j',k'\right| \tag{2.18}$$

é o operador moeda, que atua no subespaço $\mathcal{H}_2 \otimes \mathcal{H}_2$. O operador moeda pode ser definido livremente, desde que se mantenha unitário. No entanto, existem algumas moedas notáveis, como a moeda de Hadamard, definida por

$$H_4 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}. \tag{2.19}$$

Outras moedas importantes no estudo das caminhadas quânticas bidimensionais são a moeda de Fourier, definida por

$$F_4 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}, \tag{2.20}$$

e a moeda de Grover, definida por

$$G_4 = \frac{1}{2}\begin{pmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix}. \tag{2.21}$$

Há uma certa flexibilidade para a definição do operador S: ele deve ser unitário e deve realizar uma translação condicional no plano. Parte da liberdade para escolher S é a forma de associar os estados de moeda com translações. Apresentaremos aqui alguns operadores de deslocamento para a malha bidimensional infinita.

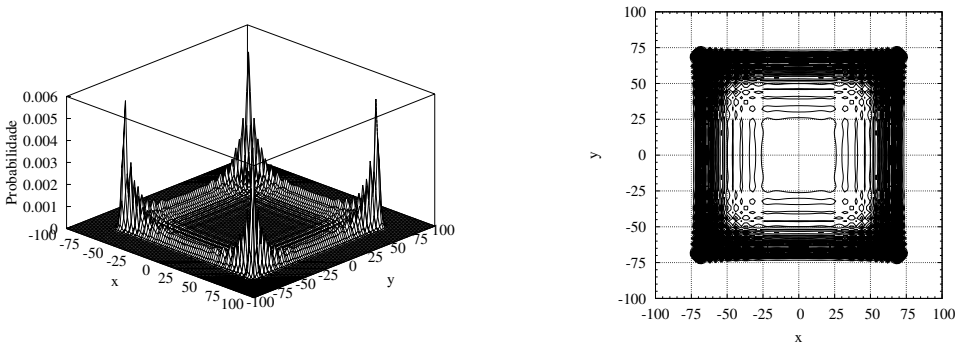

Figura 2.3: Distribuição de probabilidades do caminhante de Hadamard com condição inicial dada pela Equação (2.23), após cem passos. Esquerda: gráfico 3D. Direita: gráfico de contorno.

O primeiro, descrito por Oliveira et al. (2006a), é dado por

$$S_a = \sum_{j,k=0}^{1} \sum_{x,y=-\infty}^{+\infty} |j,k\rangle \langle j,k| \otimes \left|x + (-1)^j, y + (-1)^k\right\rangle \langle x,y| \,. \tag{2.22}$$

Percebe-se que este operador descreve um movimento ao longo das diagonais da malha matemática — o caminhante se desloca ao longo da diagonal principal caso o valor da moeda seja $|00\rangle$ ou $|11\rangle$, e ao longo da diagonal secundária caso o valor da moeda seja $|01\rangle$ ou $|10\rangle$. Nesse caso, dizemos que a malha física é diagonal (em relação à malha matemática).

Para conhecermos a evolução do caminhante quântico precisamos definir o operador de evolução e sua condição inicial. Se considerarmos a moeda de Hadamard com operador de deslocamento dado por S_a e tivermos como condição inicial o estado

$$|\Psi(0)\rangle = \frac{1}{2}\left(|0,0\rangle + i\,|0,1\rangle + i\,|1,0\rangle - |1,1\rangle\right) \otimes |0,0\rangle \,, \tag{2.23}$$

então após cem passos a distribuição de probabilidades do caminhante será aquela da Figura 2.3.

Também podemos considerar a possibilidade de, em um certo instante, uma ou mais ligações do sítio (x, y) para seus vizinhos estarem interrompidas. Neste

caso, precisamos das funções

$$\mathcal{L}_1(j,k;x,y) = \begin{cases} (-1)^j & \text{se ligação para } x+(-1)^j, y+(-1)^k \text{ está fechada,} \\ 0 & \text{caso contrário,} \end{cases} \tag{2.24}$$

e

$$\mathcal{L}_2(j,k;x,y) = \begin{cases} (-1)^k & \text{se ligação para } x+(-1)^j, y+(-1)^k \text{ está fechada,} \\ 0 & \text{caso contrário,} \end{cases} \tag{2.25}$$

com $j,k \in \{0,1\}$. Sempre que $\mathcal{L}_1(1-j,1-k;x+(-1)^j,y+(-1)^k) = 0$ precisamos impor $\mathcal{L}_1(j,k;x,y) = 0$, e analogamente para $\mathcal{L}_2$.

Agora podemos redefinir o operador de deslocamento da Equação (2.22) para levar em consideração a possibilidade de ligações interrompidas,

$$S_{abl} = \sum_{j,k=0}^{1} \sum_{x,y=-\infty}^{+\infty} |j+\mathcal{L}_1(j,k;x,y), k+\mathcal{L}_2(j,k;x,y)\rangle \langle 1-j,1-k| \otimes |x+\mathcal{L}_1(j,k;x,y), y+\mathcal{L}_2(j,k;x,y)\rangle \langle x,y| \,. \tag{2.26}$$

No painel esquerdo da Figura 2.4 temos parte da malha usada na caminhada quântica bidimensional com o operador de deslocamento S_a. A malha matemática está representada por uma linha pontilhada e a malha física por uma linha cheia. No exemplo, temos uma ligação interrompida entre os sítios (x,y) e $(x+1,y+1)$.

Se aplicamos o operador de evolução $U_a = S_{abl} \circ (C \otimes I_P)$ ao estado (2.17), como em (Oliveira et al., 2006a), obtemos a equação de evolução

$$\psi_{1-j,1-k;x,y}(t+1) = \sum_{j',k'=0}^{1} C_{j+\mathcal{L}_1(j,k;x,y),k+\mathcal{L}_2(j,k;x,y);j',k'} \times \psi_{j',k';x+\mathcal{L}_1(j,k;x,y),y+\mathcal{L}_2(j,k;x,y)}(t), \tag{2.27}$$

que descreve a evolução temporal das amplitudes do caminhante quântico.

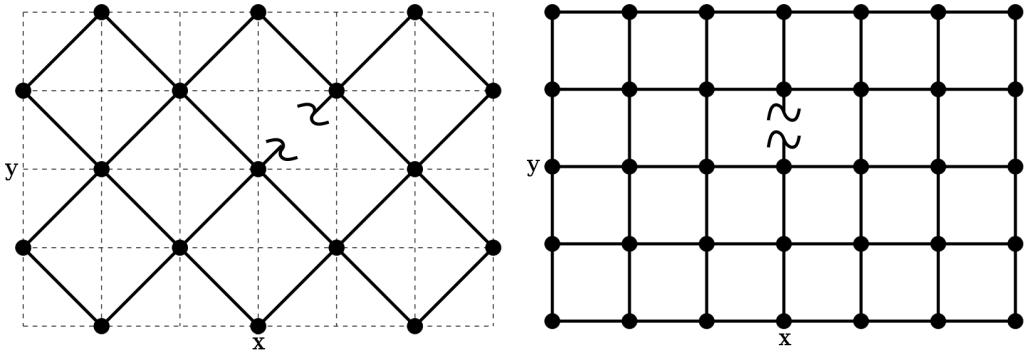

Figura 2.4: Parte da malha para uma caminhada quântica bidimensional, mostrando uma ligação interrompida. Esquerda: Malha diagonal. Direita: Malha natural.

Também podemos definir um segundo operador de deslocamento, a fim de obter uma malha física que coincida com a malha matemática. Marquezino e Portugal (2008) descrevem o operador

$$S_b = \sum_{j,d=0}^{1} \sum_{x,y=-\infty}^{+\infty} |j,d\rangle \langle j,d| \otimes \left|x + (-1)^j(1-\delta_{j,d}), y + (-1)^j \delta_{j,d}\right\rangle \langle x,y| \,. \quad (2.28)$$

Veremos no Capítulo 5, através de exemplos, que a distribuição de probabilidade obtida com o operador S_b difere daquelas obtidas com o operador S_a somente por uma rotação de $\pi/4$.

Se quisermos incluir a possibilidade de ligações interrompidas nesta segunda malha, precisaremos da função

$$\mathcal{L}(j,d;x,y) = \begin{cases} (-1)^j & \text{se ligação para } x + (-1)^j(1-\delta_{j,d}), \\ & y + (-1)^j \delta_{j,d} \text{ está fechada}, \\ 0 & \text{caso contrário}, \end{cases} \quad (2.29)$$

com $j, d \in \{0, 1\}$. Como no caso anterior, precisamos impor $\mathcal{L}(j,d;x,y) = 0$ sempre que $L(1-j, 1-d; x + (-1)^j(1-\delta_{j,d}), y + (-1)^j \delta_{j,d}) = 0$.

Se aplicamos o operador de evolução $U_b = S_b \circ (C \otimes I_P)$ ao estado (2.17) e incluirmos no operador deslocamento a função $\mathcal{L}$, como em (Marquezino e Portugal,

2008), obtemos a equação de evolução

$$\psi_{1-j,1-d;x,y}(t+1) = \sum_{j',d'=0}^{1} C_{j+\mathcal{L}(j,d;x,y),d\oplus\mathcal{L}(j,d;x,y);j',d'} \times \psi_{j',d';x+\mathcal{L}(j,d;x,y)(1-\delta_{j,d}),y+\mathcal{L}(j,d;x,y)\delta_{j,d}}(t), \quad (2.30)$$

em que $\oplus$ significa soma modulo 2. Essa equação descreve a evolução temporal das amplitudes do caminhante quântico.

No painel direito da Figura 2.4 temos parte da malha usada na caminhada quântica bidimensional com o operador de deslocamento S_b. No exemplo, temos uma ligação interrompida entre os sítios (x, y) e $(x, y+1)$.

2.4 Caminhada em malha finita

Até agora estudamos caminhadas quânticas em malhas uni- e bidimensionais infinitas. Há, no entanto, alguns casos relevantes que surgem quando consideramos malhas finitas com condições de contorno periódicas ou reflexivas.

A caminhada no N-ciclo pode ser compreendida como uma caminhada em malha unidimensional com N vértices, indexados pelo conjunto $\{0, 1, \cdots, N-1\}$, e condição de contorno periódica — ou seja, o vértice v_{N-1} está ligado por uma aresta ao vértice v_0. A caminhada se passa em um espaço de Hilbert $\mathcal{H}_2 \otimes \mathcal{H}_P$, em que $\mathcal{H}_2$ é o subespaço-moeda, gerado pela base $\mathcal{B}_C = \{|0\rangle, |1\rangle\}$, e $\mathcal{H}_P$ é o subespaço-posição, gerado pela base $\mathcal{B}_P = \{|0\rangle, |1\rangle, \cdots, |N-1\rangle\}$. O estado genérico do caminhante quântico no N-ciclo, no instante t, é dado por

$$|\Psi(t)\rangle = \sum_{j=0}^{1} \sum_{x=0}^{N-1} \psi_{j,x}(t) |j\rangle |x\rangle, \quad (2.31)$$

com $\psi_{j,x}(t) \in \mathbb{C}$ e $\sum_j \sum_x |\psi_{j,x}(t)|^2 = 1$.

A evolução do sistema ao longo do tempo é dada por um operador unitário $U = S \circ (C \otimes I_P)$, similar ao que já definimos para a caminhada na reta infinita, em que S é o operador deslocamento, I_P é o operador identidade no subespaço-posição

e $C = \sum_{j,k} C_{j,k} |j\rangle \langle k|$ é o operador moeda, que atua no subespaço $\mathcal{H}_2$. O operador de deslocamento pode ser definido como na Equação (2.4)— ou como na Equação (2.15), caso ligações interrompidas sejam consideradas — apenas substituindo a soma na posição por uma soma modulo N.

Também podemos considerar a caminhada em um segmento da reta. Nesse caso, temos uma caminhada quântica em malha unidimensional com N vértices, indexados pelo conjunto $\{0, 1, \cdots, N-1\}$, e condição de contorno reflexiva. O espaço de Hilbert é o mesmo descrito para a caminhada no N-ciclo, assim como o operador de moeda. O operador de deslocamento é definido a partir da Equação (2.15), impondo uma ligação interrompida entre os vértices v_0 e v_{N-1}, ou seja, fazendo $\mathcal{L}(1,0) = 0$ e $\mathcal{L}(0, N-1) = 0$.

A caminhada no toro de N vértices pode ser compreendida como uma caminhada em malha bidimensional com dimensões $\sqrt{N} \times \sqrt{N}$ e condição de contorno periódica. A caminhada se passa em um espaço de Hilbert $\mathcal{H}_2 \otimes \mathcal{H}_2 \otimes \mathcal{H}_P$, em que $\mathcal{H}_2 \otimes \mathcal{H}_2$ é o subespaço-moeda, gerado pela base $\mathcal{B}_C = \{|j,k\rangle : 0 \leq j,k \leq 1\}$, e $\mathcal{H}_P$ é o subespaço-posição, gerado pela base $\mathcal{B}_P = \{|x,y\rangle : 0 \leq x,y \leq \sqrt{N}-1\}$. O estado genérico do caminhante quântico no toro de N vértices, no instante t, é dado por

$$|\Psi(t)\rangle = \sum_{j,k=0}^{1} \sum_{x,y=0}^{\sqrt{N}-1} \psi_{j,k;x,y}(t) |j,k\rangle |x,y\rangle , \tag{2.32}$$

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

A evolução do sistema ao longo do tempo é dada por um operador unitário $U = S \circ (C \otimes I_P)$, similar ao que já definimos para a caminhada malha bidimensional infinita, em que S é o operador deslocamento, I_P é o operador identidade no subespaço-posição e $C = \sum_{j,k;j',k'} C_{j,k;j',k'} |j,k\rangle \langle j',k'|$ é o operador moeda, que atua no subespaço $\mathcal{H}_2 \otimes \mathcal{H}_2$. O operador de deslocamento pode ser definido como na Equação (2.22)— ou como na Equação (2.26), caso ligações interrompidas sejam consideradas — apenas substituindo as somas na posição por somas modulo $\sqrt{N}$.

Também podemos considerar a caminhada em uma caixa de N vértices, ou seja, uma malha bidimensional com dimensões $\sqrt{N} \times \sqrt{N}$ e condição de contorno

reflexiva. O espaço de Hilbert é o mesmo descrito para a caminhada no toro de N vértices, assim como o operador de moeda. O operador de deslocamento é definido a partir da Equação (2.26), impondo ligações interrompidas entre os vértices $v_{0,y}$ e $v_{\sqrt{N}-1,y}$ e entre os vértices $v_{x,0}$ e $v_{x,\sqrt{N}-1}$. Ou seja, as funções $\mathcal{L}_1$ e $\mathcal{L}_2$ devem ser definidas de acordo. Também existem outras variações, como condições de contorno absorventes (Bach et al., 2004; Kempe, 2003a), que não serão abordadas aqui.

2.5 Caminhada no hipercubo

Uma caminhada quântica em tempo discreto no hipercubo (Figura 2.5) de dimensão n se passa em um espaço de Hilbert $H_C \otimes \mathcal{H}_P$, em que H_C é o subespaço-moeda, de dimensão n, e $\mathcal{H}_P$ é o subespaço-posição, de dimensão 2^n. A base canônica para $\mathcal{H}_C$ é o conjunto $\{|j\rangle\}$, para $0 \leq j \leq n-1$, enquanto a base canônica para H_P é o conjunto $\{|x\rangle\}$, com $0 \leq x \leq 2^n - 1$.

O estado genérico do caminhante quântico no hipercubo de dimensão n, no instante t, é dado por

$$|\Psi(t)\rangle = \sum_{j=0}^{n-1} \sum_{x=0}^{2^n-1} \psi_{j,x}(t) \, |j, x\rangle \, , \tag{2.33}$$

com $\psi_{j,x}(t) \in \mathbb{C}$ e $\sum_j \sum_x |\psi_{j,x}(t)|^2 = 1$.

A evolução do sistema ao longo do tempo é dada por um operador unitário $U = S \circ (C \otimes I_P)$, em que $C = \sum_{j,j'} C_{j,j'} \, |j\rangle \, \langle j'|$ é uma operação unitária no subespaço-moeda, I_P é a identidade no subespaço-posição e S é o operador deslocamento, definido como

$$S = \sum_{j=0}^{n-1} \sum_{x=0}^{2^n-1} |j, x \oplus e_j\rangle \, \langle j, x| \, . \tag{2.34}$$

Aqui, $x \oplus e_j$ é a soma binária bit-a-bit entre os vetores binários de n componentes, $x = (x_{n-1}, \cdots, x_1, x_0)$ e e_j, um vetor nulo exceto pela componente indexada por j, a qual vale 1. Usaremos a representação binária ou decimal conforme seja mais

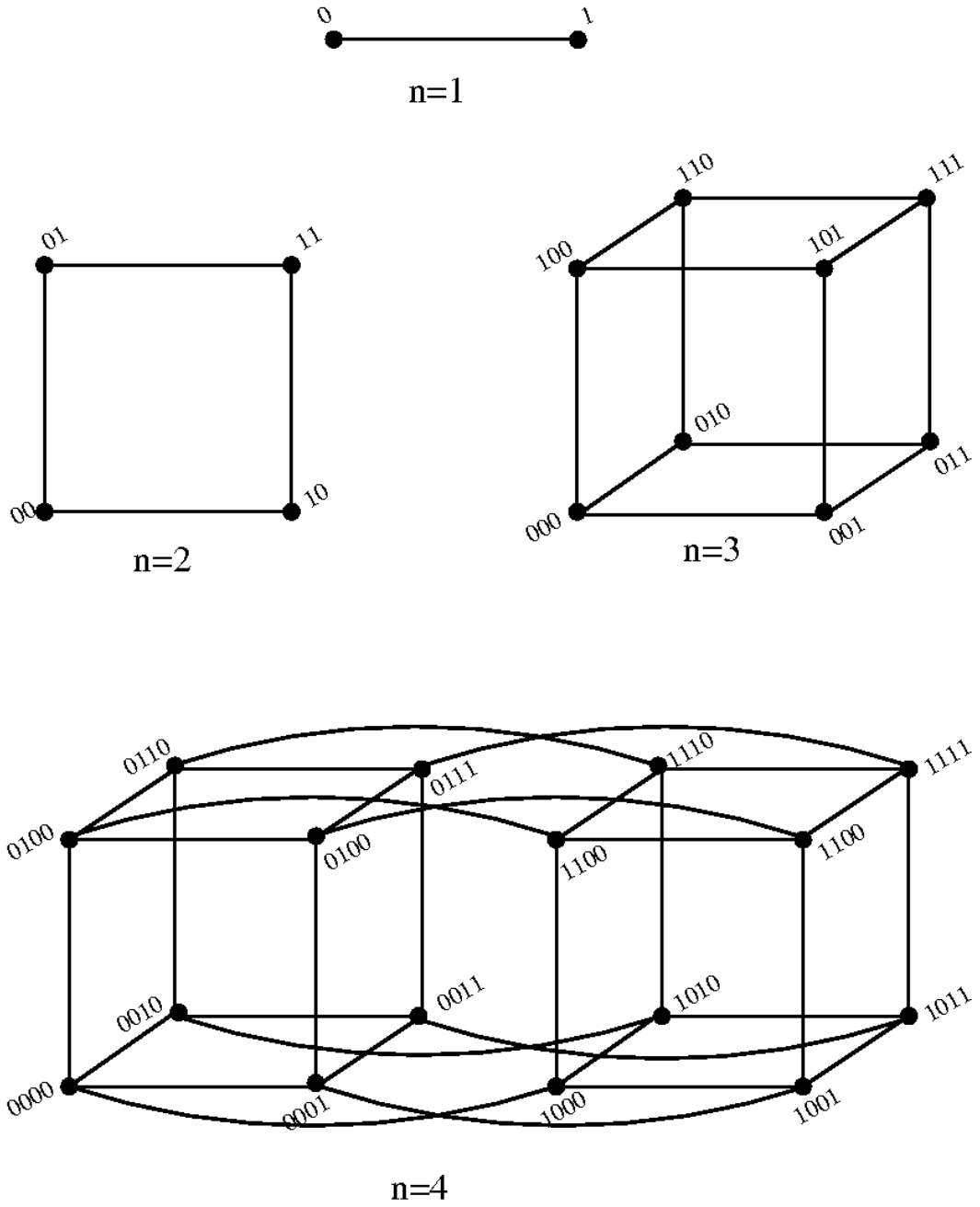

Figura 2.5: Hipercubos de dimensões $n = 1, 2, 3$ e 4, com vértices indexados no sistema binário.

conveniente. Elas poderão ser facilmente reconhecidas pelo contexto, no entanto.

Se aplicamos o operador U ao estado genérico dado pela Equação (2.33), considerando uma moeda também genérica de entradas $C_{j,k}$, obtemos a equação

$$\psi_{j,x}(t+1) = \sum_{k=0}^{n-1} C_{j,k}\psi_{j,x\oplus e_j}(t), \tag{2.35}$$

que descreve a evolução temporal das amplitudes do caminhante quântico.

Também podemos considerar a possibilidade de ligações interrompidas no hipercubo. Para isso, definimos os vetores binários

$$e'_j(x) = \begin{cases} e_j, & \text{se ligação para } x \oplus e_j \text{ está fechada,} \\ 0, & \text{caso contrário.} \end{cases} \tag{2.36}$$

Sempre que $e'_j(x) = 0$, é necessário impor que $e'_j(x \oplus e_j) = 0$. O operador de deslocamento modificado passa a ser

$$S' = \sum_{j=0}^{n-1} \sum_{x=0}^{2^n-1} \left|j, x \oplus e'_j(x)\right\rangle \left\langle j, x\right| . \tag{2.37}$$

Note que, se no sítio x a ligação na direção j está interrompida então

$$|x \oplus e'_j\rangle\langle x| = |x\rangle\langle x \oplus e_j| = |x\rangle\langle x| \tag{2.38}$$

e não há fluxo de probabilidade transferida através das ligações interrompidas. O operador de deslocamento modificado S' é unitário para qualquer número de ligações interrompidas.

Podemos usar o modelo de ligações interrompidas para estudar a caminhada quântica descoerente no hipercubo (ver Seção 3.1.4). Nesse caso, a evolução se dá como segue. A cada passo da evolução, a topologia do hipercubo é definida, abrindo cada ligação com probabilidade p e ajustando os vetores $e'_j(x)$ de acordo com a Equação (2.36). Então, $S' \cdot (I \otimes C)$ é aplicado a $|\Psi(t)\rangle$ para gerar o estado no instante $t+1$. Note que neste modelo o estado da malha no instante $t+1$ não

está correlacionado com o estado anterior, no instante t.

A equação que descreve a evolução temporal das amplitudes de um caminhante quântico no hipercubo com ligações interrompidas é dada por

$$\psi_{j,x}(t+1) = \sum_{k=0}^{n-1} C_{j,k}\psi_{j,x\oplus e'_j(x)}(t), \tag{2.39}$$

em que $\oplus$ denota soma binária bit a bit.

2.6 Caminhada na malha hexagonal

A malha hexagonal recebeu atenção de físicos de matéria condensada por muitos anos e, mais recentemente, o desenvolvimento dos grafenos — arranjos hexagonais bidimensionais de átomos de Carbono — renovou o interesse no assunto. A malha hexagonal com N sítios não é uma lattice própria. Seguindo métodos padrão da física da matéria condensada (Kittel e McEuen, 1996), distinguimos entre os $\frac{N}{2}$ sítios da lattice (brancos) e os $\frac{N}{2}$ sítios da base (pretos), usando um código de cores, como mostrado na Figura 2.6.

Vamos considerar a distância entre dois sítios adjacentes da malha hexagonal como a distância unitária. Então, os vetores $\mathbf{a_1}$ e $\mathbf{a_2}$ que conectam dois sítios vizinhos da lattice (veja a Figura 2.6) possuem norma $\sqrt{3}$ e formam um ângulo de 60^o. O vetor unitário $\mathbf{b}$ que localiza o sítio da base adjacente a um dado sítio da lattice é dado por $\mathbf{b} = \frac{1}{3}(\mathbf{a_1} + \mathbf{a_2})$. Um ponto arbitrário da lattice é endereçado por um vetor com componentes inteiras

$$\mathbf{r} = n_1\mathbf{a_1} + n_2\mathbf{a_2}. \tag{2.40}$$

Cada ponto (n_1, n_2) da lattice possui um ponto associado da base em $\mathbf{r} + \mathbf{b}$.

Por simplicidade, vamos supor que o número de sítios é $N = 2m^2$, com m inteiro, de modo que $\sqrt{\frac{N}{2}}$ também é um inteiro. Assumimos periodicidade em ambas as direções, de modo que os inteiros $n_1, n_2 \in [0, m-1]$. Então, para uma malha de N elementos, temos $N/2$ sítios brancos da lattice, (n_1, n_2), e os $N/2$ sítios

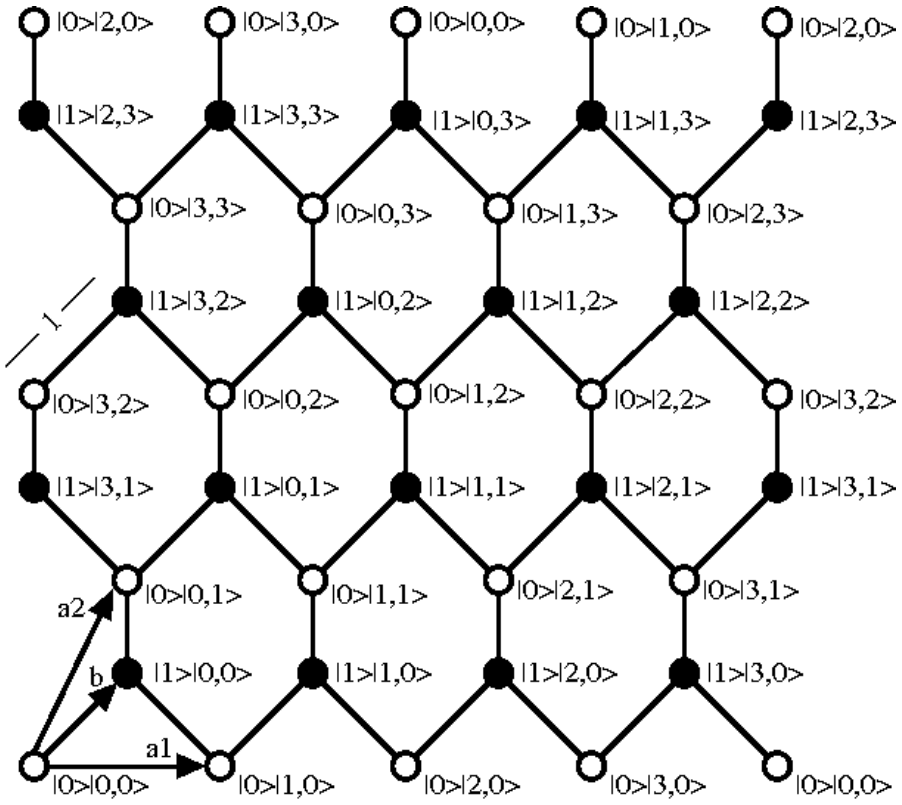

Figura 2.6: Vetores elementares para a malha hexagonal. Os sítios brancos formam uma lattice e os sítios pretos formam a base associada. Aqui temos um exemplo para a malha com $N = 32$ elementos, sendo 16 da lattice e 16 da base. Note a identificação dos elementos do contorno.

correspondentes da base. Na Figura 2.6 temos $N = 32$ e, portanto, $m = \sqrt{\frac{N}{2}} = 4$.

Os kets $|n_1, n_2\rangle$ geram o subespaço-posição associado aos $N/2$ pontos da lattice. A fim de levar em conta também os $N/2$ sítios da base, vamos introduzir um q-bit auxiliar, $\{|0\rangle, |1\rangle\}$, que é $|0\rangle$ para um ponto da lattice e $|1\rangle$ para um ponto da base. Então $|0; n_1, n_2\rangle \equiv |0\rangle \otimes |n_1, n_2\rangle$ indica o estado associado a um ponto da lattice e $|1; n_1, n_2\rangle$, o estado associado ao ponto da base correspondente. O subespaço de dimensão N da lattice, $\mathcal{H}_P$, é gerado pelos kets $\{|s; n_1, n_1\rangle\}$ com $s = 0, 1$ e n_1, n_2 inteiros em $[0, m-1]$. Em cada sítio há três direções possíveis de movimento, as quais indexamos com um inteiro $j = 0, 1, 2$. Eles definem um subespaço-moeda tridimensional, $\mathcal{H}_C$, gerado por $\{|0\rangle, |1\rangle, |2\rangle\}$. O espaço de Hilbert completo, de dimensão $3N$, é $\mathcal{H} = \mathcal{H}_C \otimes \mathcal{H}_P$ e possui uma base genérica $|j, s; n_1, n_2\rangle$ que forma um conjunto ortonormal. Um estado genérico $|\Psi\rangle \in \mathcal{H}$ pode ser expresso como

$$|\Psi\rangle = \sum_{j,\, n_1, n_2} a_{j,n_1,n_2} |j, 0, n_1, n_2\rangle + b_{j,n_1,n_2} |j, 1, n_1, n_2\rangle \tag{2.41}$$

em que a_{j,n_1,n_2} (b_{j,n_1,n_2}) são os componentes da lattice (base) e a condição de normalização $\langle\Psi|\Psi\rangle = 1$ é assumida. Um passo em qualquer direção a partir de um ponto da lattice (base) leva a um ponto da basis (lattice), de acordo com a regra de propagação

$$|j, s, n_1, n_2\rangle \rightarrow |j, s \oplus 1; n_1 - \alpha_j, n_2 - \beta_j\rangle \tag{2.42}$$

em que $\oplus$ é a soma binária e $\hat{\mathbf{v}}_j = (\alpha_j, \beta_j)$ são os vetores bidimensionais

$$\hat{\mathbf{v}}_0 = (0, 0), \quad \hat{\mathbf{v}}_1 = (1, 0), \quad \text{e} \quad \hat{\mathbf{v}}_2 = (0, 1). \tag{2.43}$$

Esses deslocamentos em $\mathcal{H}$ são implementados com o operador de deslocamento,

$$S = \sum_{j,\, s,\, \hat{\mathbf{r}}} |j, 1 - s, \hat{\mathbf{r}} - (-1)^s \mathbf{v}_j\rangle \langle j, s, \hat{\mathbf{r}}| \tag{2.44}$$

no qual introduzimos a notação abreviada $\hat{\mathbf{r}}$ para (n_1, n_2) e subentendemos a soma modulo m para estas componentes.

O operador de evolução de uma caminhada quântica na malha hexagonal é

$$U = S \cdot (G_3 \otimes I_P), \tag{2.45}$$

em que I_P é a identidade em $\mathcal{H}_P$. A operação de Grover tridimensional G_3 atua em $\mathcal{H}_C$ e, na representação acima, é dada por

$$G_3 = \frac{1}{3}\begin{pmatrix} -1 & 2 & 2 \\ 2 & -1 & 2 \\ 2 & 2 & -1 \end{pmatrix}. \tag{2.46}$$

Após t iterações, um estado inicial $|\Psi(0)\rangle$ evolui para $|\Psi(t)\rangle = U^t |\Psi(0)\rangle$. Note que U é um operador real, como exigido pelo formalismo do algoritmo abstrato de busca.

Até aqui, mostramos como é possível definir a caminhada quântica em diversas topologias. A seguir, discutiremos acerca das distribuições limite e do *mixing time* de caminhadas quânticas. Também apresentaremos alguns resultados originais de nossa pesquisa.

Capítulo 3

Distribuições limite e *mixing time*

No estudo das caminhadas aleatórias, o conceito de *mixing time* desempenha um papel fundamental. *Grosso modo*, trata-se do tempo médio gasto para a distribuição de probabilidades induzida por um caminhante aleatório aproximar-se de uma distância ϵ da distribuição estacionária. É natural perguntar, portanto, se a caminhada quântica possui *mixing time* e se este é inferior ao clássico.

Seja $P(x,t)$ a probabilidade de encontrar o caminhante quântico no vértice x de um grafo qualquer, no instante t. Esta probabilidade depende da condição inicial. Como a distribuição de probabilidades da caminhada quântica resulta de um processo unitário, ela não pode convergir para uma distribuição estacionária. De fato, como matrizes unitárias preservam a norma de vetores, a distância entre os vetores que descrevem passos subsequentes da caminhada não converge para zero (Aharonov et al., 2001). Foi demonstrado, porém, que a distribuição média, definida no caso discreto no tempo como

$$\bar{P}(x,T) = \frac{1}{T}\sum_{t=0}^{T-1} P(x,t), \tag{3.1}$$

de fato converge para uma distribuição limite (Aharonov et al., 2001). Também denotaremos $\bar{P}_T(x) \equiv \bar{P}(x,T)$ onde for conveniente. Essa distribuição é equivalente a escolher um instante t uniformemente em $[0, T-1]$, deixar o caminhante quântico evoluir por t passos, e depois medir a posição do caminhante. Definimos, portanto,

a distribuição estacionária como

$$\pi(x) \equiv \lim_{T\to\infty} \bar{P}(x,T). \tag{3.2}$$

Formalmente, podemos definir o *mixing time* quântico de dois modos diferentes. Ambas as definições dependem da condição inicial do caminhante quântico.

Definição 3.1 (Mixing time)**.** *O mixing time médio M_ϵ de um caminhante quântico em relação a uma distribuição de referência π é*

$$M_\epsilon = \min\{T \,|\, \forall t \geq T, \left\|\bar{P}_t - \pi\right\| \leq \epsilon\}, \tag{3.3}$$

em que $\|A - B\| \equiv \sum_x |A(x) - B(x)|$ é a variação total da distância entre duas distribuições.

Esta definição (Aharonov et al., 2001) captura a taxa com a qual a distribuição de probabilidades média se aproxima da distribuição assintótica de um caminhante quântico. Uma definição alternativa para o *mixing time*, apresentada por Moore e Russell (2002), captura o primeiro instante no qual a caminhada está perto por uma distância ϵ de uma distribuição de referência π,

Definição 3.2 (Mixing time instantâneo)**.** *O mixing time instantâneo I_ϵ de um caminhante quântico é*

$$I_\epsilon = \min\{t \,|\, \|P_t - \pi\| \leq \epsilon\}. \tag{3.4}$$

Um resultado importante para o cálculo do *mixing time* foi dado por Aharonov et al. (2001), o qual enunciaremos a seguir.

Teorema 3.1. *Considere um caminhante quântico qualquer em um grafo G com n nós e com um espaço de moeda de dimensão d. Então, para um dado estado inicial $|\Psi_0\rangle$, a distância entre a distribuição de probabilidades média e a distribuição estacionária satisfaz*

$$\|\bar{P}(x,T) - \pi(x)\| \leq \frac{\pi}{T\Delta}\left(\ln\frac{nd}{2} + 1\right), \tag{3.5}$$

em que Δ é definido como a menor distância entre dois autovalores distintos do operador de evolução do caminhante.

Neste capítulo estaremos interessados em calcular a distribuição estacionária de caminhantes quânticos em diferentes topologias e em seguida estudar propriedades de seu *mixing time*. Na Seção 3.1, discutimos acerca da caminhada quântica no hipercubo. Na Seção 3.2, analisamos a caminhada quântica na malha bidimensional com condições de contorno periódicas. Na Seção 3.3, fazemos uma breve discussão sobre os resultados aqui apresentados.

3.1 Caminhada quântica no hipercubo

O *mixing time* de uma cadeia de Markov no hipercubo de dimensão n está em $O(n \log n)$. No artigo de Aharonov et al. (2001) encontramos um limite assintótico superior para o *mixing time* de caminhadas quânticas em grafos genéricos. Particularizando para o hipercubo, esse limite é $O\left(\frac{n^{1.5}}{\epsilon}\right)$. Nossas simulações numéricas indicam um *mixing time* em $O\left(\frac{n}{\epsilon}\right)$, ou seja, melhor que o *mixing time* clássico (Marquezino et al., 2008). Moore e Russell (2002) consideraram *mixing times* com relação à distribuição uniforme, tanto no modelo discreto no tempo como no modelo contínuo no tempo. No caso contínuo no tempo eles encontraram que a distribuição limite média não é uniforme. Também concluíram que sempre se pode encontrar $\epsilon > 0$ tal que não existe *mixing time* para a distribuição uniforme. Como a caminhada quântica discreta no tempo pode ser obtida a partir do modelo contínuo por meio de um processo de limite adequado (Strauch, 2006), esperaríamos que a distribuição limite no caso discreto no tempo fosse também não-uniforme. De fato, é importante ressaltar que a distribuição uniforme *não coincide* com a estacionária, conforme demonstramos em nosso trabalho (Marquezino et al., 2008).

Nesta seção, mostramos que a distribuição assintótica de um caminhante quântico discreto no tempo sobre o hipercubo não é uniforme. Obtivemos a expressão explícita para essa distribuição e caracterizamos o *mixing time* médio para

esta distribuição (Marquezino et al., 2008). O *mixing time* instantâneo também é um conceito bastante útil, que captura o primeiro instante em que a distribuição da posição está próxima a uma distância ϵ de uma distribuição de referência. Moore e Russell (2002) mostraram que, para ambos os modelos de caminhada quântica no hipercubo, o *mixing time* instantâneo para a distribuição uniforme depende linearmente da dimensão n. Isso representa uma melhoria em relação à caminhada correspondente clássica, de $O(n \log n)$ para $O(n)$. Em nossas simulações numéricas, confirmamos esse resultado para o caso discreto no tempo e mostramos que existe $\epsilon > 0$ tal que o *mixing time* instantâneo para a distribuição estacionária (não-uniforme) não existe (Marquezino et al., 2008).

Todas essas propriedades são afetadas pela descoerência. O impacto da descoerência sobre caminhadas quânticas foi estudada principalmente em sistemas uni- ou bidimensionais usando medições repetidas (Brun et al., 2003; Kendon e Tregenna, 2003), ou ruído topológico de ligações interrompidas (Romanelli et al., 2005; Abal et al., 2007; Oliveira et al., 2006a). Kendon e Tregenna (2003) também apresentaram resultados numéricos mostrando o impacto de medições repetidas no *hitting time* de um caminhante quântico discreto no tempo no hipercubo. Chamamos de *hitting time* o tempo médio gasto pelo caminhante quântico para sair de um vértice inicial x_0 até atingir um vértice final x_f. Na presença de descoerência, espera-se que a distribuição média da posição de um caminhante quântico em um hipercubo de dimensão n convirja para a distribuição uniforme, como no caso clássico. Recentemente, o efeito de medições repetidas em um caminhante quântico no hipercubo foi estudado por Alagic e Russell (2005) usando técnicas de super-operadores para o caso contínuo no tempo. Eles concluíram que para taxas de descoerência pequenas, tanto o *hitting time* como o *mixing time* — em relação à distribuição uniforme — permanecem lineares em n. Porém, se a taxa de medições for superior a um certo limiar, o resultado clássico é reobtido, com a distribuição média convergindo para a uniforme em tempo $O(n \log n)$. Nesta seção, consideramos o efeito que a abertura aleatória de ligações do hipercubo exerce na

distribuição de probabilidades da caminhada quântica discreta no tempo (Marquezino et al., 2008). Esse mecanismo não envolve medições, sendo um exemplo de ruído unitário (Shapira et al., 2003).

Em geral, a descoerência é considerada um obstáculo ao desenvolvimento da computação quântica, por destruir o emaranhamento e as superposições necessárias para o processamento eficiente da informação no computador quântico. No entanto, existem resultados bastante surpreendentes que mostram como uma quantidade controlada de descoerência pode ser útil para obtenção de certas distribuições de probabilidades. Kendon e Tregenna (2003) mostraram que uma pequena taxa de descoerência pode ser usada para gerar distribuições quase uniformes na caminhada quântica na linha. Maloyer e Kendon (2007) mostraram que o *mixing time* de uma caminhada quântica discreta no tempo em um ciclo de N elementos pode ser reduzido permitindo um certo nível de descoerência de medições repetidas, desde que essas medições afetem a posição do caminhante quântico. Em nosso trabalho, encontramos um efeito similar no caso do hipercubo com descoerência de ligações interrompidas (Marquezino et al., 2008). Existe um taxa de descoerência crítica, para a qual o *mixing time* possui um mínimo. Esse resultado mostra um novo caso para o qual a descoerência, gerada através de aleatoriedade topológica e sem medições, melhora uma propriedade útil da mecânica quântica. Nesse caso, possibilitando *mixing times* rápidos.

3.1.1 A caminhada coerente

A caminhada quântica no hipercubo já foi descrita na Seção 2.5. Para calcularmos a distribuição limite dessa caminhada, precisamos antes resolver o problema de autovalor e autovetor de seu operador de evolução. Esse problema torna-se mais simples se considerarmos a caminhada coerente no espaço de Fourier.

Como o hipercubo é um grafo de Cayley do grupo $\mathbb{Z}_2^n$, a transformação mais

adequada é a transformada de Fourier nesse grupo, gerada pela base de 2^n vetores

$$|k\rangle \equiv \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{k\cdot x} |x\rangle , \tag{3.6}$$

para $k \in [0, 2^n - 1]$, em que $k \cdot x \equiv \sum_{j=0}^{n-1} x_j k_j$ é o produto bit a bit. As amplitudes transformadas são

$$\tilde{\psi}_{j,k} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{k\cdot x} \psi_{j,x}. \tag{3.7}$$

O operador de deslocamento é diagonal no espaço de Fourier. Para verificar essa propriedade, fazemos

$$\begin{aligned} S|j,k\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{k\cdot x} |j, x \oplus e_j\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{k\cdot (x\oplus e_j)} |j,x\rangle \\ &= (-1)^{k_j} |j,k\rangle . \end{aligned} \tag{3.8}$$

Desse modo, o operador S_k, que atua em $|\Psi_k\rangle \equiv \langle k|\Psi\rangle = \sum_{j=0}^{n-1} \tilde{\psi}_{j,k} |j\rangle$, possui elementos matriciais dados por

$$\begin{aligned} S_k(i,j) &= \langle i,k|S|j,k\rangle \\ &= (-1)^{k_j} \langle i,k|j,k\rangle \\ &= (-1)^{k_j} \delta_{i,j}. \end{aligned} \tag{3.9}$$

Portanto, o operador U_k, que atua em $|\Psi_k\rangle$, tem elementos matriciais dados por $U_k(i,j) = (-1)^{k_i} C_{i,j}$. Os autovetores de U são o produto tensorial dos autovetores de U_k e $|k\rangle$.

Daqui em diante, iremos considerar o caso particular da moeda de Grover n-dimensional, dada por $C_{j,k} \equiv \frac{2}{n} - \delta_{j,k}$. Esta moeda obedece as simetrias por permutações observadas no hipercubo de dimensão n, além de ser o operador deste tipo mais distante da identidade (Moore e Russell, 2002). Passaremos a descrever o

problema de autovalores de U_k. Seus autovalores dependem somente da dimensão n e do peso de Hamming de k, definido como $|k| = \sum_{j=0}^{n-1} k_j$. Entretanto, seus autovetores, $|\nu_j(k)\rangle$ para $j = 1 \cdots n$, dependem de k.

Para pesos de Hamming $|k| = 0$ (ou $|k| = n$), um conjunto de $n-1$ autovetores degenerados com autovalor $\lambda = 1$ (ou $\lambda = -1$) é dado por $|\nu_i(0)\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |i\rangle)$, para $i \in [1, n-1]$. O autovetor restante, com autovalor $\lambda = -1$ (ou $\lambda = 1$) é a superposição uniforme $|\nu_n(0)\rangle = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} |j\rangle$.

No caso em que o peso de Hamming toma valores $0 < |k| < n$, há $n-|k|-1$ autovalores degenerados com autovalor -1 e $|k|-1$ autovetores degenerados com autovalor 1. Indexando as entradas do vetor k segundo a notação $k = (k_0, k_1, \cdots, k_{n-1})$, seja $\mathcal{I}_0 = \{j; k_j = 0\}$ o conjunto de índices para entradas nulas do vetor binário k, e seja $\mathcal{I}_1 = \{j; k_j = 1\}$ o conjunto de índices para entradas não-nulas de k. Seja $m_{\mathcal{I}_0} = \min(\mathcal{I}_0)$ e $m_{\mathcal{I}_1} = \min(\mathcal{I}_1)$. Os autovetores com autovalor -1 são dados por $|\nu_j(k)\rangle = \frac{1}{\sqrt{2}}(|m_{\mathcal{I}_0}\rangle - |j\rangle)$, para $j \in \mathcal{I}_0/\{m_{\mathcal{I}_0}\}$. Os autovetores com autovalor 1 são dados por $|\nu_j(k)\rangle = \frac{1}{\sqrt{2}}(|m_{\mathcal{I}_1}\rangle - |j\rangle)$, para $j \in \mathcal{I}_1/\{m_{\mathcal{I}_1}\}$.

Os dois autovalores restantes são os mais relevantes para nossa análise. Eles podem ser expressos como $e^{\pm i\omega_k}$, em que

$$\cos\omega_k \equiv 1 - \frac{2|k|}{n}. \tag{3.10}$$

Os autovetores conjugados correspondentes são $|\nu_n(k)\rangle = \sum_{j=0}^{n-1} \alpha_j(k)\,|j\rangle$ e $|\nu_{n-1}(k)\rangle = \sum_{j=0}^{n-1} \alpha_j^*(k)\,|j\rangle$, com componentes $\alpha_j(k)$ dadas por

$$\alpha_j(k) = \frac{1}{\sqrt{2}} \left(\frac{k_j}{\sqrt{|k|}} - i\frac{1-k_j}{\sqrt{n-|k|}} \right). \tag{3.11}$$

Na última equação, $i = \sqrt{-1}$.

Este conjunto de autovetores normalizados formam uma base não-ortogonal, e estão resumidos na Tabela 3.1.

Tabela 3.1: Autovalores e autovetores de U_k. As quantidades ω_k e $\alpha_j(k)$ são definidas nas Equações (3.10) e (3.11), respectivamente.

Autovalor	**Autovetor $\|\nu_i(k)\rangle$**	**índice i**	**multiplicidade**
Peso de Hamming $\|k\| = 0$			
-1	$\frac{1}{\sqrt{2}}(\|0\rangle - \|i\rangle)$	$i \in [1, n-1]$	$n-1$
1	$\frac{1}{\sqrt{n}}\sum_{j=0}^{n-1}\|j\rangle$	n	1
Pesos de Hamming $1 \leq \|k\| \leq n-1$			
-1	$\frac{1}{\sqrt{2}}(\|m_{\mathcal{I}_0}\rangle - \|i\rangle)$	$i \in \mathcal{I}_0/\{m_{\mathcal{I}_0}\}$	$n - \|k\| - 1$
1	$\frac{1}{\sqrt{2}}(\|m_{\mathcal{I}_1}\rangle - \|i\rangle)$	$i \in \mathcal{I}_1/\{m_{\mathcal{I}_1}\}$	$\|k\| - 1$
$e^{i\omega_k}$	$\sum_{j=0}^{n-1}\alpha_j(k)\|j\rangle$	n	1
$e^{-i\omega_k}$	$\sum_{j=0}^{n-1}\alpha_j^*(k)\|j\rangle$	$n-1$	1
Peso de Hamming $\|k\| = n$			
1	$(\|0\rangle - \|i\rangle)/\sqrt{2}$	$i \in [1, n-1]$	$n-1$
-1	$\sum_{j=0}^{n-1}\|j\rangle/\sqrt{n}$	n	1

3.1.2 Distribuição limite

Passamos a considerar o problema de determinar a distribuição limite para uma caminhada quântica no hipercubo de dimensão n. O vetor de estado inicial está localizado no vértice $x = 0$ e uniformemente distribuído no subespaço da moeda,

$$|\Psi(0)\rangle = \frac{1}{\sqrt{n}}\sum_{j=1}^{n}|j\rangle \otimes |x=0\rangle\,. \tag{3.12}$$

Esta escolha respeita a simetria de permutação do hipercubo. O estado inicial é expresso em termos de autovetores de U_k,

$$|\Psi(0)\rangle = \frac{1}{\sqrt{n2^n}}\sum_{k=0}^{2^n-1}\sum_{i=1}^{n}a_i(k)\;|\nu_i(k)\rangle \otimes |k\rangle\,, \tag{3.13}$$

em que os coeficientes são somatórios dos componentes dos autovetores, $a_i(k) \equiv \frac{1}{\sqrt{n2^n}}\sum_{j=0}^{n-1}\langle\nu_i(k)|j\rangle$. Esses somatórios são iguais a zero exceto quando $i = n-1$ ou $i = n$, de modo que somente esses dois autovetores contribuem. Assim,

$$|\Psi(0)\rangle = \sum_{k=0}^{2^n-1}\left(a_{n-1}(k)\,|\nu_{n-1}(k)\rangle + a_n(k)\,|\nu_n(k)\rangle\right) \otimes |k\rangle\,. \tag{3.14}$$

Os coeficientes relevantes são $a_n(k) = (\sqrt{|k|} + i\sqrt{n-|k|})/\sqrt{n\,2^{n+1}}$ e $a_{n-1}(k) = a_n^*(k)$, em que $|k| \in [1, n-1]$. Se $|k| = 0$ ou $|k| = n$, então $a_n = 1/\sqrt{2^{n+1}}$ e $a_{n-1} = 0$. O estado do caminhante no instante t é dado por

$$|\Psi(t)\rangle = \sum_{k=0}^{2^n-1} \left(a_{n-1}(k)\, e^{-i\omega_k t}\, |\nu_{n-1}(k)\rangle + a_n(k)\, e^{i\omega_k t}\, |\nu_n(k)\rangle \right) \otimes |k\rangle\,. \tag{3.15}$$

A probabilidade de encontrar o caminhante no instante t no vértice x é dada por $P(x,t) = \sum_{j=0}^{n-1} |\langle j, x|\Psi(t)\rangle|^2$. Esta quantidade pode ser calculada no espaço de Fourier, usando a identidade da Equação (3.6) para obter

$$\langle j, x|\Psi(t)\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} (-1)^{k\cdot x} \langle j, k|\Psi(t)\rangle. \tag{3.16}$$

Desse modo, temos

$$P(x,t) = \frac{1}{2^n} \sum_{k,k'=0}^{2^n-1} (-1)^{(k\oplus k')\cdot x} \sum_{j=0}^{n-1} \langle j, k|\Psi(t)\rangle\langle\Psi(t)|k', j\rangle \tag{3.17}$$

Portanto, eliminando do somatório os termos nulos, temos

$$\begin{aligned} P(x,t) \;=\; & \frac{1}{2^n} \sum_{k,k'=0}^{2^n-1} (-1)^{(k\oplus k')\cdot x} \big\{ a_{n-1}(k) a_{n-1}^*(k') \langle \nu_n(k)|\nu_n(k')\rangle e^{-i(\omega_k - \omega_{k'})t} + \\ & a_n(k) a_n^*(k') \langle \nu_{n-1}(k)|\nu_{n-1}(k')\rangle e^{i(\omega_k - \omega_{k'})t} + \\ & a_n(k) a_{n-1}^*(k') \langle \nu_{n-1}(k)|\nu_n(k')\rangle e^{i(\omega_k + \omega_{k'})t} + \\ & a_{n-1}(k) a_n^*(k') \langle \nu_n(k)|\nu_{n-1}(k')\rangle e^{-i(\omega_k + \omega_{k'})t} \big\}\,. \end{aligned} \tag{3.18}$$

Nosso objetivo é calcular a distribuição de probabilidade assintótica $\pi(x)$, definida pela Equação (3.2). Os dois primeiros termos contribuem somente se $|k| = |k'|$, pois $\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} e^{\pm i(\omega_k - \omega_{k'})t} = \delta_{|k|,|k'|}$ e não há contribuição dos dois últimos termos devido a $\lim_{T\to\infty} \sum_{t=0}^{T-1} e^{\pm i(\omega_k + \omega_{k'})t} = 0$. Portanto, a distribuição assintótica

pode ser calculada como

$$\pi(x) = \frac{1}{2^n} \sum_{k,k'=0}^{2^n-1} \delta_{|k|,|k'|} (-1)^{(k\oplus k')\cdot x} \left\{ |a_{n-1}(k)|^2 \langle \nu_n(k)|\nu_n(k')\rangle + \right.$$
$$\left. |a_n(k)|^2 \langle \nu_{n-1}(k)|\nu_{n-1}(k')\rangle \right\}. \quad (3.19)$$

O produto interno entre os autovetores não-triviais é dado por

$$\begin{aligned} \langle \nu_n(k)|\nu_n(k')\rangle &= \langle \nu_{n-1}(k)|\nu_{n-1}(k')\rangle \\ &= \frac{n\,(k\cdot k') + |k|(n-2|k|)}{2|k|(n-|k|)}. \end{aligned} \quad (3.20)$$

Após usar $|a_n(k)|^2 = |a_{n-1}(k)|^2 = 1/2^{n+1}$ obtemos a distribuição estacionária

$$\pi(x) \;=\; \frac{2}{2^{2n}} \;+\; \frac{1}{2^{2n}} \sum_{\substack{k,k'=0 \\ (|k|=|k'|\neq 0,n)}}^{2^n-1} (-1)^{(k\oplus k')\cdot x} \left[\frac{n\,(k\cdot k') + |k|(n-2|k|)}{2|k|(n-|k|)} \right]. \quad (3.21)$$

A expressão acima não é eficiente para calcular $\pi(x)$. É possível simplificá-la notando que $\pi(x)$ depende somente do peso de Hamming de x, ou seja, $|x| = \sum_{j=0}^{n-1} x_j$. Após alguma álgebra, encontramos

$$\begin{aligned} \pi(x) = \frac{2}{2^{2\,n}} + \frac{1}{2^{2n}} \sum_{i=1}^{n-1} \sum_{j=0}^{i} \sum_{l=0}^{|x|} \sum_{m=0}^{|x|} \sum_{p=0}^{m} (-1)^l \times \\ \binom{|x|-m}{l-m+p} \binom{m}{p} \binom{n-|x|-i+m}{i-j-l+m-p} \times \\ \binom{i-m}{j-p} \binom{n-|x|}{i-m} \binom{|x|}{m} \frac{i\,(n-2\,i)+nj}{2i\,(n-i)}, \end{aligned} \quad (3.22)$$

em que os coeficientes combinatoriais são $\binom{n}{m} = \frac{n!}{(n-m)!\,m!}$ para $n \geq m \geq 0$ e $\binom{n}{m} = 0$ caso contrário. Esta expressão é equivalente à Equação (3.21) e, para alguns valores de x, ela fornece resultados bem simples, tais como

$$\pi\,(0) = \frac{1}{4^n} + \frac{\Gamma\left(n+\frac{1}{2}\right)}{2\sqrt{\pi} n \Gamma(n)}. \quad (3.23)$$

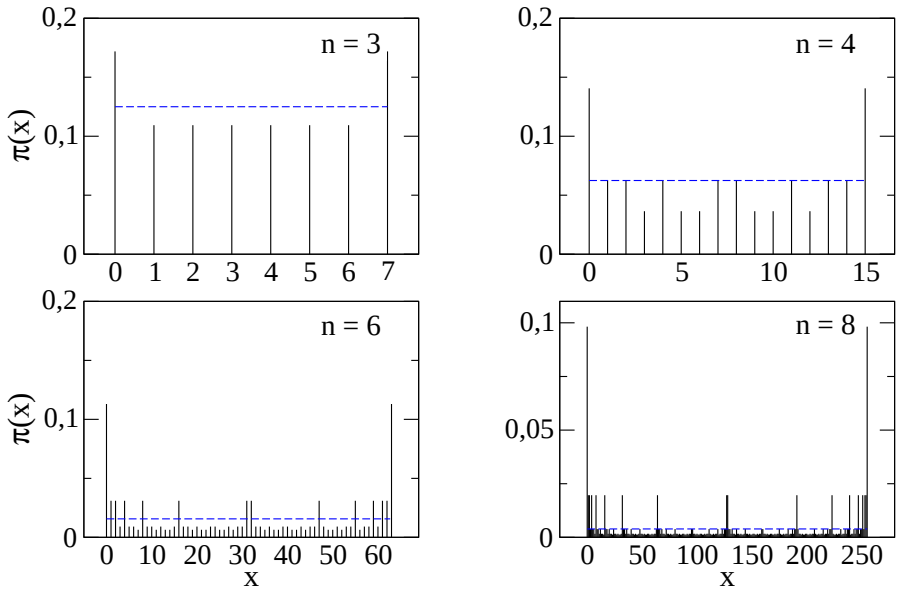

Figura 3.1: Distribuições limite para caminhadas quânticas em hipercubos com $n = 3, 4, 6, 8$ obtidas da Equação (3.22) com a condição inicial (3.12). Como referência, mostramos a distribuição uniforme como uma linha horizontal pontilhada.

Deve-se notar que a identidade $\pi(x) = \pi(2^n - 1 - x)$ também ajuda no cálculo de $\pi(x)$. Claramente, a distribuição assintótica para $\bar{P}(x,t)$ não é uniforme para a condição inicial da Equação (3.12). Note, por exemplo, que para n suficientemente grande, Equação (3.23) nos dá $\pi(0) \approx 1/\sqrt{2\pi(2n+1)} \gg 2^{-n}$. Realizamos implementações numéricas que confirmam a Equação (3.22). Esta distribuição é mostrada para hipercubos de diversas dimensões na Figura 3.1. O máximo da distribuição ocorre no sítio inicial $x_0 = 0$ e em $\bar{x}_0 = 2^n - 1$. Note que $\pi(x)$ toma somente $1 + \lfloor n/2 \rfloor$ valores diferentes, correspondentes aos sítios com distâncias de Hamming $0, 1, 2, \ldots, \lfloor n/2 \rfloor$ em relação ao sítio inicial x_0 ou ao seu oposto, $\bar{x}_0$.

É interessante considerarmos a probabilidade $p(x)$ de encontrar o caminhante a uma distância de Hamming $|x|$ do sítio inicial, para tempos grandes. Há $\binom{n}{|x|}$ sítios com peso de Hamming $|x|$ e, portanto, a probabilidade que estamos considerando é dada por

$$p(|x|) = \binom{n}{|x|} \pi(x). \tag{3.24}$$

A distribuição estacionária $\pi(x)$ depende somente de $|x|$ e pode ser calculada eficientemente a partir da Equação (3.22). A probabilidade $p(|x|)$ toma no máximo

$1 + \lfloor n/2 \rfloor$ valores diferentes, assim como $\pi(x)$. Na Figura 3.2, $p(|x|)$ é comparado com a probabilidade correspondente para uma distribuição uniforme sobre o hipercubo. Nesse caso, a situação mais provável seria encontrar o caminhante com uma distância de Hamming $\sim n/2$ do sítio de partida. Em vez disso, notamos que a situação mais provável para o caminhante quântico coerente com moeda de Grover é encontrá-lo próximo aos vértices x_0 e $\bar{x}_0$ do hipercubo, com $|x| \sim 0$ ou $|x| \sim n$.

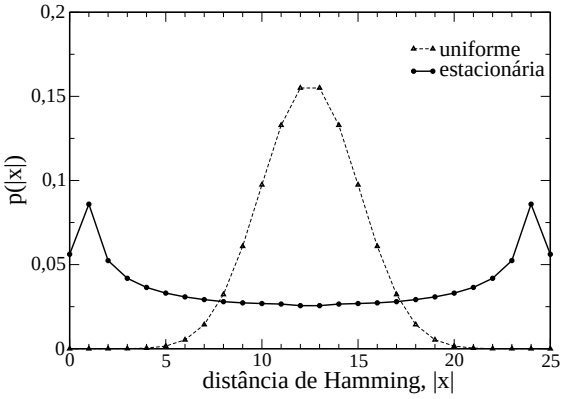

Figura 3.2: Probabilidade assintótica de encontrar o caminhante com uma distância de Hamming $|x|$ em relação ao sítio inicial, dado pela Equação (3.24), para $n = 25$. A distribuição binomial $\frac{1}{2^n}\binom{n}{|x|}$, que corresponde à distribuição de posição uniforme do caminhante, é mostrada para comparação.

Em resumo, o fato do caminhante quântico iniciar localizado em $x_0 = 0$ marca ambos os vértices x_0 e $\bar{x}_0$ como sítios especiais e efeitos persistentes de interferência dão origem a uma distribuição não-uniforme sobre os sítios. A probabilidade de encontrar a partícula com uma dada distância de Hamming varia muito mais lentamente do que para uma distribuição uniforme sobre os sítios. Devemos ressaltar que os resultados obtidos para a distribuição assintótica são dependentes da condição inicial. Por exemplo, já que a superposição uniforme da moeda **e posição**, $\frac{1}{\sqrt{n2^n}} \sum_{j=0}^{n-1} \sum_{x=0}^{2^n-1} |j, x\rangle$, é um autovetor de U com autovalor 1, para esta condição inicial em particular a distribuição permanece uniforme.

3.1.3 *Mixing time* de uma evolução coerente

Agora vamos considerar o *mixing time* de uma evolução coerente. Iremos nos basear em simulações numéricas, que nos permitem extrair informações so-

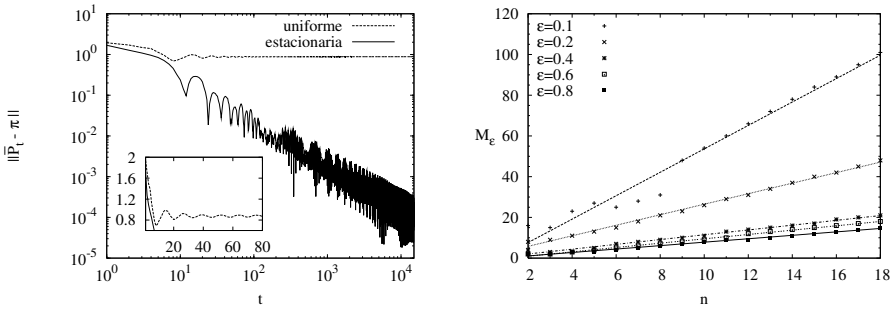

Figura 3.3: Esquerda: distância da distribuição média no instante t até as distribuições uniforme e estacionária, para um caminhante quântico coerente movendo-se sobre um hipercubo de dimensão $n = 8$. Eixos em escala logarítmica no gráfico maior e linear no detalhe. Direita: mixing time em função da dimensão n para diferentes limiares ϵ.

bre o *mixing time* da caminhada quântica no hipercubo, e em resultados gerais apresentados por Aharonov et al. (2001).

O painel esquerdo da Figura 3.3 mostra a dependência temporal da distância entre a distribuição média e a distribuição estacionária, dada pela Equação (3.21). Para tempos grandes, essa distância decai aproximadamente como $\sim 1/t$ enquanto a distância correspondente à distribuição uniforme permanece essencialmente constante.

Para um caminhante quântico em um grafo genérico com condição inicial arbitrária, uma cota superior para distância entre a distribuição de probabilidades média e a distribuição assintótica $\pi(x)$ pode ser obtida a partir do Teorema 3.1, apresentado no início deste capítulo. Particularizando para o hipercubo de dimensão n, temos

$$\|\bar{P}(x,T) - \pi(x)\| \leq \frac{\pi}{T\Delta}\left(1 + \ln(n2^{n-1})\right). \tag{3.25}$$

Esta cota é expressa em termos da separação mínima Δ entre autovalores distintos de U. Para um hipercubo de dimensão n com moeda de Grover, essa separação mínima é $\Delta = \min|e^{i\omega_k} - 1| = 2/\sqrt{n}$. Portanto, para n grande, o *mixing time* médio M_ϵ é limitado por $O\left(\frac{n^{3/2}}{\epsilon}\right)$. O painel direito da Figura 3.3 mostra a dependência linear do *mixing time* médio com a dimensão n e o limiar ϵ, para o estado inicial dado pela Equação (3.12). Para dimensões suficientemente grandes, o *mixing time*

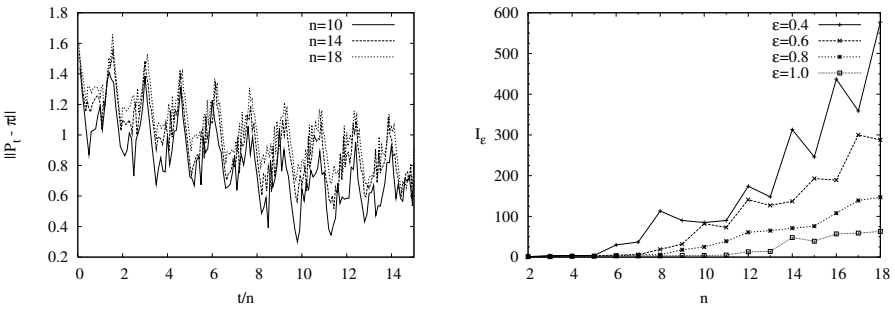

Figura 3.4: Esquerda: distância para a distribuição estacionária $\pi(x)$, como uma função de t/n. Direita: *mixing time* instantâneo I_ϵ para a distribuição estacionária como função da dimensão n.

cresce como n/ϵ, consistentemente com a cota obtida.

Se o caminhante quântico, no instante t, estiver localizado em um vértice com peso de Hamming par (ímpar), sabemos que no instante $t+1$ ele estará em uma superposição de vértices com peso de Hamming ímpar (par). O *mixing time* instantâneo, I_ϵ, o primeiro instante para o qual a distribuição de probabilidades $P(x,t)$ está perto por uma distância ϵ de uma dada distribuição, precisa ser calculado considerando a paridade apropriada para cada passo da caminhada. No painel esquerdo da Figura 3.4, mostramos a variação total da distância da distribuição instantânea para a distribuição estacionária $\pi(x)$, dada pela Equação (3.22), como uma função de t/n, para diversos valores de n. Note que os mínimos locais não se aproximam de zero conforme t aumenta. Portanto, para $\epsilon \lesssim 0.3$ e n fixo, não há mixing time instantâneo com limiar ϵ. No painel direito da Figura 3.4, mostramos o mixing time instantâneo como uma função da dimensão n do hipercubo. Observam-se grandes oscilações em I_ϵ para ϵ pequenos e a dependência em n é claramente não-linear.

Moore e Russell (2002) exploram o mixing time instantâneo para a distribuição uniforme com a paridade apropriada. Esta é uma noção bastante útil que captura o primeiro instante em que a distribuição do caminhante quântico está perto por uma distância ϵ da distribuição uniforme. Na Figura 3.5, mostramos este mixing time instantâneo com limiar ϵ como uma função da dimensão n do hipercubo, para diferentes limiares. Ele aumenta linearmente com n com inclinação

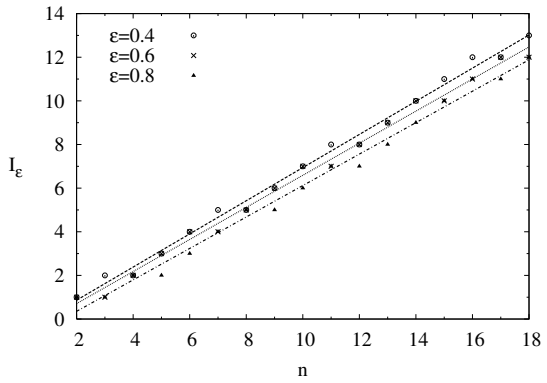

Figura 3.5: Mixing time instantâneo para a distribuição uniforme como função da dimensão n.

próxima a $\pi/4 \approx 0.8$. Como reportado por Moore e Russell (2002) e confirmado por nossos cálculos numéricos, para $t/n = \pi/4$ a distribuição da posição do caminhante quântico no hipercubo de dimensão n é próxima da distribuição uniforme. Entretanto, a distância para a distribuição estacionária $\pi(x)$ não possui essa propriedade, como mostrado no painel esquerdo da Figura 3.4.

3.1.4 Descoerência e *mixing times*

Agora vamos considerar os efeitos da descoerência na distribuição estacionária e no mixing time médio da caminhada quântica no hipercubo. Como fonte de descoerência consideramos ruído topológico que abre ligações entre sítios do hipercubo aleatoriamente, como descrito na Seção 2.5. Os efeitos deste modelo de ruído baseado em ligações interrompidas foi estudado anteriormente para uma caminhada quântica na reta (Romanelli et al., 2005; Abal et al., 2007) e no plano (Oliveira et al., 2006a). O ruído de ligação interrompida é um exemplo de "ruído unitário" (Shapira et al., 2003) que pode afetar a dinâmica de um modo diferente do resultado de realizar medições parciais na moeda ou na posição com probabilidade p. Este tipo de ruído é caracterizado por uma sequência de operações unitárias independentes aplicadas sobre um estado inicial,

$$|\Psi(t)\rangle = U_t U_{t-1} \dots U_1 |\Psi(0)\rangle . \tag{3.26}$$

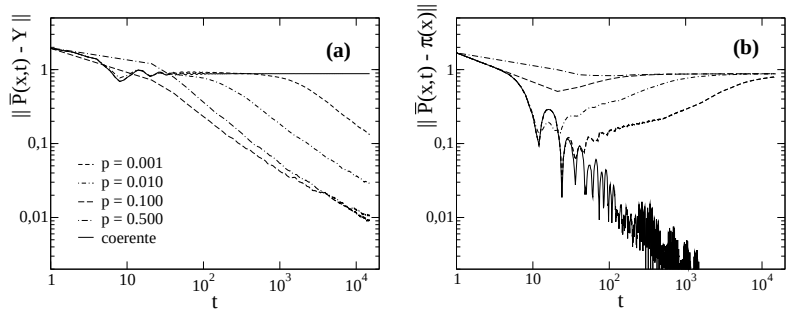

Figura 3.6: Evolução da distância da distribuição média (a) para a distrição uniforme, $Y = 2^{-n}$ e (b) para a distribuição estacionária, $\pi(x)$, obtida da Equação (3.21). Diversas taxas de descoerência são mostradas, juntamente com o caso coerente ($p = 0$), para uma dimensão fixa $n = 8$.

Não são realizadas medições durante a evolução. Cada operador U_i, para $i = 1, \ldots t$, é da forma da equação de evolução original, $U = S \circ (C \otimes I)$, porém com um operador de deslocamento modificado, S', responsável pelo estado corrente da malha, ou seja, responsável por quais ligações estão interrompidas no instante i. A generalização do operador de deslocamento deve preservar a unitariedade. Este operador já foi descrito detalhadamente na Seção 2.5.

Na presença de descoerência, a distribuição estacionária do hipercubo de dimensão n não depende da condição inicial. Na Figura 3.6, mostramos a evolução da distância (a) para a distribuição uniforme e (b) para a distribuição estacionária do caso coerente, dada pela Equação (3.21). Está claro que a introdução até mesmo de uma descoerência fraca faz com que a distribuição assintótica se torne uniforme em um tempo característico $\sim 1/p$. Após este tempo, a distância para a distribuição uniforme decai de acordo com o inverso de uma lei de potências que independe da taxa de descoerência p. Para taxas de descoerência fracas, a distribuição de probabilidades média permanece próxima àquela do caso coerente para um tempo da ordem de $p^{-1/2}$, como mostrado no painel (b) da Figura 3.6.

Também percebe-se a partir da Figura 3.6 que o caso $p = 0.1$ vai mais rapidamente para a distribuição uniforme que os demais casos exibidos, tanto para valores menores como para valores maiores que p. De fato, quando a taxa de descoerência é reduzida ou aumentada próximo a este valor crítico, a taxa de

convergência para a distribuição uniforme se torna menor. O fato de que uma taxa de descoerência crítica p_c minimiza o mixing time foi reportado anteriormente por Kendon e Tregenna (2003) no contexto de um caminhante quântico descoerente no ciclo. A dependência do mixing time de um caminhante quântico no hipercubo com a taxa de descoerência ou, equivalentemente, com a probabilidade p de ligações interrompidas é mostrada no painel esquerdo da Figura 3.7. Um mínimo pode ser identificado próximo a $p_c \approx 0.1$, que corresponde a uma taxa de descoerência que pode proporcionar um mixing time mais rápido no hipercubo. O valor crítico parece independente — ou ao menos fracamente dependente — da dimensão do hipercubo. Esta conclusão é similar à obtida por Kendon e Tregenna (2003) para o ciclo com medições repetidas, apesar de termos empregado aqui um tipo diferente de descoerência em um grafo também muito diferente.

No painel direito da Figura 3.7, mostramos a dependência com a dimensão, do mixing time médio para a distribuição uniforme, para diferentes níveis de descoerência. O *mixing time* linear para a distribuição estacionária do caso coerente — Equação (3.22) — também é mostrado (curvas com círculos). Os mixing times descoerentes aumentam com a dimensão em uma taxa que é aproximadamente $n^{7/3}$, ou seja, ligeiramente mais rápido que quadrático, de modo que *mixing times* com ligações interrompidas são maiores que o *mixing time* coerente para todas as dimensões. Note que os dados na Figura 3.7 também confirmam que a caminhada com $p \approx 0.1$ converge mais rapidamente que caminhadas com outras taxas de descoerência.

Os resultados desta seção foram publicados no periódico *Physical Review A* (Marquezino et al., 2008). Na seção seguinte, discutiremos acerca da distribuição estacionária e do *mixing time* da caminhada quântica em uma malha bidimensional com condições de contorno periódicas.

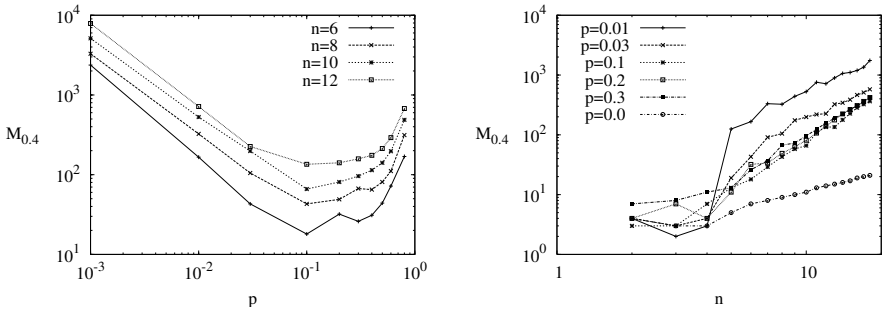

Figura 3.7: Esquerda: *mixing time* médio para a distribuição uniforme em um hipercubo descoerente como uma função da probabilidade de ligações interrompidas p. Direita: a mesma quantidade como função da dimensão n. O *mixing time* médio para a distribuição média para o caso coerente também é mostrado para comparação (curva com círculos).

3.2 Caminhada quântica na malha bidimensional

A análise da caminhada quântica na malha bidimensional com condições de contorno periódicas é de grande interesse, principalmente por suas aplicações algorítmicas. Ambainis et al. (2005) demonstraram como procurar um vértice marcado nesse grafo em tempo $O(\sqrt{N}\log N)$. Posteriormente, Tulsi (2008) conseguiu melhorar esse tempo para $O(\sqrt{N\log N})$. Nesta seção, iremos calcular analiticamente a distribuição estacionária da caminhada para o caso de uma malha de N vértices com $\sqrt{N}$ ímpar. Para isso, antes iremos resolver o problema de autovalores e autovetores do operador de evolução da caminhada. Uma vez calculada essa distribuição, iremos analisar o mixing time da caminhada numericamente. Em seguida, iremos estudar a caminhada com uma modificação no operador de evolução conforme a prescrição do algoritmo AKR de busca na malha. O cálculo da distribuição estacionária dessa caminhada modificada será feito somente de modo numérico, e o cálculo será utilizado para estudar o *mixing time*. Nossos resultados sugerem uma possível relação entre o *mixing time* e o tempo de execução do algoritmo de busca.

3.2.1 A caminhada coerente com inversão de moeda

A caminhada quântica na malha bidimensional já foi estudada na Seção 2.3. Então, havíamos considerado a caminhada em uma malha infinita. Agora, estamos

interessados no comportamento do caminhante quântico em uma malha finita de dimensões $\sqrt{N} \times \sqrt{N}$, i.e., com N vértices, e com condição de contorno periódica. A partícula portanto vive no espaço de Hilbert $\mathcal{H}_2 \otimes \mathcal{H}_2 \otimes \mathcal{H}_P$, em que $\mathcal{H}_2 \otimes \mathcal{H}_2$ é o subespaço associado à moeda, de dimensão 4, e $\mathcal{H}_P$ é o subespaço associado à posição, de dimensão N. A base canônica para o subespaço-moeda é $\mathcal{B}_C = \{|d, s\rangle : 0 \leq d, s \leq 1\}$ e a base canônica para o subespaço-posição é $\mathcal{B}_P = \{|x, y\rangle : 0 \leq x, y \leq \sqrt{N}\}$. O estado genérico do caminhante no instante t é, portanto,

$$|\Psi(t)\rangle = \sum_{d,s=0}^{1} \sum_{x,y=0}^{\sqrt{N}-1} \psi_{d,s;x,y}(t) \, |d, s\rangle \, |x, y\rangle \, . \tag{3.27}$$

O operador de evolução para um passo da caminhada é $U = S \circ (C \otimes I)$, como nos casos anteriores, com S representando o operador de deslocamento, C representando o operador moeda e I representando o operador identidade no subespaço-posição. O operador C utilizado será a moeda de Grover, já definida anteriormente. O operador de deslocamento considerado, porém, é diferente de todos aqueles vistos no Capítulo 2. Definimo-lo como

$$S = \sum_{d,s=0}^{1} \sum_{x,y=0}^{\sqrt{N}-1} |d, 1-s\rangle \langle d, s| \otimes |x + (-1)^s \delta_{d0}, y + (-1)^s \delta_{d1}\rangle \langle x, y| \, , \tag{3.28}$$

com as somas na posição sendo efetuadas modulo $\sqrt{N}$.

Note que, além de estarmos considerando um caminhante em malha finita, também estamos definindo um operador de deslocamento que efetua uma inversão do sentido da moeda após cada passo. A razão disto é o fato desta escolha possuir consequências interessantes para os algoritmos abstratos de busca (Ambainis et al., 2005).

Para calcularmos a distribuição limite da caminhada quântica na malha bidimensional, primeiro precisamos resolver o problema de autovalor e autovetor de seu operador de evolução. Esse problema torna-se mais simples se considerarmos a caminhada coerente no espaço de Fourier.

Assim como na análise do hipercubo empregamos a transformada de Fourier no grupo $\mathbb{Z}_2^n$, na análise da malha bidimensional temos de considerar a transformada no grupo apropriado, a fim de que efetivamente os cálculos sejam simplificados. A malha bidimensional é um grafo de Cayley do grupo $\mathbb{Z}^2_{\sqrt{N}}$, de modo que devemos aplicar a transformada de Fourier deste grupo, que é gerada pela base de N vetores

$$\left|\chi_{k_x,k_y}\right\rangle = \frac{1}{\sqrt{N}} \sum_{x,y=0}^{\sqrt{N}-1} \omega^{xk_x+yk_y} \left|x,y\right\rangle, \tag{3.29}$$

com $\omega = e^{\frac{2\pi i}{\sqrt{N}}}$.

As componentes do operador de evolução no espaço de Fourier são

$$\left\langle d,s,\kappa_{k'_x,k'_y}\right| U \left|d',s',\kappa_{k_x,k_y}\right\rangle = \omega^{(-1)^s(\delta_{d0}k_x+\delta_{d1}k_y)} G_{d,s\oplus 1;\,d',s'}\,\delta_{k_x,k'_x}\delta_{k_y,k'_y}. \tag{3.30}$$

Para cada k_x, k_y, definimos um operador de evolução reduzido no espaço da moeda, dado por

$$U_{k_x,k_y}(d,s;\,d',s') = \omega^{(-1)^s(\delta_{d0}k_x+\delta_{d1}k_y)} G_{d,1-s;\,d',s'}, \tag{3.31}$$

que é uma matriz 4×4 e pode ser diagonalizado. Os autovetores de U são o produto tensorial dos autovetores de U_{k_x,k_y} e $\left|\kappa_{k_x,k_y}\right\rangle$. Agora, passamos a descrever o espectro de U_{k_x,k_y}.

Se $k_x = 0$ e $k_y = 0$, então os autovetores de U_{k_x,k_y} com autovalor 1 são $\left|s^C\right\rangle \equiv \frac{1}{2}(1,1,1,1)^T$, $\frac{1}{\sqrt{2}}(1,-1,0,0)^T$ e $\frac{1}{\sqrt{2}}(0,0,1,-1)^T$. O autovetor com autovalor -1 é $\frac{1}{\sqrt{2}}(1,1,-1,-1)^T$.

Se $k_x \neq 0$ ou $k_y \neq 0$, então os autovetores de U_{k_x,k_y} com autovalor 1 são

$$\left|\nu^{+1}_{k_x,k_y}\right\rangle = \frac{1}{4\sin\frac{\theta}{2}} \begin{pmatrix} \omega^{k_x}\left(\omega^{k_y}-1\right) \\ 1-\omega^{k_y} \\ \omega^{k_y}\left(1-\omega^{k_x}\right) \\ \omega^{k_x}-1 \end{pmatrix} \tag{3.32}$$

e os autovetores com autovalor -1 são

$$\left|\nu^{-1}_{k_x,k_y}\right\rangle = \frac{1}{4\cos\frac{\theta}{2}} \begin{pmatrix} -\omega^{k_x}\left(1+\omega^{k_y}\right) \\ -\left(1+\omega^{k_y}\right) \\ \omega^{k_y}\left(1+\omega^{k_x}\right) \\ 1+\omega^{k_x} \end{pmatrix}, \tag{3.33}$$

em que

$$\cos\theta = \frac{1}{2}\left(\cos\frac{2\pi k_x}{\sqrt{N}} + \cos\frac{2\pi k_y}{\sqrt{N}}\right). \tag{3.34}$$

Os autovetores com autovalores $e^{i\theta}$ são

$$\left|\nu^{+\theta}_{k_x,k_y}\right\rangle = \frac{i}{2\sqrt{2}\sin\theta} \begin{pmatrix} e^{-i\theta} - \omega^{k_x} \\ e^{-i\theta} - \omega^{-k_x} \\ e^{-i\theta} - \omega^{k_y} \\ e^{-i\theta} - \omega^{-k_y} \end{pmatrix}. \tag{3.35}$$

Note que θ depende de k_x e de k_y. Substituindo θ por $-\theta$, obtemos os autovetores com autovalor $e^{-i\theta}$. Os autovetores $\left|\nu^{\pm\theta}_{k_x,k_y}\right\rangle$ possuem norma unitária e $\langle\nu^{\pm\theta}_{k_x,k_y}|s^C\rangle = \frac{1}{\sqrt{2}}$. Eles formam uma base ortonormal para o espaço reduzido.

Tomamos o estado

$$|\Psi(0)\rangle = \left|s^C\right\rangle|x=0,y=0\rangle \tag{3.36}$$

como a condição inicial, ou seja, o caminhante começa no ponto $(0,0)$ da malha e uniformemente distribuído no subespaço-moeda. Na base de autovetores, a condição inicial é dada por

$$|\Psi(0)\rangle = \frac{1}{\sqrt{N}}\left|s^C\right\rangle|\chi_{0,0}\rangle + \frac{1}{\sqrt{2N}}\sum_{\substack{k_x,k_y=0 \\ (k_x,k_y)\neq(0,0)}}^{\sqrt{N}-1}\left|\nu^{\theta}_{k_x,k_y}\right\rangle\left|\chi_{k_x,k_y}\right\rangle + \left|\nu^{-\theta}_{k_x,k_y}\right\rangle\left|\chi_{k_x,k_y}\right\rangle. \tag{3.37}$$

Aplicando U^t em $|\Psi(0)\rangle$, obtemos o estado do caminhante quântico no instante t,

que é dado por

$$|\Psi(t)\rangle = \frac{1}{\sqrt{N}} \left|s^C\right\rangle |\chi_{0,0}\rangle + \frac{1}{\sqrt{2N}} \sum_{\substack{k_x,k_y=0 \\ (k_x,k_y)\neq(0,0)}}^{\sqrt{N}-1} e^{i\theta t} \left|\nu^{\theta}_{k_x,k_y}\right\rangle \left|\chi_{k_x,k_y}\right\rangle + e^{-i\theta t} \left|\nu^{-\theta}_{k_x,k_y}\right\rangle \left|\chi_{k_x,k_y}\right\rangle . \tag{3.38}$$

3.2.2 Distribuição limite

Para calcular a distribuição limite na malha bidimensional finita com moeda de Grover e condições de contorno periódicas, será útil considerarmos o seguinte teorema, válido para grafos mais gerais.

Teorema 3.2 (Aharonov et al. (2001))**.** *Seja U o operador de evolução do caminhante quântico e sejam $|\phi_j\rangle, \lambda_j$ os autovetores e autovalores de U, respectivamente. Seja v um vértice qualquer do grafo. Para um estado inicial dado por $|\Psi(0)\rangle = \sum_j a_j |\phi_j\rangle$, temos*

$$\lim_{T\to\infty} \bar{P}(v,t) = \sum_{i,j,a} a_i a_j^* \langle a, v|\phi_i\rangle\langle\phi_j|a, v\rangle, \tag{3.39}$$

em que o somatório corre somente nos pares i, j tais que $\lambda_i = \lambda_j$.

Particularizando o Teorema 3.2 para a malha bidimensional e usando os coeficientes da condição inicial (3.37), além do fato de que $\langle\nu^{\theta'}_{k'_x,k'_y}|\nu^{\theta}_{k_x,k_y}\rangle = \langle\nu^{-\theta'}_{k'_x,k'_y}|\nu^{-\theta}_{k_x,k_y}\rangle$, podemos simplificar a expressão da distribuição limite para

$$\pi(x,y) = \frac{1}{N^2} + \frac{1}{N^2} \sum_{\substack{(k_x,k_y)\neq(0,0) \\ (k'_x,k'_y)\neq(0,0) \\ \theta(k_x,k_y)=\theta(k'_x,k'_y)}} \sum \langle\nu^{\theta}_{k'_x,k'_y}|\nu^{\theta}_{k_x,k_y}\rangle \omega^{x(k_x-k'_x)+y(k_y-k'_y)}. \tag{3.40}$$

Quando $\theta(k'_x,k'_y) = \theta(k_x,k_y)$, temos

$$\langle\nu^{\theta}_{k'_x,k'_y}|\nu^{\theta}_{k_x,k_y}\rangle = \frac{1 - 2\cos^2\theta(k_x,k_y) + \cos\theta(k_x-k'_x, k_y-k'_y)}{2\sin^2\theta(k_x,k_y)}. \tag{3.41}$$

Precisamos analisar todos casos em que $\theta(k'_x,k'_y) = \theta(k_x,k_y)$. Quando $\sqrt{N}$ é

ímpar, podemos fixar (k_x, k_y) e obter os seguintes casos para (k'_x, k'_y):

(1) Se $k_x \neq 0$ e $k_y = 0$, então

$$(k'_x, k'_y) = (k_x, 0), (0, k_x), (-k_x, 0) \text{ ou } (0, -k_x) \mod \sqrt{N}.$$

(2) Se $k_x = 0$ e $k_y \neq 0$, então

$$(k'_x, k'_y) = (0, k_y), (k_y, 0), (0, -k_y) \text{ ou } (-k_y, 0) \mod \sqrt{N}.$$

(3) Se $k_x \neq 0$, $k_y \neq 0$ e $k_x \equiv \pm k_y \mod \sqrt{N}$, então

$$(k'_x, k'_y) = (k_x, k_y), (k_x, -k_y), (-k_x, k_y) \text{ ou } (-k_x, -k_y) \mod \sqrt{N}.$$

(4) Se $k_x \neq 0$, $k_y \neq 0$ e $k_x \not\equiv \pm k_y \mod \sqrt{N}$, então

$$\begin{aligned}(k'_x, k'_y) = (k_x, k_y), (k_y, k_x), (-k_x, k_y), (k_y, -k_x),& \\ (k_x, -k_y), (-k_y, k_x), (-k_x, -k_y) \text{ ou } (-k_y, -k_x) \mod \sqrt{N}.&\end{aligned}$$

Abrindo o segundo somatório da Equação (3.40) para cada par (k_x, k_y) e realizando algumas simplificações, podemos chegar à seguinte expressão para a

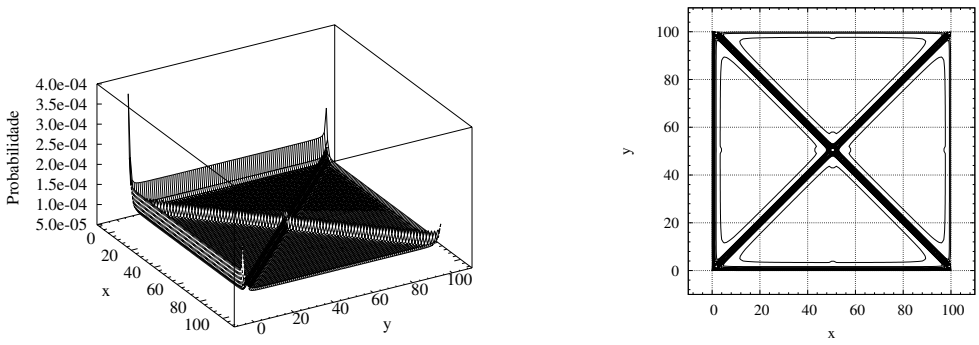

Figura 3.8: Esquerda: distribuição limite para um caminhante quântico em malha bidimensional com $\sqrt{N} = 101$, obtida a partir da Equação (3.42) com condição inicial (3.36). Direita: gráfico de contorno para a mesma distribuição.

distribuição limite:

$$\begin{aligned}
\pi(x,y) = \frac{1}{N} + \frac{2}{N^2}\sum_{k_x=1}^{\sqrt{N}-1}\frac{1}{3+\cos\tilde{k}_x}\times \\
\left(\left(\cos\tilde{k}_x(x-y)+\omega^{k_x(x+y)}\right)\left(1+\cos\tilde{k}_x\right)+2\left(\omega^{2xk_x}+\omega^{2yk_x}\right)\right)+ \\
+\frac{1}{2N^2}\sum_{\substack{k_x,k_y=1 \\ k_y\notin\{k_x,\sqrt{N}-k_x\}}}^{\sqrt{N}-1}\frac{1}{\sin^2\theta}\Bigg(\omega^{x(k_x-k_y)+y(k_y-k_x)}\left(\cos\left(\tilde{k}_x-\tilde{k}_y\right)-\cos 2\theta\right)+ \\
+\omega^{2xk_x}\left(\cos^2\tilde{k}_x-\cos 2\theta\right)+\omega^{2yk_y}\left(\cos^2\tilde{k}_y-\cos 2\theta\right)+ \\
+\omega^{x(k_x-k_y)+y(k_x+k_y)}\left(\cos\theta(k_x-k_y,k_x+k_y)-\cos 2\theta\right)+ \\
+\frac{1}{2}\omega^{x(k_x+k_y)+y(k_y-k_x)}\left(\sin^2\tilde{k}_x+\sin^2\tilde{k}_y\right)+ \\
+\frac{1}{2}\omega^{2xk_x+2yk_y}\left(\cos\tilde{k}_x-\cos\tilde{k}_y\right)^2 \\
+\omega^{x(k_x+k_y)+y(k_x+k_y)}\left(\cos\left(\tilde{k}_x+\tilde{k}_y\right)-\cos 2\theta\right)\Bigg), \quad (3.42)
\end{aligned}$$

em que $\tilde{k}_x = \frac{2\pi k_x}{\sqrt{N}}$ e $\tilde{k}_y = \frac{2\pi k_y}{\sqrt{N}}$.

A distribuição é mostrada para uma malha de dimensões 101×101 na Figura 3.8. O máximo da distribuição ocorre no sítio inicial $(x_0, y_0) = (0, 0)$.

3.2.3 Mixing time e algoritmo de busca

Nesta seção, iremos considerar inicialmente o *mixing time* da evolução coerente descrita anteriormente. Em seguida, realizaremos uma análise numérica da distribuição estacionária da caminhada com operador moeda modificado. A modificação que iremos introduzir na moeda corresponde ao formalismo do algoritmo abstrato de busca, como veremos no Capítulo 4.

Pelo Teorema 3.1, temos uma cota superior para a distância entre a distribuição de probabilidades média e a distribuição estacionária, para a caminhada quântica em um grafo genérico com condição inicial arbitrária. Particularizando para a malha bidimensional com N vértices, como o grau de cada vértice é $d = 4$, temos

$$\|\bar{P}(x,T) - \pi(x)\| \leq \frac{\pi}{T\Delta}\left(\ln 2N + 1\right), \tag{3.43}$$

em que Δ é a separação mínima entre autovalores distintos de U. O valor de Δ, portanto, é o valor mínimo de $|e^{i\theta(k_x,k_y)} - e^{i\theta(k'_x,k'_y)}|$ para k_x, k_y e k'_x, k'_y na faixa de $[0, \sqrt{N} - 1]$. Para N grande e valores pequenos de k_x, k_y e k'_x, k'_y, temos

$$\Delta \approx \sqrt{\frac{2\pi}{N}}\left|\sqrt{k_x^2 + k_y^2} - \sqrt{k_x'^2 + k_y'^2}\right|. \tag{3.44}$$

Portanto, obtemos o valor mínimo de Δ tomando valores de k_x, k_y e k'_x, k'_y tais que $\theta(k_x, k_y) \neq \theta(k'_x, k'_y)$ e tais que

$$\sqrt{k_x^2 + k_y^2} \approx \sqrt{k_x'^2 + k_y'^2}. \tag{3.45}$$

Podemos fazer bons palpites examinando a estrutura da equação acima. Com esses palpites podemos tentar encontrar uma cota superior para M_ϵ numericamente. Os valores de Δ que encontramos são $O(1/\sqrt{N})$ ou $O(1/N)$. Portanto, os melhores palpites para M_ϵ são $O\left(\frac{\sqrt{N\log N}}{\epsilon}\right)$ ou $O\left(\frac{N\log N}{\epsilon}\right)$. De fato, plotando M_ϵ contra $\sqrt{N\log N}$ para diversos valores de ϵ, obtivemos as linhas retas da Figura 3.9. Os dados numéricos, apesar de ainda não serem conclusivos, sugerem que uma cota

superior para o crescimento do *mixing time* seja dado por

$$M_\epsilon = O\left(\frac{\sqrt{N \log N}}{\epsilon}\right). \tag{3.46}$$

No painel esquerdo da Figura 3.9, nós temos a variação total da distância entre a distribuição de probabilidades média e as distribuições uniforme e estacionária. Este resultado numérico está consistente com uma distribuição estacionária não-uniforme para o caminhante quântico na malha bidimensional periódica, conforme já havíamos observado na Figura 3.8, obtida a partir da Equação (3.42). Também observamos que a distribuição média converge para a estacionária segundo uma lei de potências próxima de $1/t$.

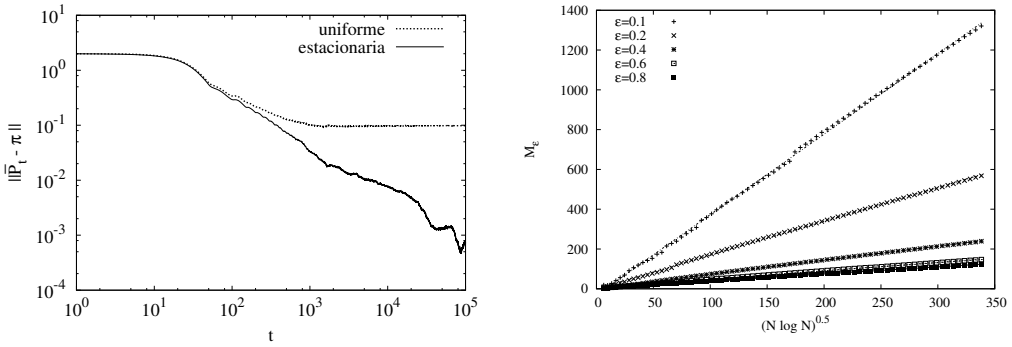

Figura 3.9: Painel esquerdo: variação total da distância entre a distribuição média e as distribuições uniforme e estacionária do caminhante na malha bidimensional com o operador de deslocamento da Equação (3.28), em função do tempo. Painel direito: *mixing time* para a distribuição estacionária em função do tamanho da malha.

Nosso interesse ao analisar o *mixing time* da caminhada usando o operador de deslocamento com inversão de moeda — definido na Equação (3.28) — é a aplicação ao algoritmo de busca por um vértice marcado na malha bidimensional. O algoritmo será explicado mais detalhadamente no Capítulo 4. Por enquanto, basta considerarmos uma caminhada na malha bidimensional que, além de usar o operador de deslocamento com inversão de moeda, também utiliza um operador de moeda modificado,

$$C' = -I \otimes |x_0, y_0\rangle \langle x_0, y_0| + G \otimes (I - |x_0, y_0\rangle \langle x_0, y_0|), \tag{3.47}$$

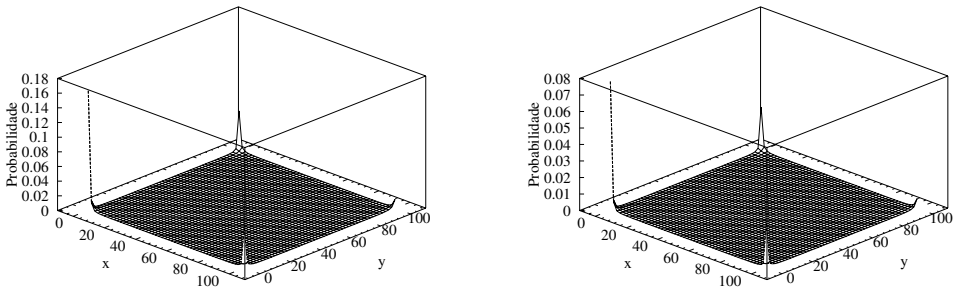

Figura 3.10: Caminhada quântica em malha bidimensional com $\sqrt{N} = 101$ e uma moeda modificada usada para procurar por um vértice marcado. Painel esquerdo: distribuição de probabilidades após $t = 200$ passos, correspondente ao instante de máxima probabilidade no vértice marcado. Painel direito: distribuição estacionária aproximada com $T = 10^4$ passos de simulação.

em que $|x_0, y_0\rangle$ é o vértice procurado. Ou seja, o operador C' aplica a moeda $-I$ se o caminhante está no vértice procurado e aplica a moeda de Grover usual, G, caso contrário. Portanto, o operador C' marca o vértice $|x_0, y_0\rangle$ com um sinal negativo. Com a definição desse operador de evolução, Ambainis et al. (2005) mostraram que um caminhante quântico partindo da distribuição uniforme vai para o vértice marcado após $O(\sqrt{N \log N})$ passos, com probabilidade $O(\sqrt{\log N})$. Isso significa que o algoritmo deve ser repetido em média $O\left(\frac{1}{\sqrt{\log N}}\right)$ vezes. Tulsi (2008) melhorou esse resultado, mostrando como obter probabilidade $O(1)$ no vértice marcado.

O efeito do operador de evolução modificado é aumentar a probabilidade de encontrar o caminhante no vértice marcado em alguns instantes específicos, o primeiro dos quais pode ser calculado pelo método empregado por Ambainis et al. (2005). O painel esquerdo da Figura 3.10 mostra a distribuição de probabilidades para o caso $\sqrt{N} = 101$ após $t = 200$ passos, o que corresponde ao instante de probabilidade máxima no vértice marcado. O tempo de execução do algoritmo pode ser associado ao *mixing time* instantâneo tomando como distribuição de referência aquela com probabilidade máxima no vértice marcado e tomando um valor pequeno de ϵ. Esta observação, no entanto, não é muito útil, já que não conhecemos essa distribuição *a priori*.

Por outro lado, conhecemos a distribuição estacionária para malhas com $\sqrt{N}$

ímpar e podemos utilizá-la como distribuição de referência para o cálculo do *mixing time*. O painel direito da Figura 3.11 sugere que uma cota superior para o *mixing time* médio da caminhada seja dada por

$$M_\epsilon = O\left(\frac{\sqrt{N \log N}}{\epsilon}\right). \tag{3.48}$$

Desse modo, nosso resultado indica uma possível relação entre o *mixing time* e a complexidade do algoritmo de busca no grafo. Há interesse na investigação dessa relação em um trabalho futuro.

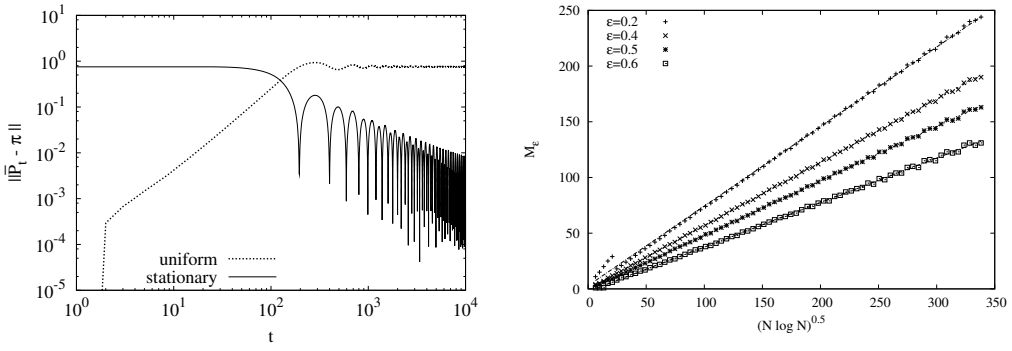

Figura 3.11: Painel esquerdo: variação total da distância entre a distribuição média e as distribuições uniforme e estacionária para o caminhante quântico em malha bidimensional com moeda modificada para buscar um vértice marcado. Painel direito: *mixing time* para a distribuição estacionária em função do tamanho da malha.

3.3 Discussões

Neste capítulo, apresentamos resultados originais de nossa pesquisa referentes a análise de distribuição limite e do *mixing time* das caminhadas no hipercubo e na malha bidimensional finita com condições de contorno periódicas. Inicialmente, consideramos a caminhada no hipercubo de dimensão n. O efeito de descoerência de ligações interrompidas — um tipo específico de ruído unitário que não envolve medições — sobre propriedades do *mixing time* da caminhada também foi investigado.

A distribuição estacionária para a caminhada coerente no hipercubo de di-

mensão n com moeda de Grover foi encontrada analiticamente, para o caso particular da condição inicial simétrica. Concluímos que ela em geral *não é* a distribuição uniforme, como no caso clássico, mas depende fortemente da condição inicial. No entanto, para a condição inicial considerada, observa-se que em sua distribuição estacionária todos os *pesos de Hamming* são aproximadamente equiprováveis.

De acordo com nossas simulações numéricas, o *mixing time* M_ϵ do hipercubo de dimensão n cresce como $O(n/\epsilon)$. Mostramos que este fato é consistente com um resultado mais geral de Aharonov et al. (2001), que particularizado para o hipercubo de dimensão n, fornece uma cota superior de $O(\frac{n^{3/2}}{\epsilon})$.

A variação total da distância entre a distribuição instantânea e a distribuição uniforme mostra um mínimo local para $t = \frac{\pi}{4}n$, como reportado por Moore e Russell (2002), porém um efeito similar em relação à distribuição estacionária não foi observado. Também encontramos que o *mixing time* instantâneo para a distribuição estacionária, quando existe, possui dependência não-linear com a dimensão do hipercubo. Estes resultados estabelecem uma conexão entre as caminhadas quânticas discreta e contínua no tempo, no hipercubo. A distribuição média de ambas as caminhadas no hipercubo de dimensão n não converge para a distribuição uniforme, porém ambas são uniformes ou próximas da distribuição uniforme por uma distância ϵ em certos instantes ($t = \frac{\pi}{4}n$).

Descoerência, mesmo a taxas pequenas, fazem com que a distribuição estacionária seja uniforme. O decaimento da variação total da distância ocorre após um tempo característico $1/p$ e segue uma lei de potências inversa, independente de p. Para o caso de ruído unitário de ligações interrompidas, que não envolve medições, uma taxa de decaimento ótima foi encontrada para o qual o *mixing time* é mínimo. No *mixing time* em função da dimensão, a mesma taxa de descoerência ótima proporciona uma convergência mais rápida no caso descoerente. Um efeito similar foi reportado por Kendon e Tregenna (2003) para o ciclo com descoerência de medições parciais.

Além do estudo da caminhada no hipercubo, também analisamos o com-

portamento do caminhante sobre uma malha bidimensional finita com condições de contorno periódicas. Foi utilizado um operador de deslocamento que efetua inversões na moeda a cada passo, à semelhança do algoritmo de busca AKR. Calculamos analiticamente a distribuição estacionária da caminhada nesse grafo, para malhas com N vértices e $\sqrt{N}$ ímpar, fazendo também algumas considerações sobre o *mixing time*. Nossas simulações numéricas indicam que o *mixing time* cresce como $O\left(\frac{\sqrt{N}\log N}{\epsilon}\right)$. Em seguida, calculamos numericamente a distribuição estacionária para uma caminhada que, além do operador de deslocamento com inversão de moeda, também utiliza um operador de moeda modificado. Essa moeda marca um determinado vértice, fazendo com que a probabilidade do caminhante ser encontrado ali em certos instantes seja amplificada. A distribuição estacionária da caminhada com moeda modificada foi obtida numericamente por meio de simulação. A curva do *mixing time* M_ε obtido, para diversos valores de ε, sugere um crescimento dado por $M_\varepsilon = O\left(\frac{\sqrt{N}\log N}{\varepsilon}\right)$, indicando uma relação entre o *mixing time* e a complexidade do algoritmo de busca no grafo.

Capítulo 4

Algoritmo abstrato de busca

Após o trabalho seminal de Grover (1996), sabe-se que um computador quântico pode buscar um elemento em um banco de dados não-estruturado em tempo quadraticamente mais rápido que um computador clássico. Recentemente, diversos algoritmos de busca foram desenvolvidos tendo como base o modelo de caminhadas quânticas em tempo discreto. Trata-se de uma formulação diferente em muitos aspectos daquela proposta inicialmente por Grover, mas que pode trazer vantagens em implementações práticas. O modelo de caminhadas quânticas é particularmente interessante quando consideramos o problema de *busca espacial*. Trata-se de uma variação do problema de busca em que os N itens do espaço de busca estão armazenados em N posições diferentes na memória, sendo necessário tempo para mover-se entre estas posições. Em algumas instâncias do problema de busca espacial, o algoritmo de Grover pode não alcançar ganho quadrático de complexidade. Para a busca espacial na malha bidimensional, por exemplo, foi demonstrado por Benioff (2002) que um algoritmo quântico baseado no algoritmo de Grover não tem ganho em relação ao algoritmo clássico, ou seja, possui complexidade $O(N)$. Aaronson e Ambainis (2003) conseguiram um algoritmo mais eficiente para esse problema, de complexidade $O(\sqrt{N}\log^2 N)$. Childs e Goldstone (2004) mostraram que a busca espacial em malhas bidimensionais, se baseada no modelo de caminhada quântica *contínua* no tempo, possui complexidade $\Omega(N)$. Entretanto, Ambainis et al. (2005) mostraram que um algoritmo quântico para o mesmo problema, baseado em uma

caminhada quântica *discreta* no tempo, é mais eficiente que todas as tentativas anteriores. O algoritmo de Ambainis et al. (2005) possui complexidade $O(\sqrt{N}\log N)$, e ainda pôde ser melhorado pelo algoritmo de Tulsi (2008), alcançando complexidade $O(\sqrt{N\log N})$. No entanto, ainda não há uma prova de otimalidade para este algoritmo. O primeiro algoritmo quântico para o problema de busca espacial com desempenho ótimo foi dado por Shenvi et al. (2003b), para buscas no hipercubo.

Outro aspecto interessante de muitos dos algoritmos quânticos de busca baseados em caminhadas quânticas é o fato de serem instâncias de um algoritmo mais geral, definido para um grafo regular genérico, conhecido como *algoritmo abstrato de busca*. O próprio algoritmo de Grover pode ser visto como caso particular do algoritmo abstrato de busca em um grafo completo. O algoritmo de Shenvi-Kempe-Whaley (SKW) é uma aplicação do algoritmo abstrato de busca ao hipercubo de dimensão n (Shenvi et al., 2003b). Ele encontra o elemento buscado em tempo $O(\sqrt{2^n})$, contra $O(2^n)$ do algoritmo clássico. O algoritmo de Ambainis-Kempe-Rivosh (AKR) é o algoritmo abstrato de busca para a malha d-dimensional finita com condições de contorno periódicas (Ambainis et al., 2005). Tomando N como o número total de sítios da malha, pode-se verificar que o algoritmo AKR para o caso bidimensional possui complexidade $O(\sqrt{N}\log N)$, contra $O(N)$ do algoritmo clássico. O algoritmo de Tulsi (2008) melhora o algoritmo AKR, reduzindo sua complexidade para $O(\sqrt{N\log N})$.

Este capítulo está organizado da seguinte forma. Na Seção 4.1, descrevemos o algoritmo de Grover. Na Seção 4.2, descrevemos o algoritmo de Shenvi-Kempe-Whaley (SKW). Na Seção 4.3, descrevemos o algoritmo de Ambainis-Kempe-Rivosh (AKR). Na Seção 4.4, descrevemos o algoritmo de Tulsi. No Apêndice B, apresentamos uma análise do algoritmo abstrato de busca, generalizando a discussão deste capítulo.

4.1 Algoritmo de Grover

Nesta seção faremos uma breve revisão do algoritmo de Grover. Para detalhes adicionais, consulte o livro de Nielsen e Chuang (2000) ou o livro de Portugal et al. (2004). O objetivo do algoritmo de Grover é determinar a posição de um elemento t em uma lista não-ordenada. Equivalentemente, podemos definir este problema da seguinte forma. Suponhamos que o domínio de uma função f é $\{0, 1, \cdots, N-1\}$ e que sua imagem é

$$f(x) = \begin{cases} 1, & \text{se } x = t \\ 0, & \text{caso contrário.} \end{cases} \tag{4.1}$$

Suponhamos que alguém tenha implementado a função f em um computador clássico, sem nos fornecer detalhes desta implementação — ou seja, sem nos informar o valor t. Podemos obter a imagem de qualquer valor de entrada utilizando a função f implementada. No entanto, somente através de busca exaustiva podemos descobrir o valor de x para o qual $f(x) = 1$. Portanto, $O(N)$ é a complexidade do melhor algoritmo clássico para encontrar o elemento buscado t.

Agora consideremos um computador quântico de n q-bits, em que $n = \log N$. Seu estado é descrito por um vetor unitário em um espaço vetorial de dimensão $N = 2^n$. A base ortonormal mais simples para este espaço vetorial é $\{|0\rangle, |1\rangle, \cdots, |N-1\rangle\}$, usualmente chamada de base computacional. Suponhamos que alguém tenha implementado o operador de reflexão $R_t = I - 2|t\rangle\langle t|$ com relação ao estado buscado $|t\rangle$, em um computador quântico, sem nos fornecer detalhes desta implementação, do mesmo modo como procedemos no caso anterior. Note que

$$R_t |x\rangle = \begin{cases} -|x\rangle, & \text{se } |x\rangle = |t\rangle \\ |x\rangle, & \text{se } |x\rangle \perp |t\rangle. \end{cases} \tag{4.2}$$

O operador R_t, portanto, marca o elemento procurado, modificando sua fase em relação aos demais. Desse modo, funciona de modo análogo ao oráculo f do algoritmo clássico. No computador quântico, entretanto, podemos nos aproveitar do

paralelismo quântico e aplicar o operador R_t ao estado superposto

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle \,, \tag{4.3}$$

também chamado de estado diagonal. Esse estado pode ser obtido eficientemente, em tempo $O(\log N)$, através da aplicação da transformada de Hadamard ao estado $|0\rangle$. É interessante notar que, devido à linearidade do operador R_t, ao aplicá-lo ao estado diagonal, estamos efetuando uma consulta simultânea a todos os elementos do espaço de busca e marcando somente o elemento procurado.

Após a aplicação de R_t, a amplitude do estado procurado sofreu apenas uma mudança de fase, mas não de magnitude. Portanto, o resultado de uma medição efetuada nesse momento teria uma probabilidade muito baixa de sucesso, a saber, $1/N$. Para amplificar a amplitude do estado procurado ainda é necessário mais um operador de reflexão, $R_s = I - 2\,|s\rangle\,\langle s|$, com relação ao estado diagonal $|s\rangle$. É fácil verificar que o estado diagonal é um autovetor de R_s com autovalor 1.

A aplicação sucessiva de $U_G = R_s R_t$ sobre o estado diagonal rotaciona o estado no subespaço bidimensional gerado por $|s\rangle$ e $|t\rangle$, de modo que após $O(\sqrt{N})$ iterações o estado do computador quântico está muito próximo do estado procurado. Logo após o último passo do algoritmo, pode-se efetuar uma medição no registrador e obter o estado $|t\rangle$ com probabilidade $1 - O(1/\sqrt{N})$.

4.2 Algoritmo de Shenvi-Kempe-Whaley

O algoritmo SKW efetua uma busca no hipercubo de dimensão n por meio de uma caminhada quântica. Uma descrição da caminhada quântica no hipercubo pode ser encontrada na Seção 2.5. No algoritmo SKW, tomamos como condição inicial o estado diagonal $|\Psi_0\rangle = \left|s^C\right\rangle \otimes \left|s^P\right\rangle$, em que $\left|s^C\right\rangle$ é a superposição uniforme sobre todos os n estados da moeda, e $\left|s^P\right\rangle$ é a superposição uniforme sobre todos os 2^n vértices do hipercubo.

Consideremos inicialmente a caminhada com a moeda

$$C_0 = -I + 2\left|s^C\right\rangle\left\langle s^C\right|, \tag{4.4}$$

também conhecida como moeda de Grover de dimensão n — convém recordar que na Equação (2.21) já havíamos definido a moeda de Grover para dimensão 4. O operador de evolução, portanto, é dado por $U = S \circ (C_0 \otimes I_P)$, com o operador deslocamento descrito na Equação (2.34). Note que o estado diagonal $\left|s^C\right\rangle\left|s^P\right\rangle$ é um autovetor de U com autovalor 1, de modo que a caminhada de Grover no hipercubo mantém essa condição inicial inalterada. Em geral, os algoritmos de busca baseados em caminhadas quânticas necessitam que se faça uma modificação em uma caminhada quântica padrão. No entanto, a ordem do algoritmo de busca resultante dependerá do espectro de autovalores da caminhada original não-modificada.

Suponhamos que desejamos procurar o vértice v_0. O vértice precisa ser marcado de alguma forma, e isso é feito por meio de um operador moeda modificado. A atuação do novo operador moeda depende do vértice onde o caminhante está: se estiver no vértice procurado v_0, a moeda atua como o operador C_1; mas se ele estiver em qualquer outro vértice, atua como a moeda original C_0. A nova moeda, portanto, é definida como

$$C' = C_0 \otimes I_P + (C_1 - C_0) \otimes |v_0\rangle\langle v_0|. \tag{4.5}$$

Esta nova moeda define um novo operador de evolução dado por $U' = SC'$.

No algoritmo SKW, aplica-se ao vértice procurado a moeda $C_1 = -I_C$, enquanto aos demais vértices aplica-se a moeda de Grover definida na Equação (4.4). Substituindo esses valores na Equação (4.5), temos o operador evolução

$$\begin{aligned} U' =& S \circ (C_0 \otimes I_P) - S \circ ((C_0 + I_C) \otimes |v_0\rangle\langle v_0|) \\ =& S \circ (C_0 \otimes I_P) - 2S \circ (\left|s^C\right\rangle\left\langle s^C\right| \otimes |v_0\rangle\langle v_0|). \end{aligned} \tag{4.6}$$

Como $\left|s^C\right\rangle$ é autovetor de C_0 com autovalor 1, podemos reescrever a equação anterior,

$$\begin{aligned} U' =& S \circ (C_0 \otimes I_P) - 2S \circ (C_0 \left|s^C\right\rangle \left\langle s^C\right| \otimes |v_0\rangle \langle v_0|) \\ =& U - 2U(\left|s^C\right\rangle \left\langle s^C\right| \otimes |v_0\rangle \langle v_0|). \end{aligned} \tag{4.7}$$

Desse modo, temos,

$$U' = U(I - 2\left|s^C\right\rangle \left\langle s^C\right| \otimes |v_0\rangle \langle v_0|). \tag{4.8}$$

Portanto, vimos que o algoritmo SKW pode ser escrito como uma composição de dois operadores unitários, $U' = UR_{v_0}$, em que R_{v_0} é o operador de reflexão em torno do vértice marcado. Um algoritmo de busca que possa ser escrito nesta forma é um algoritmo abstrato de busca se U satisfizer algumas condições:[1] (1) U precisa ser uma matriz real, (2) U precisa ter um único autovetor com autovalor 1, e (3) este autovetor precisa ser o estado inicial do algoritmo. Usualmente, U é o operador de evolução de uma caminhada de Grover não-modificada no grafo onde se deseja efetuar a busca.

O algoritmo SKW é, portanto, um algoritmo abstrato de busca. Logo, podemos analisá-lo com auxílio da teoria desenvolvida para essa classe mais geral de algoritmos. Voltando à seção anterior, vemos que o algoritmo de Grover também é um algoritmo abstrato de busca, apesar de não ter sido desenvolvido originalmente como busca baseada em caminhadas quânticas.[2] No entanto, a maior motivação para o estudo do algoritmo abstrato de busca é o desenvolvimento de novos algoritmos quânticos baseados em caminhadas quânticas em grafos.

Utilizando o arcabouço matemático do algoritmo abstrato de busca, podemos calcular a complexidade do algoritmo SKW com base no problema de autovalores de U. Trataremos disso mais tarde. Por ora, basta sabermos que o algoritmo SKW

[1] Também pode-se admitir que essas propriedades sejam satisfeitas em um subespaço que seja preservado pelo operador U.

[2] O algoritmo de Grover pode ser formulado como um algoritmo abstrato de busca baseado na caminhada quântica em um grafo completo (Ambainis et al., 2005).

possui complexidade $O(\sqrt{2^n})$.

4.3 Algoritmo de Ambainis-Kempe-Rivosh

O algoritmo AKR efetua uma busca em uma malha d-dimensional $N^{1/d} \times N^{1/d} \times \cdots \times N^{1/d}$, com condições de contorno periódicas. Neste trabalho, vamos nos concentrar no caso bidimensional, ou seja, quando temos uma malha $\sqrt{N} \times \sqrt{N}$. A caminhada quântica nesta malha se passa em um espaço de Hilbert $\mathcal{H}_C \otimes \mathcal{H}_P$, em que $\mathcal{H}_C$ é o subespaço-moeda, de dimensão 4, e $\mathcal{H}_P$ é o subespaço-posição, de dimensão N. A base canônica para $\mathcal{H}_C$ é $\{|d, j\rangle\}$, para $0 \leq d, j \leq 1$. A variável d indica a direção do deslocamento ($d = 0$ para deslocamento horizontal e $d = 1$ para deslocamento vertical), enquanto a variável j indica o sentido do deslocamento ($j = 0$ indica que o caminhante se move para frente e $j = 1$ indica que o caminhante se move para trás). A base canônica para $\mathcal{H}_P$ é $\{|x, y\rangle\}$, em que $0 \leq x, y \leq \sqrt{N} - 1$.

O estado genérico do caminhante quântico na malha bidimensional $\sqrt{N} \times \sqrt{N}$ após t iterações é dado pela Equação (2.17), que reproduzimos aqui por comodidade,

$$|\Psi(t)\rangle = \sum_{d,j=0}^{1} \sum_{x,y=0}^{\sqrt{N}-1} \psi_{d,j;x,y}(t) \, |d, j\rangle \, |x, y\rangle \, , \tag{4.9}$$

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

A evolução do sistema ao longo do tempo é dada por um operador unitário $U' = SC'$, em que S é o operador deslocamento e C' é uma moeda modificada. O operador deslocamento é definido como

$$S = \sum_{j,k=0}^{1} \sum_{x,y=0}^{\sqrt{N}-1} |d, 1-j\rangle \langle d, j| \otimes \left|x + (-1)^j \delta_{d,0}, y + (-1)^j \delta_{d,1}\right\rangle \langle x, y| \, . \tag{4.10}$$

Note que este operador difere do apresentado na Equação (2.22) por haver uma inversão no sentido do movimento logo após a aplicação do operador deslocamento. Essa modificação na definição usual do operador deslocamento é essencial para melhoria da eficiência do algoritmo. Sem essa modificação, Ambainis et al. (2005)

mostraram que o algoritmo não teria nenhum ganho de complexidade em relação à busca clássica. O operador moeda para os sítios não-marcados é a moeda de Grover, definida na Equação (4.4). O operador moeda para o sítio marcado é $C_1 = -I$, como na seção anterior. O operador moeda modificado é, portanto,

$$C' = C_0 \otimes I_P - (I_C + C_0) \otimes |x_0, y_0\rangle \langle x_0, y_0| \tag{4.11}$$

em que (x_0, y_0) é o sítio marcado.

Também é possível verificar que o algoritmo AKR é um algoritmo abstrato de busca. Nesse caso, o operador de evolução deve ser reescrito como

$$U' = U(I - 2 \left|s^C\right\rangle \left\langle s^C\right| \otimes |x_0, y_0\rangle \langle x_0, y_0|), \tag{4.12}$$

em que U é o operador de evolução da caminhada não-modificada.

O estado inicial do algoritmo AKR é $|\Psi(0)\rangle = \left|s^C\right\rangle \left|s^P\right\rangle$. O operador U' deve ser aplicado $O(\sqrt{N \log N})$ vezes. Logo após o último passo, a sobreposição entre o estado final e o sítio marcado é $O(1/\sqrt{\log N})$. A fim de melhorar a probabilidade de encontrar o sítio marcado, são necessárias em média $O(\sqrt{\log N})$ repetições do algoritmo. Desse modo, a complexidade final do algoritmo AKR é $O(\sqrt{N} \log N)$.

4.4 Algoritmo de Tulsi

O algoritmo de Tulsi (2008) é uma melhoria sobre o algoritmo AKR. Ele adiciona um q-bit ao sistema, que serve para controlar as demais operações. A caminhada quântica, neste caso, se passa em um espaço de Hilbert $\mathcal{H}_b \otimes \mathcal{H}_C \otimes \mathcal{H}_P$, em que $\mathcal{H}_b$ é o espaço bidimensional do q-bit auxiliar, e $\mathcal{H}_C$ e $\mathcal{H}_P$ são os mesmos espaços descritos no algoritmo AKR.

O estado genérico do caminhante quântico para o algoritmo de Tulsi após t iterações é dado por

$$|\Psi(t)\rangle = \sum_{b,d,j=0}^{1} \sum_{x,y=0}^{\sqrt{N}-1} \psi_{b;d,j;x,y}(t) \, |b\rangle \, |d, j\rangle \, |x, y\rangle \, , \tag{4.13}$$

com $\psi_{b;d,j;x,y}(t) \in \mathbb{C}$ e $\sum_{b,d,j} \sum_{x,y} |\psi_{d,j;x,y}(t)|^2 = 1$. No algoritmo de Tulsi, o estado inicial é dado por $|\Psi(0)\rangle = |1\rangle \left|s^C\right\rangle \left|s^P\right\rangle$.

A evolução do sistema ao longo do tempo é formulada a partir da Equação (4.12), referente à evolução do algoritmo AKR. Passaremos a descrever as diferenças introduzidas pelo algoritmo de Tulsi. Inicialmente, vamos denotar por $R_{(x_0,y_0)}$ o operador de reflexão em torno do sítio marcado. Também vamos denotar por c_1U o operador U controlado pelo primeiro q-bit, e analogamente para $c_1R_{(x_0,y_0)}$. O operador controlado atua no q-bit alvo se e somente se o q-bit de controle for igual a $|1\rangle$. A operação controlada quântica é diferente da clássica quando o q-bit de controle está em superposição de estados. Maiores detalhes sobre as portas lógicas da computação quântica por ser encontradas no Apêndice A.

Vamos definir as operações unitárias sobre um q-bit,

$$X_\delta = \begin{pmatrix} \cos\delta & \sin\delta \\ -\sin\delta & \cos\delta \end{pmatrix} \tag{4.14}$$

e

$$\bar{Z} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{4.15}$$

No algoritmo AKR, a primeira operação seria a reflexão em torno do sítio marcado. Já no algoritmo de Tulsi, antes de aplicar o operador de reflexão controlado, é necessário efetuar uma pequena rotação no q-bit auxiliar por meio do operador X_δ. Depois da reflexão, aplica-se o inverso da rotação no q-bit auxiliar, por meio do operador $X_\delta^\dagger$. No algoritmo AKR, ficaria restando somente aplicar o operador da caminhada não-modificada. Já no algoritmo de Tulsi, a ação do operador de caminhada é controlada pelo primeiro q-bit, e depois depois dessa aplicação ainda resta realizar a operação $\bar{Z}$ no q-bit auxiliar.

Portanto, a evolução do algoritmo de Tulsi é dada pelo operador unitário

$$U' = (\bar{Z})_b \cdot c_1U \cdot (X_\delta^\dagger)_b \cdot c_1\bar{R}_{sc,m} \cdot (X_\delta)_b. \tag{4.16}$$

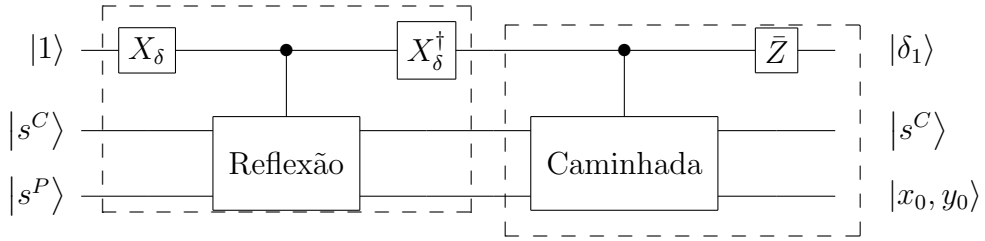

Figura 4.1: Circuito quântico para o algoritmo de Tulsi, mostrando apenas uma iteração, por simplicidade. As portas devem ser repetidas $O(\sqrt{N \log N})$ vezes para que o estado final indicado seja obtido.

Pode-se verificar que este operador está no formato do algoritmo abstrato de busca, pois $(X_\delta^\dagger)_b \cdot c_1 \bar{R}_{sc,m} \cdot (X_\delta)_b$ é uma reflexão em torno do estado procurado, e o operador $(\bar{Z})_b \cdot c_1 U \cdot (X_\delta^\dagger)_b$ satisfaz as três propriedades do operador de caminhada, descritas na seção anterior. O circuito na Figura 4.1 mostra a sequência de operações para o algoritmo de Tulsi. Note que δ é um parâmetro que pode ser escolhido antes do início da execução do algoritmo.

O estado marcado no algoritmo de Tulsi é $|t_\delta\rangle = |\delta_1\rangle \left|s^C\right\rangle \left|s^P\right\rangle$, em que $|\delta_1\rangle = -\sin\delta\,|0\rangle + \cos\delta\,|1\rangle$. Se escolhermos $\cos\delta = \Theta(\sqrt{1/\log N})$, o algoritmo requer $O(\sqrt{N \log N})$ iterações para alcançar o estado procurado, da mesma forma que o algoritmo AKR. No entanto, para esse mesmo valor de δ, a probabilidade de acerto do algoritmo é constante. Desse modo, a complexidade final do algoritmo de Tulsi é $O(\sqrt{N \log N})$, apresentando vantagem em relação ao algoritmo AKR.

No Apêndice B, apresentamos uma análise do algoritmo abstrato de busca, que generaliza os algoritmos descritos neste capítulo.

Capítulo 5

Simulações computacionais

Muitas vezes, o estudo analítico de certas propriedades das caminhadas quânticas pode ser uma tarefa altamente complexa, ou mesmo inviável. Nessas situações, a saída é realizar simulações computacionais da caminhada com diversos parâmetros de entrada e, com base dos dados obtidos, inferir as propriedades investigadas. Em certas situações, o estudo numérico de uma caminhada pode fornecer informações valiosas a fim de direcionar um futuro estudo analítico.

Neste capítulo apresentamos os resultados originais que obtivemos com base em simulações computacionais. Na Seção 5.1, descrevemos um simulador genérico de caminhadas quânticas para malhas unidimensionais e bidimensionais (Marquezino e Portugal, 2008), chamado QWalk. Na Seção 5.2, discutimos acerca da descoerência em algoritmos de busca baseados em caminhadas quânticas (Abal et al., 2009). As conclusões dessa seção foram obtidas a partir de simulações computacionais, ainda que o simulador QWalk não tenha sido utilizado diretamente. Na Seção 5.3, apresentamos algumas observações numéricas sobre o desempenho de algumas variantes do algoritmo abstrato de busca. Finalmente, na Seção 5.4, fazemos algumas discussões sobre os assuntos aqui tratados.

5.1 O simulador QWalk

Um simulador de caminhadas quânticas é muito importante para o desenvolvimento dessa área de pesquisa. Sem um simulador genérico, o esforço dos pesquisa-

dores é desviado para a implementação de simulações numéricas específicas, quando o foco deveria estar nos aspectos físicos e matemáticos da pesquisa. Nesta seção descreveremos o QWalk, um simulador de código-fonte aberto, multiplataforma, desenvolvido em linguagem C e distribuído com a licença de *software* livre GNU GPL. O simulador faz parte das contribuições originais desta tese (Marquezino e Portugal, 2008), sendo aplicado a caminhadas quânticas para malhas unidimensionais e bidimensionais. O simulador QWalk permite que a comunidade científica realize simulações importantes em caminhadas quânticas utilizando comandos simples e também facilita a geração de gráficos para visualizar os resultados. Em sua versão atual, o simulador QWalk pode reproduzir muitas das simulações presentes em artigos de pesquisa. Malhas bidimensionais finitas com topologias genéricas podem ser definidas e a descoerência pode ser simulada de dois modos diferentes: através de medições ou quebra de ligações da malha. Nesta seção usaremos exemplos para explicar o funcionamento do simulador e para mostrar alguns resultados da literatura que são facilmente reproduzidos pelo simulador.

O simulador QWalk consiste de três ferramentas: *qw1d* simula caminhadas quânticas em malhas unidimensionais; *qw2d*, em malhas bidimensionais; e *qwamplify* melhora a visualização dos gráficos gerados por *qw2d*, amplificando algumas regiões. A seguir, daremos alguns exemplos de resultados recentes da literatura, a fim de demonstrar o funcionamento do simulador. Outros exemplos também são disponibilizados com os arquivos descarregados. No Apêndice C encontram-se instruções de instalação e comandos adicionais do simulador.

5.1.1 Malhas bidimensionais

Para usar o *qw2d* é necessário escrever um arquivo de entrada em qualquer editor de texto ASCII — sem formatação. Esse arquivo de entrada consiste de palavras-chave que definem as opções de simulação. A maioria das palavras-chave importantes são explicadas através dos exemplos dessa seção. Aquelas que não são cobertas aqui, são discutidas no Apêndice C.

Após criar um arquivo de entrada — digamos que ele se chame *file.in* — basta que se digite *qw2d file.in* no prompt de comando do sistema operacional utilizado. Os resultados da simulação ficam armazenados em alguns arquivos de saída. Esses arquivos são descritos em detalhes no Apêndice C, e incluem arquivos para as amplitudes complexas da função de onda no final da simulação, arquivos para a distribuição de probabilidades final, para a distribuição estacionária, para estatísticas, para scripts de gnuplot e muitos outros.

5.1.2 Experimento de fenda dupla

Na Figura 5.1 nós temos o resultado de uma simulação do experimento de fenda dupla com caminhantes quânticos, reproduzindo alguns resultados obtidos recentemente por Oliveira et al. (2007). Esta simulação gastou menos de 2s em um Pentium IV 2.6GHz com 512MB de memória RAM, 512KB de memória cache e sistema operacional Linux. A fim de realizar essa simulação com *qw2d* o arquivo de entrada precisa ter as seguintes palavras-chave, todas em caixa alta:

```
BEGIN
 COIN HADAMARD
 STATE HADAMARD
 STEPS 100
 BLPERMANENT
 SCREEN 60 -100 60 100
 LATTYPE DIAGONAL
END
```

Não é necessário posicionar os comandos exatamente como no exemplo. O usuário pode pular linhas entre os diferentes comandos ou até mesmo escrever tudo em uma única linha. É importante, no entanto, manter todas as palavras-chave na seção principal do arquivo de entrada, delimitadas por uma palavra-chave `BEGIN` e uma palavra-chave `END`. Caso contrário, as palavras-chave serão interpretadas como comentários. Os gráficos da Figura 5.1 foram gerados pelo *gnuplot*, utilizando um script fornecido como saída do *qw2d*. A geração dos gráficos gastou cerca de 30s.

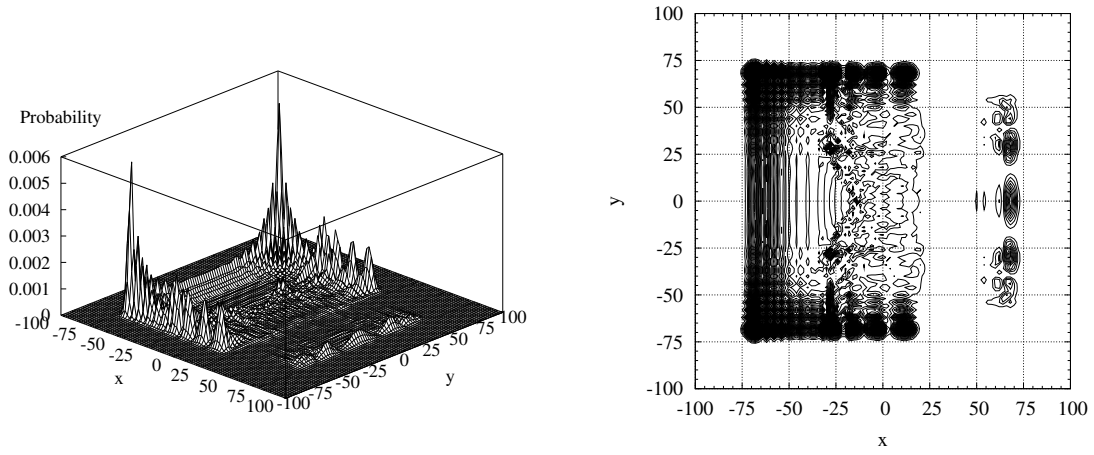

Figura 5.1: Distribuição de probabilidades após um experimento de fenda dupla. Um fator amplificação 5 foi usado para $x > 20$, a fim de melhorar a visualização. Esquerda: Gráfico 3D. Direita: Gráfico de contorno.

A palavra-chave `COIN` define a moeda usada na simulação. No exemplo anterior escolhemos a moeda de Hadamard. Poderíamos também ter escolhido as moedas de Fourier ou Grover com as opções `FOURIER` ou `GROVER` respectivamente. Também poderíamos ter escolhido uma moeda unitária arbitrária, usando a opção `CUSTOM`. Nesse caso, uma seção extra no arquivo de entrada seria requerida, a fim de definir a moeda arbitrária. Veja o Apêndice C.

Analogamente, a palavra-chave `STATE` define o estado inicial da simulação. No exemplo anterior nós escolhemos o estado que fornece um maior espalhamento para a moeda de Hadamard. Também poderíamos ter escolhido o estado inicial correspondente para as moedas de Grover ou Fourier, ou mesmo um estado inicial arbitrário.

A palavra-chave `STEPS` define o número de iterações que serão executadas pela simulação. No exemplo anterior o caminhante realizava cem passos de simulação. O usuário deve ter em mente que quanto maior o tempo de simulação em uma malha sem fronteiras, mais longe da origem a partícula irá se encontrar, o que significa que será necessário reservar um espaço maior na memória do computador para a simulação. Portanto, esta palavra-chave pode aumentar não somente o tempo de execução mas também a memória requerida pela simulação. Será expli-

cado mais adiante como evitar esse consumo excessivo de memória quando fixamos certas condições de contorno para a malha.

As fendas foram criadas com ajuda da palavra-chave `BLPERMANENT` na seção principal do arquivo de entrada. Com este comando, declaramos que algumas ligações na simulação serão interrompidas permanentemente. A fim de definir a posição destas ligações interrompidas, nós usamos uma seção separada no arquivo de entrada, com os comandos

```
BEGINBL
 LINE 20 100 20 7
 LINE 20 5   20 -5
 LINE 20 -7  20 -100
ENDBL
```

O comando `LINE x0 y0 x1 y1` isola todos os pontos que passam pelo segmento que vai de $(x0, y0)$ até $(x1, y1)$. A linha de pontos isolados pode ser paralela aos eixos x ou y, ou pode formar um ângulo de 45 graus com um destes. Também é possível isolar um único ponto com o comando `POINT x0 y0`. Usando combinações destes dois comandos é possível simular não somente experimentos de fenda única e fenda dupla, mas também a evolução do caminhante em uma enorme variedade de contornos. Neste primeiro exemplo, temos uma fenda em $(20, 6)$ e outra em $(20, -6)$.

Um anteparo de observação pode ser definido com a palavra-chave `SCREEN`, seguida das coordenadas inicial e final. O anteparo pode ser definido paralelo aos eixos x ou y, ou formando um ângulo de 45 graus com um destes. No exemplo o anteparo vai de $(60, -100)$ até $(60, 100)$.

Como no exemplo anterior somente uma fração muito pequena da onda passava pelas fendas, a visualização obtida inicialmente pela simulação era bastante precária. A fim de resolver isso, usamos a ferramenta *qwamplify* para amplificar por um fator 5 toda a região em que $x \geq 20$. Esta ferramenta pode ser usada digitando algo como `qwamplify file.dat [opções]` no prompt de comando. O programa cria uma cópia de segurança de `file.dat` e substitui o arquivo original

por um novo, com parte da função de onda amplificada. Ajuda em relação às opções disponíveis pode ser obtida simplesmente digitando `qwamplify` no prompt de comando.

O comando `LATTYPE DIAGONAL` declara que o operador de deslocamento utilizado na simulação é aquele definido pela Equação (2.22), ou seja,

$$S_a = \sum_{j,k=0}^{1} \sum_{x,y=-\infty}^{+\infty} |j,k\rangle \langle j,k| \otimes \left|x + (-1)^j, y + (-1)^k\right\rangle \langle x,y| \,. \tag{5.1}$$

O operador de deslocamento padrão, que também pode ser declarado explicitamente com o comando `LATTYPE NATURAL`, é aquele definido pela Equação (2.28), ou seja,

$$S_b = \sum_{j,d=0}^{1} \sum_{x,y=-\infty}^{+\infty} |j,d\rangle \langle j,d| \otimes \left|x + (-1)^j(1-\delta_{j,d}), y + (-1)^j\delta_{j,d}\right\rangle \langle x,y| \,. \tag{5.2}$$

A malha natural, quando comparada à malha diagonal, fornece distribuições de probabilidade rotacionadas de um ângulo de 45 graus. Podemos observar esse comportamento na Figura 5.2, referente à simulação de cem passos de um caminhante de Hadamard sem fendas. Note que há uma região inutilizada nos quatro cantos do gráfico. Na maioria das situações, a malha diagonal proporciona uma melhor visualização que a malha natural. Esse tipo de malha não-diagonal foi usado por Inui et al. (2004) com um operador de deslocamento ligeiramente diferente. Nossa equação de evolução, no entanto, possui a vantagem de preservar a distribuição de probabilidades final para um mesmo operador de moeda — a menos da rotação mencionada anteriormente.

Na Figura 5.3 temos os resultados de dois experimentos de fenda dupla com caminhantes de Hadamard. As fendas foram posicionadas exatamente como no exemplo anterior: uma em $(20, 6)$ e outra em $(20, -6)$. Na primeira simulação executamos $T = 100$ passos; na segunda simulação, $T = 800$ passos. A simulação levou cerca de 15min no segundo caso. Nos gráficos da Figura 5.3 temos o padrão que seria observado em anteparos posicionados, respectivamente, ao longo de

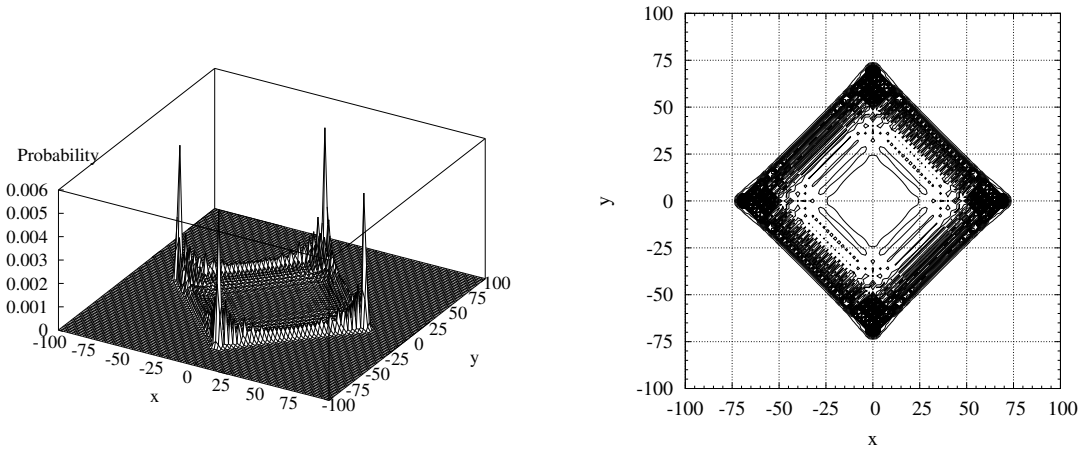

Figura 5.2: Distribuição de probabilidades após cem passos de um caminhante de Hadamard. Aqui, o operador de deslocamento é tal que a malha matemática coincide com a malha física. Esquerda: Gráfico 3D. Direita: Gráfico de contorno.

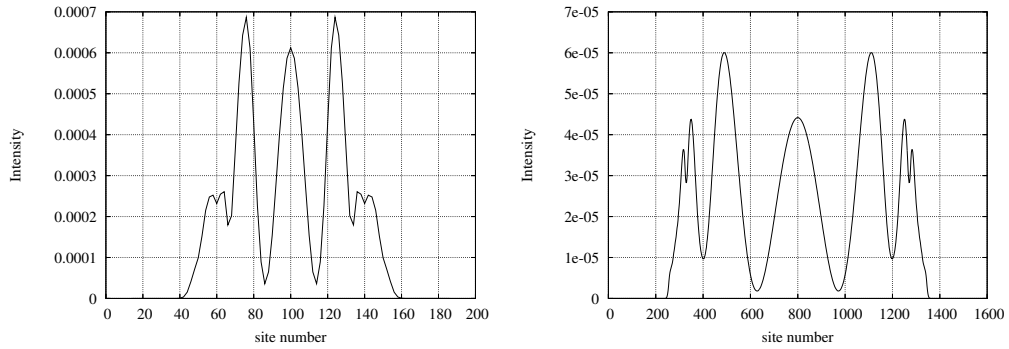

Figura 5.3: Simulação de anteparos de observação no experimento de fenda dupla. Esquerda: Simulação com $T = 100$ passos e anteparo ao longo de $x = 60$. Direita: Simulação com $T = 800$ passos e anteparo ao longo de $x = 500$.

$x = 60$ e $x = 500$. No segundo caso observamos que os mínimos locais são menores e a curva mais suave.

5.1.3 Detectores

Detectores são descritos por uma coleção $\{M_m\}$ de operadores de medição, os quais satisfazem a equação de completeza, $\sum_m M_m^\dagger M_m = I$. O índice m indica o resultado da medição. De acordo com o terceiro postulado da mecânica quântica, antes do estado $|\Psi\rangle$ ser medido, a probabilidade do resultado m ocorrer é dada por $p(m) = \langle\Psi| M_m^\dagger M_m |\Psi\rangle$, e após a medição o estado colapsa para $|\Psi'\rangle = \frac{1}{\sqrt{p(m)}} M_m |\Psi\rangle$. No *qw2d* podemos definir um número arbitrário de detectores especificando uma lista de coordenadas $(m_1, n_1), \cdots, (m_N, n_N)$. Os operadores de medição, portanto, são

$$M_0 = I_4 \otimes I_\infty - \sum_{i=1}^{N} M_i \qquad \text{e} \qquad M_i = I_4 \otimes |m_i, n_i\rangle \langle m_i, n_i| , \tag{5.3}$$

para $1 \leq i \leq N$.

Na Figura 5.4 temos os gráficos de outro experimento de dupla fenda. Nessa simulação, usamos a moeda de Grover e posicionamos a parede paralelamente à diagonal secundária. A parede vai de $(-60, -100)$ até $(100, 60)$. Uma fenda vai de $(13, -27)$ até $(15, -25)$ e a outra vai de $(25, -15)$ até $(27, -13)$. Nós também posicionamos um detector próximo à primeira fenda, em $(15, -27)$. O experimento

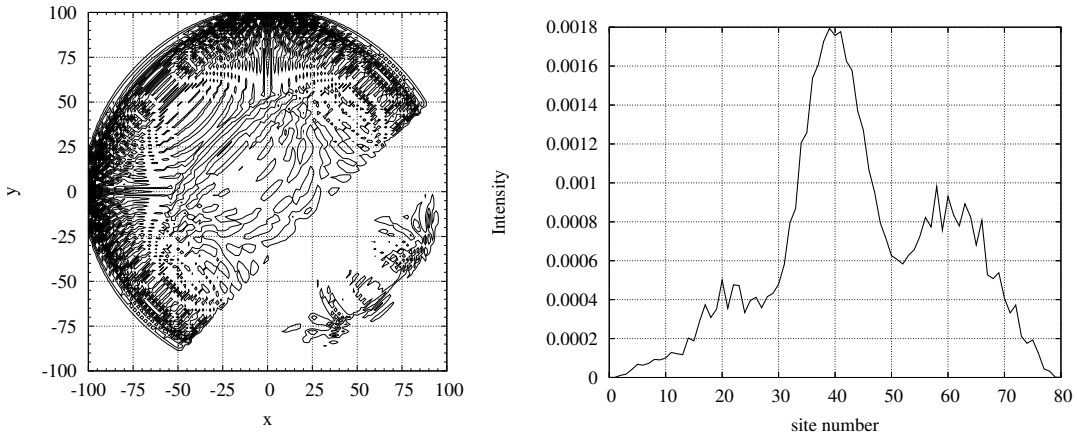

Figura 5.4: Resultados de um experimento de dupla fenda com caminhante de Grover. Tanto a parede como o anteparo estão paralelos à diagonal secundária e existe um detector próximo a uma das fendas. Esquerda: Gráfico de contorno da distribuição de probabilidade final. Direita: Simulação do anteparo.

foi repetido dez vezes a fim de tomar a média dos resultados.

As posições dos detectores são definidas na seção principal do arquivo de entrada, usando a palavra-chave `DETECTORS` seguida pelo número de detectores e suas respectivas coordenadas. O número de repetições do experimento é definida usando-se a palavra-chave `EXPERIMENTS`. Aqui temos um exemplo de como utilizar essas duas palavras-chave:

```
DETECTORS 1 15 -27

EXPERIMENTS 10
```

O anteparo, também posicionado paralelamente à diagonal secundária, vai de $(20, -100)$ até $(100, -20)$. O eixo x no painel direito da Figura 5.4 está numerado sequencialmente, começando pelo primeiro ponto do anteparo. Notamos em ambos os gráficos que os padrões de interferência são assimétricos e também mais fracos no lado do detector.

5.1.4 Malhas finitas

O simulador QWalk pode ser usado para simular caminhadas quânticas em malhas finitas. Nesta Seção estudamos o exemplo de uma malha quadrada, porém o QWalk também permite a definição de contornos arbitrários. No caso aqui considerado, os contornos foram gerados pela quebra das ligações sobre um quadrado

cujos vértices são $(-M, M)$, (M, M), $(M, -M)$ e $(-M, -M)$, como em Oliveira et al. (2006b). Diferentes valores de M foram investigados. A distribuição estacionária foi aproximada através da execução da simulação por $5 \cdot 10^3$ passos.

A fim de calcular a variação total da distância entre a distribuição de probabilidades média e a distribuição limite, precisamos da palavra-chave `MIXTIME` na seção principal do arquivo de entrada, seguido do número de passos usados para aproximar a distribuição estacionária. Esse número deve ser maior que — ou pelo menos igual a — o número de passos que serão simulados, e as posições das ligações interrompidas permanentes precisam ser corretamente declaradas de modo a definir uma região fechada da malha. Já que o número de passos simulados pode ser muito maior que o tamanho da malha obtida, podemos melhorar o desempenho da simulação usando a palavra-chave `LATTSIZE`. Essa palavra-chave declara que a malha somente é usada de $x = -max$ até $x = max$ e de $y = -max$ até $y = max$, em que max é o número inteiro passado como argumento da palavra-chave `LATTSIZE`. Essa opção reduz o consumo de memória e o tempo de processamento, devendo ser usada sempre que o caminhante é restrito a uma região finita da malha. A palavra-chave `LATTSIZE` precisa ser usada *depois* da palavra-chave `STEPS`.

Para o exemplo anterior, tomando $M = 60$, podemos preparar um arquivo de entrada com as palavras-chave

```
MIXTIME 5000

STEPS 2000

LATTSIZE 59
```

juntamente com as palavras-chave que declaram o contorno,

```
BEGINBL

 LINE -60 60 60 60

 LINE 60 60 60 -60

 LINE 60 -60 -60 -60

 LINE -60 -60 -60 60

ENDBL
```

Quando a opção `MIXTIME` é usada, a distribuição estacionária aproximada é obtida no começo da simulação. Depois disso, o arquivo de saída *.sta* registra a

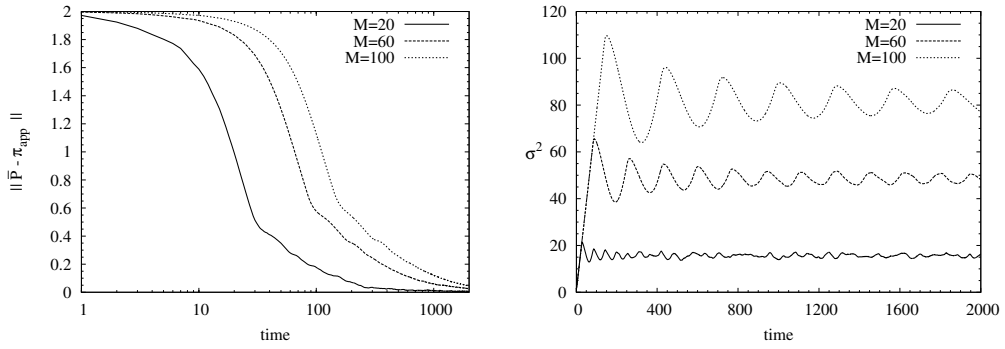

Figura 5.5: Variação total da distância entre a distribuição média e a distribuição estacionária aproximada em função do tempo (esquerda); e evolução do desvio padrão (direita). Ambos os gráficos são referentes à caminhada de Hadamard na malha diagonal e para diferentes tamanhos de caixas quadradas.

variação total da distância, a cada passo, entre a distribuição média e as distribuições estacionária e uniforme, de modo que o usuário possa facilmente gerar gráficos desta informação com auxílio de um *software* adequado, como o *gnuplot*. Em sua atual versão, o simulador calcula a distribuição uniforme sobre todos os sítios da malha matemática, sem levar em consideração o contorno em particular.

No painel esquerdo da Figura 5.5, temos a variação total da distância entre a distribuição média e a distribuição estacionária aproximada, em função do tempo, para o caminhante de Hadamard em uma malha finita com contornos retangulares, para diferentes valores de M. Notamos que a distância converge para zero, como esperado. No painel direito da Figura 5.5 temos a evolução do desvio padrão do caminhante de Hadamard na malha diagonal, para diferentes valores de M. O desvio padrão considerado é a soma do desvio da posição x com o desvio da posição y do caminhante. Notamos a oscilação do desvio padrão após o caminhante colidir com as paredes da caixa. No arquivo de saída *.sta* encontramos informação suficiente para gerar esse tipo de gráfico com apenas alguns poucos comandos do gnuplot ou qualquer outra ferramenta similar. Podemos notar que esses gráficos estão consistentes com os resultados obtidos por Oliveira et al. (2006b).

Na Figura 5.6, temos a distribuição estacionária aproximada para o caminhante de Hadamard dentro de uma caixa com $M = 60$. Pode-se notar que ela é visualmente diferente da distribuição uniforme. Além do mais, pode-se tam-

bém confirmar pelo arquivo de saída do QWalk, que a distribuição média não converge para a distribuição uniforme mesmo em tempos longos. Podemos notar semelhanças entre a distribuição estacionária da Figura 5.6, correspondente a um caminhante de Hadamard na caixa e condição inicial localizada, e a distribuição estacionária dada na Figura 3.8, referente a um caminhante de Hadamard com operador de deslocamento efetuando inversão na moeda e condição inicial distribuída uniformemente entre todos os vértices e todas as moedas.

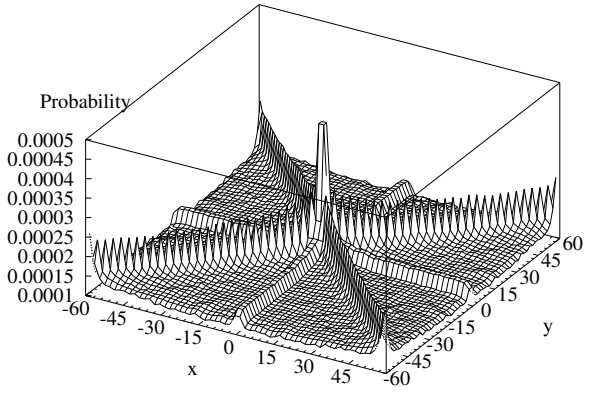

Figura 5.6: Distribuição estacionária aproximada com $5 \cdot 10^3$ passos em uma caixa com $M = 60$.

5.1.5 Descoerência

O simulador QWalk permite a simulação de caminhantes quânticos bidimensionais com duas fontes diferentes de descoerência. A primeira delas é gerada por meio de medições a partir de detectores dispostos aleatoriamente. Também é possível simular um tipo de ruído unitário (Shapira et al., 2003) que abre aleatoriamente ligações da malha. Este modelo de ruído já havia sido estudado para o caminhante quântico em uma linha (Romanelli et al., 2005) e em um plano (Oliveira et al., 2006a).

A descoerência de medições é melhor observada na simulação de malhas finitas. A fim de declarar a probabilidade de medir cada sítio, usamos a palavra-chave `DTPROB` na seção principal do arquivo de entrada, seguida de um valor numérico.

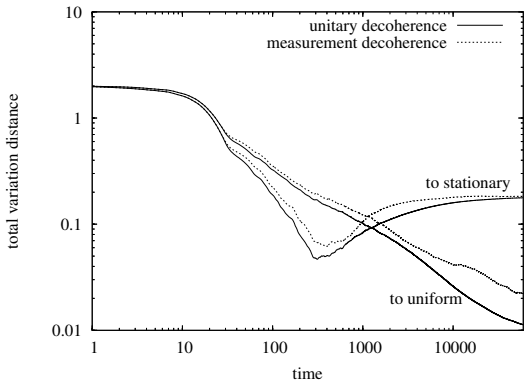

Figura 5.7: Variação total da distância, em função do tempo, da distribuição média da caminhada bidimensional coerente para as distribuições uniforme e estacionária coerente. Duas fontes de ruído são comparadas. Foi usada a moeda de Hadamard e a distribuição estacionária foi aproximada com $7 \cdot 10^4$ passos.

Na Figura 5.7 nós comparamos dois tipos de ruídos — medições e ligações interrompidas — com o mesmo parâmetro de descoerência $p = 10^{-2}$ e o mesmo tamanho de caixa $M = 20$. Em ambos os casos o experimento foi repetido dez vezes com a moeda de Hadamard a fim de tomar a média dos resultados. Nas linhas pontilhadas temos os resultados para a descoerência de medições e nas linhas cheias temos os resultados para a descoerência unitária. Em ambos os casos temos a variação total da distância, em função do tempo, entre a distribuição média e as distribuições uniforme e a estacionária coerente. A distribuição estacionária foi aproximada com $7 \cdot 10^4$ passos. Notamos que em ambos os casos a distribuição média inicialmente se aproxima da distribuição estacionária coerente até $\sim 1/p$ passos, indo para a distribuição uniforme após esse tempo.

Na Figura 5.8 nós temos o resultado de uma simulação de uma caminhada de Fourier com descoerência unitária assimétrica. Os resultados são similares aos obtidos por Oliveira et al. (2006a). Nós declaramos uma probabilidade $p_0 = 0$ de ligações interrompidas na diagonal secundária, e uma probabilidade de $p_1 = 0.2$ de ligações interrompidas na diagonal principal. Este tipo de simulação pode ser feita com a palavra-chave `BLPROB`, a qual deve ser seguida pelas probabilidades de ligações interrompidas em cada direção — nas diagonais secundária e principal, quando a malha diagonal é usada; e nas direções horizontal e vertical, quando a

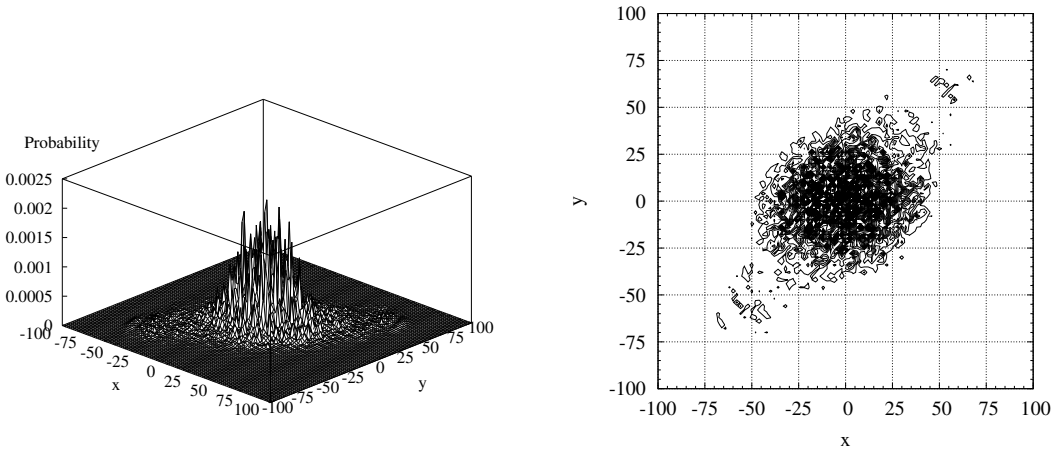

Figura 5.8: Distribuição de probabilidades após cem passos de um caminhante de Fourier descoerente na malha diagonal. A probabilidade de ligações interrompidas foi assimétrica, a saber, $p_0 = 0$ na diagonal secundária e $p_1 = 0.2$ na diagonal principal. Esquerda: Gráfico 3D. Direita: Gráfico de contorno.

malha natural é usada.

5.1.6 Malhas unidimensionais

O uso do *qw1d* é análogo ao uso do *qw2d*, exceto pela ausência de algumas palavras-chave. A saber, *qw1d* não reconhece as palavras-chave `SCREEN` e `BLPERMANENT` e não reconhece as sub-opções `FOURIER` e `GROVER` para moedas e condições iniciais. Ele também não reconhece as sub-opções `DIAGONAL` ou `NATURAL` para a palavra-chave `LATTYPE`. Por outro lado, *qw1d* funciona com três tipos de malha, selecionados com a palavra-chave `LATTYPE`: a reta infinita é selecionada pela sub-opção `LINE`; a malha finita unidimensional com condições de contorno reflectivas é selecionada pela sub-opção `SEGMENT`; e o ciclo é selecionado pela sub-opção `CYCLE`. Informação adicional sobre o uso do *qw1d* pode ser encontrada juntamente com o código-fonte ou com os arquivos binários distribuídos na Internet.

Na Figura 5.9 nós temos a distribuição de probabilidades para um caminhante de Hadamard em uma malha unidimensional infinita (Kendon e Tregenna, 2003; Ambainis et al., 2001; Konno, 2002; Nayak e Vishwanath, 2000). Comparamos o caso da evolução coerente com um descoerente, no qual a descoerência é introduzida por ligações interrompidas aleatoriamente. Poderíamos ter introduzido

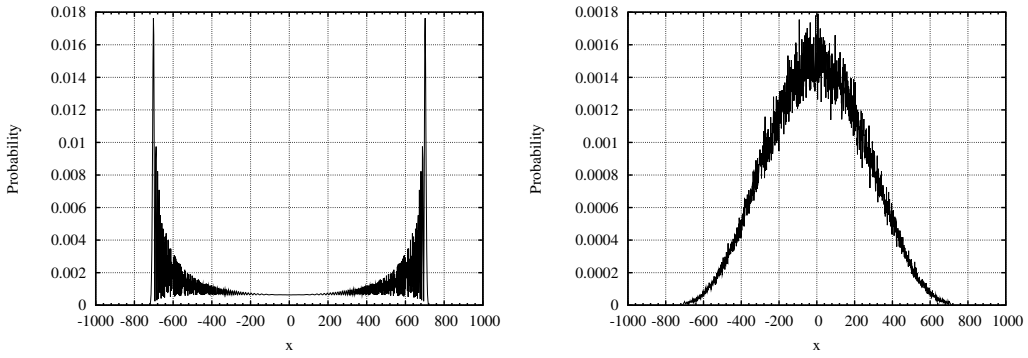

Figura 5.9: Caminhada quântica em malha unidimensional com ligações interrompidas. Esquerda: Simulação com $T = 10^3$ passos e $p = 0$. Direita: Simulação com $T = 10^3$ passos e $p = 10^{-2}$, tomando a média sobre cem experimentos.

descoerência através de medições aleatórias também. A probabilidade de ligações interrompidas no exemplo é $p = 10^{-2}$. Tanto no caso coerente como no caso descoerente, nós executamos $T = 10^3$ passos com *qw1d*. No caso descoerente, tomamos como resultado a média sobre cem experimentos independentes. Notamos pela figura que, devido ao número de passos ser muito maior que $1/p$, o comportamento clássico do caminhante quântico já emergiu (Romanelli et al., 2005).

Na Figura 5.10 temos o resultado de uma caminhada de Hadamard sobre o ciclo (Kendon, 2007; Kendon e Tregenna, 2003) com cem sítios, após $T = 2 \cdot 10^4$ passos. Aharonov et al. (2001) provaram que a caminhada quântica no ciclo com um número ímpar de sítios converge para a distribuição uniforme. O mesmo resultado não vale em geral, entretanto, para ciclos com número par de sítios, como podemos ver no painel direito da Figura 5.10. Nesse gráfico, a distribuição estacionária foi aproximada com um número grande de passos. Também é possível confirmar essa observação usando a informação gerada pelo QWalk a fim de visualizar o gráfico da variação total da distância, em cada passo, para as distribuições uniforme e estacionária aproximada. O gráfico mostraria que esta converge para zero, enquanto aquela não.

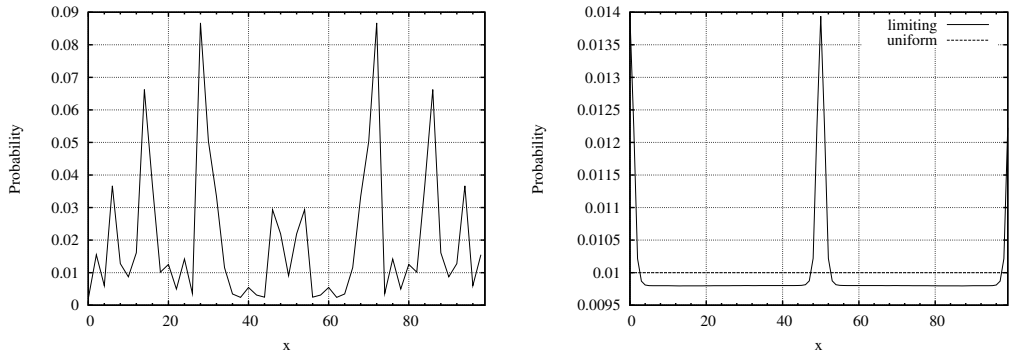

Figura 5.10: Caminhada quântica no ciclo com cem sítios. Esquerda: Distribuição de probabilidades final, após $T = 2 \cdot 10^4$ passos. Direita: Distribuição estacionária aproximada com $T = 10^5$ passos.

5.2 Descoerência em algoritmos de busca em grafos

Uma abordagem para lidar com o problema da descoerência e das imperfeições das portas lógicas em algoritmos quânticos, com aumento significativo nos recursos requisitados pelo sistema, consiste em usar codificação redundante e diversos níveis de códigos de correção de erros. Outra abordagem consiste em desenvolver algoritmos que são resistentes a certos tipos de erros que podem ser dominantes em dada implementação. Isto requer um conhecimento detalhado do efeito que diversos tipos de ruído têm sobre o desempenho do algoritmo. Parece provável que um computador quântico real irá se aproveitar de ambas as abordagens.

Os resultados desta seção já não são mais referentes ao simulador QWalk, ainda que a base das implementações utilizadas tenham aproveitado seu código. O conteúdo desta seção foi publicado inicialmente em (Abal et al., 2009) e faz parte das contribuições originais desta tese.

5.2.1 Modelos de descoerência

Em implementações físicas reais, operadores são sujeitos a descoerência. É importante determinar o grau de resistência a erros que um algoritmo possui em determinadas implementações. Em algoritmos quânticos de busca, existem três operadores: a moeda original (C), a moeda usada no vértice marcado ($-I$) e o operador de deslocamento (S). No Capítulo 4, em especial na Seção 4.2, encontram-se

informações mais detalhadas sobre a definição desses operadores nos algoritmos quânticos de busca baseados em caminhadas quânticas. Passaremos a analisar o impacto no desempenho do algoritmo de busca causado por erros em cada um desses três operadores.

Erros de fase no operador de moeda que é aplicado ao vértice marcado podem ser implementados substituindo-se $\mathcal{I}$ por

$$\widetilde{C}_{v_0}(\theta) = e^{i(\pi+\theta)} I, \tag{5.4}$$

com $\theta \in [-\pi, \pi]$. O operador de moeda ideal $-I$ no vértice marcado é recuperado para $\theta = 0$. Dizemos que o erro é sistemático quando o erro de fase θ é constante em cada passo da simulação (modelo I), e dizemos que o erro é aleatório quando o erro de fase θ em cada passo da simulação é uma variável aleatória Gaussiana com média zero e desvio padrão σ (modelo II).

Erros de fase no operador de moeda que é aplicado aos vértices não-marcados podem ser implementados reescrevendo-se C como

$$\widetilde{C}(\theta) = I - (1 - e^{i(\pi+\theta)}) \left|s\right\rangle \left\langle s\right| \tag{5.5}$$

com $\theta \in [-\pi, \pi]$. O operador de moeda de Grover é recuperado para $\theta = 0$.

O efeito de erros de fase no algoritmo de Grover original foi analisado por Long et al. (2000) e posteriormente por Shenvi et al. (2003a). Os autores deste segundo trabalho ainda investigaram a importância do crescimento dos erros de fase com o tamanho N do banco de dados. Em um trabalho recente, Li et al. (2006) estudaram o efeito de um operador C imperfeito no algoritmo SKW. Os operadores $-I$ (atuando no vértice marcado) e S foram considerados sem erros.

Erros no operador de deslocamento S podem ser implementados abrindo-se aleatoriamente, com probabilidade p por unidade de tempo, algumas ligações entre vértices conectados (modelo III). Este modelo de ruído de ligação interrompida foi considerado anteriormente para um caminhante quântico na reta e no plano (Ro-

Tabela 5.1: Modelos de descoerência para algoritmos de busca em grafos, estudados em trabalhos recentes da literatura.

	Modelo		
	I	II	III
Tipo de erro	sistemático	aleatório	topológico
Moeda marcada	Abal et al. (2009)	Abal et al. (2009)	N/A
Moeda não-marcada	Li et al. (2006)	Li et al. (2006)	N/A
Operador deslocamento	N/A	N/A	Abal et al. (2009)

manelli et al., 2005; Oliveira et al., 2006a), e também no hipercubo (Marquezino et al., 2008). Para implementar este tipo de erro, generalizamos o operador de deslocamento S de tal forma que nenhum fluxo de probabilidade é transferido através de ligações interrompidas. Este operador de deslocamento modificado é unitário para qualquer número de ligações interrompidas na malha. A cada passo da simulação, a topologia do grafo é definida, abrindo cada ligação com probabilidade p e realizando o deslocamento para o vértice vizinho somente se a ligação não estiver interrompida. O operador S original é recuperado para $p = 0$. Uma descrição mais detalhada desse modelo pode ser encontrada nas Seções 2.5 e 3.1.4.

5.2.2 Resultados para o algoritmo SKW

A Figura 5.11 mostra a probabilidade de se encontrar o caminhante no vértice marcado em função do número de passos, para cada modelo de ruído. No painel esquerdo, comparamos os resultados para o caso ideal, sem ruído, com aqueles para os modelos de ruído I e II. No painel direito, comparamos o caso ideal com a evolução segundo o modelo de ruído III. Todos os gráficos correspondem a um hipercubo de dimensão $n = 8$. Para o modelo I, tomamos um erro de fase $\theta = 0.3$; o desvio padrão no modelo II foi $\sigma = 0.3$; e a probabilidade de ligações interrompidas por unidade de tempo no modelo III foi $p = 0.02$. Note que o máximo da probabilidade no vértice marcado para o erro sistemático (modelo I), ocorre mais cedo que no caso ideal, sem ruído. O comportamento do algoritmo com ruído do

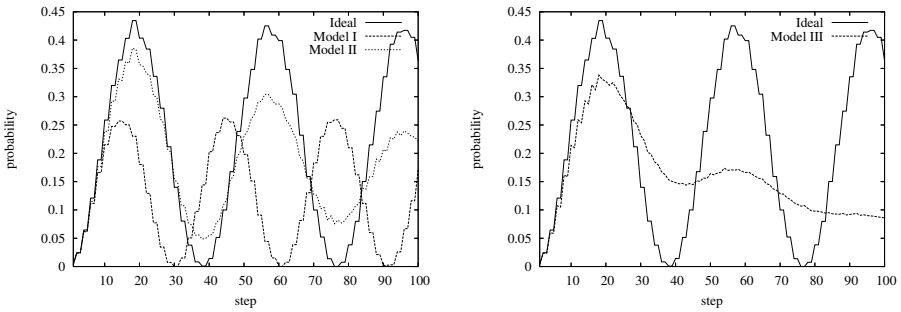

Figura 5.11: Esquerda: probabilidade no vértice marcado em função do número de passos s, comparando o caso ideal com os modelos de erro sistemático ($\theta = 0.3$) e aleatório ($\sigma = 0.3$). Direita: o mesmo para erros de ligação interrompida com $p = 0.02$.

modelo I afetando o operador de moeda no vértice marcado ($-I$) é semelhante ao observado em Li et al. (2006), no qual o operador C foi afetado. Apesar dos modelos aleatórios II e III corresponderem a diferentes tipos de ruído, eles resultaram em padrões bastante semelhantes. Em ambos os casos o primeiro máximo local na probabilidade ocorre aproximadamente no mesmo número de passos que no caso sem ruído, e atinge um valor mais baixo. Máximos locais subsequentes sofrem uma atenuação gradativa com o número de passos s.

O tempo de parada no caso sem ruído corresponde ao primeiro máximo, e para $n = 8$ este ponto é $\frac{\pi}{2}\sqrt{2^{n-1}} \approx 18$. Os gráficos da Figura 5.11 sugerem que na presença de ruído possa ser vantajoso parar o algoritmo antes desse ponto e repetí-lo outras vezes a fim de encontrar o resultado correto. Para verificar a eficácia dessa estratégia, devemos analisar o custo total do algoritmo, levando em consideração as repetições que devem ser realizadas a fim de amplificar a probabilidade de sucesso. Se a probabilidade de obter o resultado correto em uma rodada é p, então o número esperado de tentativas necessárias é $1/p$. Se a complexidade computacional para uma execução do algoritmo é $O(\sqrt{N})$, então a complexidade total é $O(\sqrt{N}/p)$. Note que se p não depender de N, ele não irá mudar a complexidade. Vamos definir o custo algorítmico $c(s)$ como o número total de passos necessários para

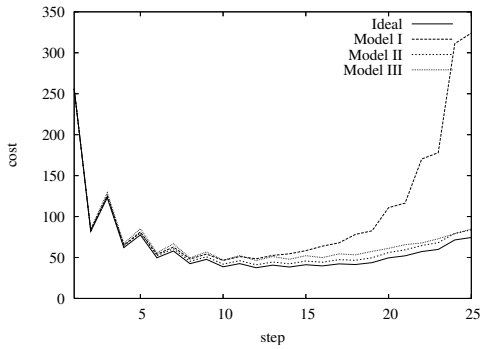

Figura 5.12: Custo $c(s)$, da Equação (5.6), versus número de passos, para o algoritmo de busca sem ruído e para o algoritmo com os três tipos de modelos de ruídos descritos no texto. O hipercubo considerado foi de dimensão $n = 8$.

encontrar o vértice procurado. Portanto,

$$c(s) = \frac{s}{p_s}, \tag{5.6}$$

em que s é o número de passos executados antes de efetuar a medição em uma rodada do algoritmo, e p_s é a probabilidade do caminhante ser encontrado no vértice procurado no instante s. Na Figura 5.12, mostramos a função custo $c(s)$ para diferentes modelos de ruído. No caso de um erro de fase sistemático, a função custo tem um mínimo bem definido em $s \approx 10$. Nosso trabalho mostra que é mais conveniente parar o algoritmo antes do instante de probabilidade máxima, tanto no caso ideal como no caso com ruído. Para os outros modelos de ruído, e também no caso sem ruído, a função custo tem um mínimo muito raso e cresce muito lentamente com o número de passos após este mínimo. No entanto, mesmo nestes casos, os resultados sugerem que seja vantajoso parar o algoritmo antes que o máximo de probabilidade do caso ideal seja alcançado, e então repetir o algoritmo mais vezes se necessário.

Na Figura 5.13, nas três curvas superiores, observamos a probabilidade de obtenção do vértice procurado em função do nível de ruído. No painel esquerdo (direito), temos os resultados para o modelo I (modelo II). Note que a fase ótima é $\theta = 0$ ou $\sigma = 0$, ou seja, quando $-I$ é usado no vértice marcado, do mesmo modo que no algoritmo padrão. O gráfico também mostra que o algoritmo é muito

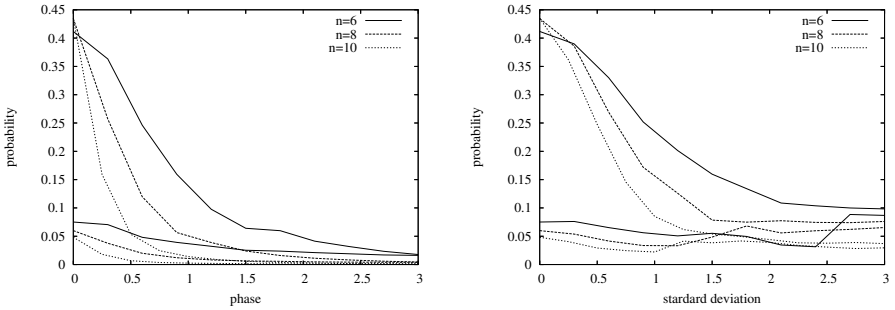

Figura 5.13: Esquerda: resultados para modelo I. Direita: resultados para o modelo II. Três curvas superiores: probabilidade máxima no vértice marcado em função do parâmetro de erro, para três valores de dimensão n do hipercubo. Três curvas inferiores: probabilidade máxima nos vértices não-marcados, usando a mesma convenção para a dependência da dimensão do hipercubo.

sensível a ruído de erros operacionais, se nenhum código de correção de erros é utilizado. As três curvas inferiores representam a maior probabilidade dentre os vértices não-marcados. Observamos que, conforme o erro de fase aumenta, a probabilidades no vértice marcado torna-se muito próxima da maior probabilidade nos vértices não-marcados. Neste caso não se pode distinguir entre a solução correta e, portanto, o algoritmo não é mais útil.

Nosso trabalho mostra que o ruído gerado por erros sistemáticos (painel esquerdo) parece ter papel mais significativo no algoritmo que o ruído gerado por erros aleatórios (painel direito).

Na Figura 5.14, temos os resultados para o modelo III. No painel esquerdo, nas três curvas superiores, observamos a probabilidade máxima no vértice marcado em função da taxa p de ligações interrompidas. As três curvas inferiores representam a probabilidade máxima dentre os vértices não marcados. No painel direito, temos a probabilidade máxima no vértice marcado em função da dimensão n do hipercubo. Neste caso, as probabilidades decaem conforme a dimensão do hipercubo aumenta, de modo semelhante ao resultado obtido por Li et al. (2006) para ruído afetando o operador de moeda de sítios não marcados.

A fim de estimar como erros do modelo II mudam a complexidade do algoritmo, usamos uma fórmula que cresce com N na forma $\theta = 1/N^{\delta}$, e aplicamo-la à Equação (5.4). Na Figura 5.15 temos o gráfico do custo escalado — o logaritmo na

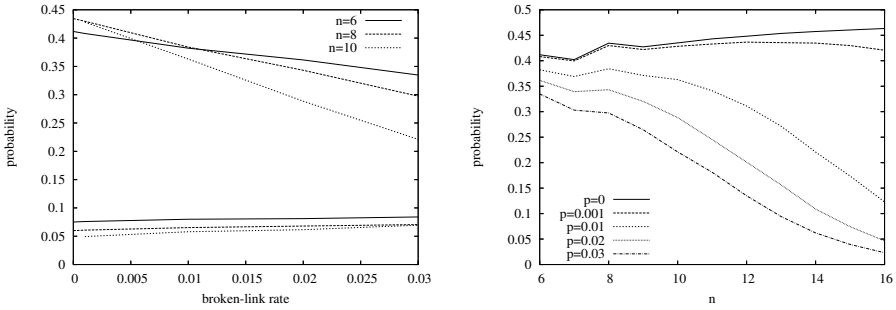

Figura 5.14: Resultados para o modelo III. Esquerda: análogo à Figura 5.13, porém como função da taxa de ligações interrompidas. Direita: probabilidade máxima no vértice marcado em função da dimensão n do hipercubo.

base N do custo algorítmico, dado pela Equação (5.6) — versus parâmetro de erro δ, para diversos valores de n. Convém lembrar que a complexidade do algoritmo SKW é $O(N^{0.5})$ e sua probabilidade de sucesso é $1/2 - O(1/n)$. Portanto, para valores maiores de δ e N, devíamos obter um custo escalado próximo a 0.5, correspondendo à complexidade do algoritmo SKW sem ruído. Nosso gráfico mostra um custo escalado próximo a 0.6, que é consistente com os valores de N considerados. Isto significa que para $\delta \geq 1$, o algoritmo SKW com com ruído possui mesma complexidade que o algoritmo SKW sem ruído. Para $\delta < 1$, a taxa de ruído cresce e o algoritmo gradativamente perde eficiência em relação à busca sem ruído. Para $\delta \approx -0.1$ o custo escalado é próximo a 1, o que significa que o algoritmo quântico possui a mesma complexidade da busca clássica, $O(N)$. Para $\delta < -0.1$ o custo escalado é maior que 1, o que significa que o desempenho da busca quântica é pior que o da busca clássica.

5.2.3 Resultados para o algoritmo AKR

O comportamento da probabilidade máxima no vértice marcado no algoritmo AKR segue um padrão semelhante ao do algoritmo SKW. A diferença principal é que para o algoritmo AKR, a probabilidade máxima diminui conforme N aumenta, enquanto para o algoritmo SKW a probabilidade máxima permanece próxima a 1/2. Os resultados numéricos para o custo no algoritmo AKR também mostram que na presença de ruído, é melhor parar o algoritmo antes antes do

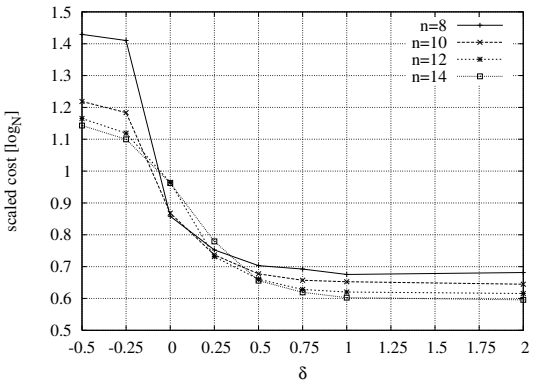

Figura 5.15: Logaritmo (base N) do custo algorítmico em função do parâmetro δ para o modelo II, comparando diferentes dimensões.

tempo de parada esperado do caso ideal.

A Figura 5.16 mostra a probabilidade máxima em função do nível de ruído. No painel esquerdo (direito) nós temos os resultados para o modelo I (modelo II). Esta figura deve ser comparada à Fig. 5.13. Note que o número de vértices das malhas correspondem ao número de vértices dos hipercubos. As curvas para o algoritmo AKR são muito similares àquelas para o algoritmo SKW e podemos tirar conclusões semelhantes para ambos os casos. A principal diferença é a distância entre as curvas, uma consequência do fato de que no algoritmo AKR a probabilidade máxima no vértice marcado diminui quando aumentamos N.

Na Figura 5.17, mostramos os resultados para o modelo III. No painel esquerdo, observamos a probabilidade máxima no vértice marcado, em função da taxa de ligações interrompidas p. No painel direito, o eixo horizontal está em escala logarítmica. Estes resultados devem ser comparados com a Figura 5.14. A probabilidade cai mais rapidamente no algoritmo AKR que no algoritmo SKW. Isto foi previsto no trabalho de Ambainis et al. (2005), no qual foi mostrado que a probabilidade no vértice marcado cresce como $O(1/\sqrt{\log N})$.

Na Figura 5.18 temos o gráfico do custo escalado $\log_N c(s)$ em função do parâmetro de erro δ. Esse gráfico é o análogo da Figura 5.15 para o algoritmo AKR. Convém lembrar que o custo no algoritmo AKR é $O(N^{0.5} \log N)$. Portanto, para valores maiores de δ e N, devemos encontrar um custo escalado um pouco acima

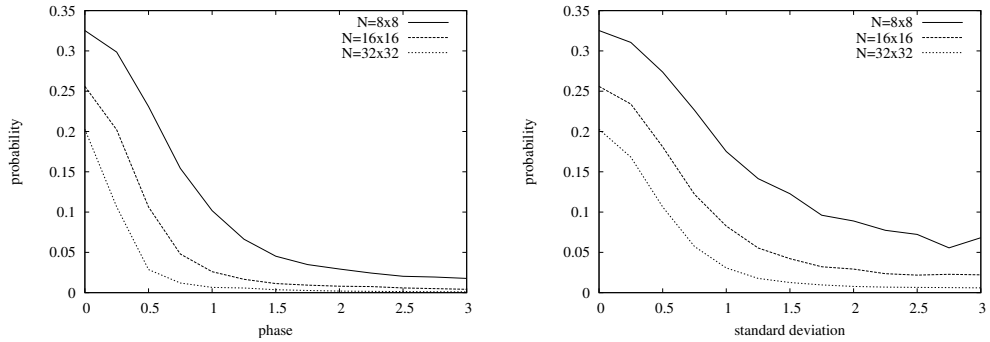

Figura 5.16: Esquerda: resultados para o modelo I. Direita: resultados para o modelo II. Probabilidade máxima no vértice marcado em função do nível de ruído para três valores de dimensão da malha.

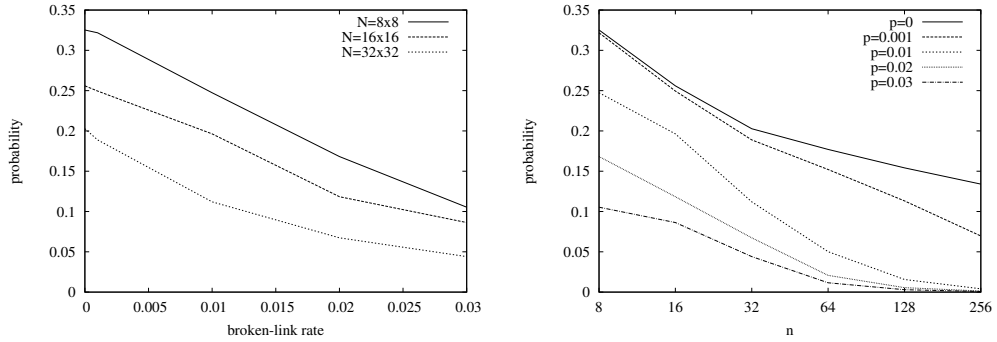

Figura 5.17: Resultados para o modelo III. Esquerda: probabilidade máxima no vértice marcado em função da taxa de ligações interrompidas. Direita: probabilidade máxima no vértice marcado em função da dimensão $\log_2 N$ de uma malha $\sqrt{N} \times \sqrt{N}$.

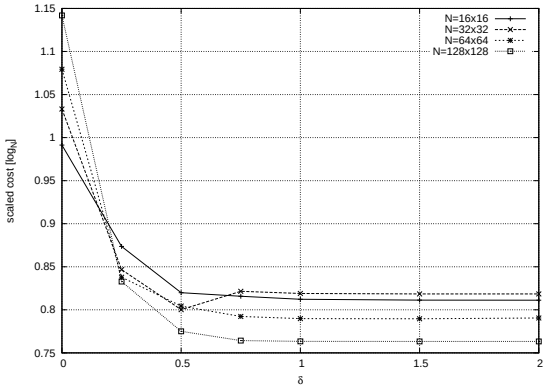

Figura 5.18: Logaritmo (base N) do custo algorítmico em função do parâmetro δ para o modelo II, comparando diferentes dimensões no algoritmo AKR.

de 0.5, correspondendo à complexidade do algoritmo AKR sem ruído. O custo escalado não é exatamente a potência de N devido à expressão do custo possuir o termo $\log N$. Nosso gráfico mostra um custo escalado próximo a 0.8, o que é consistente com os valores de N considerados. A partir da figura podemos observar que para $\delta \geq 1/2$, o algoritmo AKR com erro possui a mesma complexidade que o algoritmo sem ruído. Para $\delta < 1/2$, a taxa de ruído aumenta rápido o suficiente para que o algoritmo perca sua eficiência em relação à busca sem ruído. Quando diminuímos δ, o custo escalado aproxima-se de 1, significando que o algoritmo quântico passa a ter a mesma complexidade do algoritmo clássico, $O(N)$. Para $\delta < 0$, o custo escalado é maior que 1, significando que o desempenho da busca quântica é pior que o algoritmo clássico. Observa-se que $\delta = 1/2$ é o ponto de transição no algoritmo AKR, enquanto $\delta = 1$ é o ponto de transição no algoritmo SKW. Para comparação, note que Shenvi et al. (2003a) obtiveram $\delta = 1/4$ como ponto de transição no algoritmo de Grover.

5.3 Algoritmo de busca com redução de chamadas ao oráculo

Em nosso trabalho realizamos simulações numéricas do algoritmo AKR para a malha bidimensional e constatamos que este pode ser acelerados por meio de uma redução no número de aplicações do oráculo (Figura 5.19). O método consiste em aplicar o operador marcado um número reduzido de vezes — por exemplo, metade

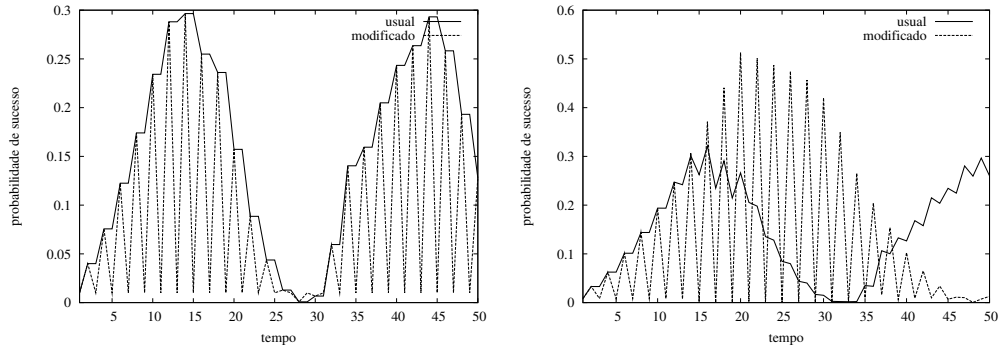

Figura 5.19: Probabilidade de sucesso em função do número de aplicações do operador de evolução, na malha 10×10 (esquerda) e na malha 11×11 (direita).

— intercalando-o com o operador original da caminhada no grafo. A aceleração alcançada por nosso método consiste em um fator constante, e que portanto não muda a classe de complexidade do algoritmo. Mesmo assim, além de representar um pequeno ganho no tempo de execução do algoritmo, também é possível que a redução no número de vezes em que o oráculo é utilizado torne o algoritmo mais robusto na presença de certos tipos de ruído.

Na Figura 5.19 temos a probabilidade de sucesso em função do número de aplicações do operador de evolução, comparando o algoritmo usual com o modificado — em que o operador marcado foi aplicado metade das vezes. Pode-se notar que, apesar da probabilidade de sucesso ser muito baixa em instantes ímpares, nos demais instantes ela fica próxima à obtida pelo algoritmo usual ou até maior. Nota-se também que o comportamento do algoritmo modificado depende fortemente da paridade da malha.

Outras estratégias de evolução também resultam em aceleração do algoritmo. Pode-se, por exemplo, aplicar o operador não-marcado uma vez para cada três aplicações do operador marcado. Na Figura 5.20, temos o custo do algoritmo modificado comparado ao algoritmo usual para malhas com número par de vértices por lado. O custo é definido como $c(N) = \frac{T_N}{p_N}$, em que T_N é o número de aplicações do oráculo até que a probabilidade de sucesso atinja o máximo, e p_N é a probabilidade de sucesso máxima.

No entanto, nem todas as estratégias de evolução funcionam. Se aplicarmos,

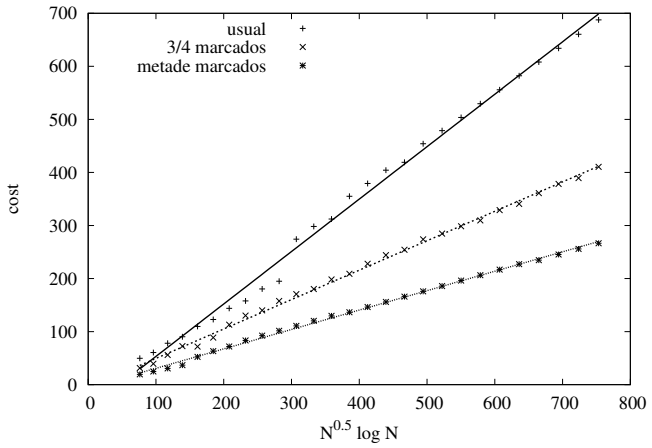

Figura 5.20: Custo do algoritmo modificado comparado ao custo do algoritmo usual.

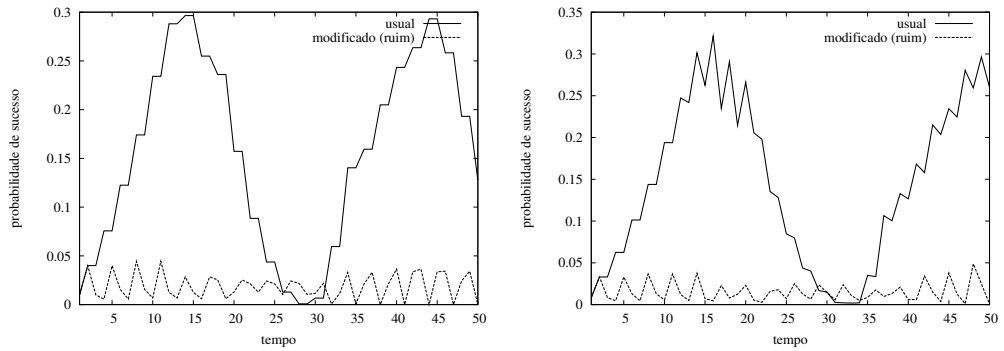

Figura 5.21: Probabilidade de sucesso em função do número de aplicações do operador de evolução, na malha 10×10 (esquerda) e na malha 11×11 (direita).

por exemplo, o operador marcado uma vez para cada duas aplicações do operador não-marcado, a probabilidade de medição do vértice buscado não é amplificada (Figura 5.21).

5.4 Discussões

Neste capítulo, inicialmente apresentamos o simulador QWalk e descrevemos seu uso. O QWalk é um simulador para malhes uni- e bidimensionais. Mostramos, através de exemplos, como o simulador pode ser usado para simular experimentos de dupla fenda, caminhantes em malhas finitas, detectores e dois tipos diferentes de descoerência. Três tipos de malha são disponíveis para caminhadas unidimensionais e três tipos são disponíveis para caminhadas bidimensionais.

As simulações apresentadas neste capítulo correspondem a trabalhos recentes em caminhadas quânticas e foram realizadas com comandos bastante simples do simulador QWalk. Além de haver simulado os resultados disponíveis da literatura com grande precisão, o simulador também é uma ferramenta importante para pesquisadores conduzirem novos experimentos computacionais.

Uma das maiores potencialidades deste simulador é a possibilidade de estudar a caminhada com diferentes contornos, definindo apropriadamente algumas ligações interrompidas permanentes. O simulador também possibilita investigar a influência de detectores na caminhada quântica e o comportamento do *mixing time* em diferentes situações. O simulador pode ser útil para simular situações físicas

interessantes, tais como o experimento da dupla fenda dado como exemplo.

Em seguida, passamos à investigação dos efeitos de operadores quânticos imperfeitos nos algoritmos quânticos de busca baseados em caminhadas quânticas. Consideramos erros de fase sistemáticos e aleatórios no operador de moeda. Os efeitos de ligações interrompidas aleatórias afetando o operador de deslocamento também foram considerados. Esse tipo de erro afeta diretamente a propagação espacial do caminhante. Consideramos a busca por um vértice marcado em hipercubos (algoritmo SKW) e em malhas bidimensionais (algoritmo AKR).

Para o algoritmo SKW, encontramos que o efeito geral do ruído no operador de moeda para o vértice marcado é similar àquele do ruído no operador de moeda para os vértices não-marcados, considerado por Li et al. (2006). Também foram encontradas muitas semelhanças com o algoritmo AKR. O efeito qualitativo geral do ruído parece semelhante em todos os algoritmos de busca considerados. Por outro lado, obtivemos resultados quantitativos para a tolerância dos algoritmos a erros.

No contexto do algoritmo de busca de Grover, foi mostrado analiticamente que erros de fase crescendo como $\frac{1}{N^\delta}$, para $\delta \leq \frac{1}{4}$, modificam a complexidade do algoritmo para $O(N^{1-2\delta})$ (Shenvi et al., 2003a). Para $\delta \geq \frac{1}{4}$, a complexidade do algoritmo de Grover com erros é igual à complexidade do caso sem ruído, $O(\sqrt{N})$. Se $\delta < \frac{1}{4}$, a vantagem sobre uma busca clássica, $O(N)$, é reduzida gradativamente. Se $\delta = 0$, ou seja, se o erro é constante, então a complexidade do algoritmo de Grover é igual à complexidade da busca clássica. Nossas simulações numéricas mostram que esse ponto de transição é próximo de $\delta = \frac{1}{2}$ para o algoritmo AKR e próximo de $\delta = 1$ para o algoritmo SKW. Para δ abaixo desses limiares, o algoritmo perde a eficiência gradativamente até se tornar pior que o algoritmo clássico, quando o parâmetro é próximo de $\delta = 0$.

Nossos resultados numéricos também mostram que é possível melhorar a eficiência do algoritmo em todos os casos (com ou sem ruído) se pararmos a execução do mesmo antes do número de passos previsto teoricamente. Neste caso, rodadas

adicionais de execução do algoritmo são necessárias para a amplificação da probabilidade de sucesso, porém mantendo o custo total inferior ao que seria observado usando-se o ponto de parada teórico.

Também apresentamos resultados numéricos preliminares que sugerem a possibilidade de acelerar o algoritmo abstrato de busca por meio de uma redução do número de aplicações do oráculo.

Capítulo 6

Conclusões

Por muito tempo, todos os algoritmos quânticos que eram desenvolvidos podiam ser vistos como variantes do algoritmo de Shor ou do algoritmo de Grover. Era necessário desenvolver uma nova técnica a fim de estimular o surgimento de algoritmos quânticos diferentes. De fato, a computação quântica ganhou novo impulso quando começaram a surgir algoritmos quânticos baseados em caminhadas quânticas. Em seguida, surgiu o conceito de algoritmo abstrato de busca como ferramenta bastante promissora para o desenvolvimento de novos algoritmos quânticos. Esse modelo nos fornece todos os elementos necessários para desenvolver um algoritmo quântico de busca em um grafo regular arbitrário e ainda permite sua análise de complexidade com base em um problema de autovalores. As caminhadas quânticas também possuem propriedades interessantes do ponto de vista da física, que já justificariam seu estudo independentemente das aplicações algorítmicas.

Este trabalho contribuiu para uma melhor compreensão da caminhada quântica no hipercubo. Calculamos analiticamente, pela primeira vez, a distribuição estacionaria do caminhante de Grover nesse grafo, para a condição inicial simétrica. Nossa equação mostrou que essa distribuição não é uniforme. Além disso, observamos que na distribuição estacionária todos os pesos de Hamming são aproximadamente equiprováveis.

De posse da equação exata da distribuição limite, pudemos estudar melhor o *mixing time* dessa caminhada. Mostramos que o *mixing time* M_ϵ da caminhada

no hipercubo de dimensão n cresce com $O(n/\epsilon)$, fato consistente com um resultado mais geral de Aharonov et al. (2001) que particularizado para o hipercubo nos fornece uma cota superior $O(\frac{n^{3/2}}{\epsilon})$. Também realizamos simulações numéricas para estudar o comportamento do mixing time da caminhada com ruído. Nossas simulações também mostraram um mínimo local para a distância entre a distribuição instantânea e a uniforme, em $t = \frac{\pi}{4}n$, como reportado por Moore e Russell (2002), porém o mesmo não foi observado em relação à distribuição estacionária. Encontramos que o *mixing time* instantâneo, quando existe, possui dependência não-linear com a dimensão do hipercubo. A distribuição média da caminhada quântica no hipercubo de dimensão n, seja ela discreta ou contínua no tempo, não converge para a distribuição uniforme, porém ambas são uniformes ou próximas da distribuição uniforme por uma distância ϵ em certos instantes ($t = \frac{\pi}{4}n$).

Na análise da caminhada com descoerência no hipercubo, encontramos resultados similares aos reportados por Kendon e Tregenna (2003) para a caminhada no ciclo. Observamos que a descoerência, mesmo a taxas pequenas, faz com que a distribuição limite seja uniforme. A distância entre a distribuição média e a uniforme cai após um tempo característico $1/p$ e segue uma lei de potências inversa, independente de p. Para o caso de ruído unitário de ligações interrompidas, sem envolver medições, foi encontrada uma taxa de descoerência ótima para a qual o *mixing time* é mínimo. O efeito observado para o ciclo por Kendon e Tregenna (2003) foi semelhante, apesar de ter sido usado outro modelo de descoerência, baseado em medições.

Também estudamos a caminhada quântica na malha bidimensional, tendo em vista a aplicação ao algoritmo de busca AKR. Uma contribuição relevante deste trabalho foi o cálculo da distribuição estacionária para o caso ímpar, analiticamente. Também analisamos numericamente o *mixing time* da caminhada em relação a essa distribuição, tendo encontrado que este cresce como $O\left(\frac{\sqrt{N \log N}}{\epsilon}\right)$. Em seguida, consideramos a caminhada com operador de moeda modificado segundo a prescrição do algoritmo AKR e fizemos simulações numéricas a fim de

relacionar o mixing time dessa caminhada à complexidade do algoritmo de busca. Nossas simulações sugerem que o *mixing time* da caminhada modificada cresce também como $O\left(\frac{\sqrt{N}\log N}{\epsilon}\right)$

Outra contribuição do trabalho foi uma melhor compreensão do impacto da descoerência nos algoritmos de busca baseados em caminhadas quânticas. Também fizemos algumas propostas de pequenas otimizações nos algoritmos de busca, parando a execução dos mesmos antes do ponto previsto teoricamente e aplicando o oráculo um número menor de vezes. Essas propostas podem ajudar também a desenvolver algoritmos mais resistentes a erros, se levarmos em conta a redução no número de portas lógicas necessárias na implementação.

Desenvolvemos uma ferramenta computacional livre, eficiente e multiplataforma, para simulações numéricas de caminhadas quânticas em malhas uni- e bidimensionais com diversos tipos de contorno. Por meio de exemplos de estudos recentes da literatura demonstramos como ela pode ser útil para a comunidade científica, permitindo a obtenção de resultados relevantes através de comandos simples.

Referências Bibliográficas

S. Aaronson e A. Ambainis. Quantum search of spatial regions. In: **Proceedings of 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)**, páginas 200–209, 2003.

G. Abal, R. Donangelo, F. L. Marquezino, A.C. Oliveira, e R. Portugal. Decoherence in search algorithms. In: **Anais do XXIX Congresso da Sociedade Brasileira de Computação, SEMISH**, páginas 293–306, Bento Gonçalves, 2009.

G. Abal, R. Donangelo, F. Severo, e R. Siri. Decoherent quantum walks driven by a generic coin operation. **Physica A**, 387:335–345, 2007.

D. Aharonov, A. Ambainis, J. Kempe, e U. Vazirani. Quantum walks on graphs. In: **Proceedings of 33th ACM Symposium on Theory of Computation (STOC)**, páginas 50–59, New York, NY, July 2001. ACM.

Y. Aharonov, L. Davidovich, e N. Zagury. Quantum random walks. **Physical Review A**, 48(2):1687–1690, 1993.

G. Alagic e A. Russell. Decoherence in quantum walks on the hypercube. **Physical Review A**, 72:062304, 2005.

A. Ambainis. Quantum walk algorithm for element distinctness. In: **Proceedings 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)**, 2004.

A. Ambainis, E. Bach, A. Nayak, A. Vishwanath, e J. Watrous. One-dimensional

quantum walks. In: **Proceedings of the 33rd Symposium on Theory of Computing (STOC)**, páginas 60–69, New York, 2001.

A. Ambainis, J. Kempe, e A. Rivosh. Coins make quantum walks faster. In: **Proceedings of the 16th annual ACM-SIAM Symposium on Discrete Algorithms**, páginas 1099–1108, 2005.

Eric Bach, Susan Coppersmith, Marcel Paz Goldschen, Robert Joynt, e John Watrous. One-dimensional quantum walks with absorbing boundaries. **Journal of Computer and System Sciences**, 69(4):562–592, 2004.

A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, e H. Weinfurter. Elementary gates for quantum computation. **Physical Review A**, 52:3457, 1995.

P. Benioff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. **Journal of Statistical Physics**, 22:563–591, 1980.

P. Benioff. Space searches with a quantum robot. **Contemporary Mathematics**, 305:1–12, 2002.

C. Bennett. Logical reversibility of computation. **IBM Journal of Research and Development**, 17:5225, 1973.

E. Bernstein e U. Vazirani. Quantum complexity theory. **SIAM Journal on Computing**, 26:1411–1478, 1997.

T.A. Brun, H.A. Carteret, e A. Ambainis. Quantum random walks with decoherent coins. **Physical Review A**, 67:032304, 2003.

P. Brémaud. **Markov chains: Gibbs fields, Monte Carlo simulation, and queues**. Springer-Verlag, New York, 1999.

K.K.H. Cheung e M. Mosca. Decomposing finite abelian groups. **Quantum Information and Computation**, 1(3):26–32, 2001.

A. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, e D. Spielman. Exponential algorithmic speedup by a quantum walk. In: **Proceedings of the 35th Annual ACM Symposium on Theory of Computing**, páginas 59–68, 2003.

A.M. Childs e J.M. Eisenberg. Quantum algorithms for subset finding. **Quantum Information and Computation**, 5(7):593–604, 2005. arXiv:quant-ph/0311038.

A.M. Childs, E. Farhi, e S. Gutmann. An example of the difference between quantum and classical random walks. **Quantum Information Processing**, 1 (1-2):35–43, 2002.

A.M. Childs e J. Goldstone. Spatial search by quantum walk. **Physical Review A**, 70:022314, 2004.

A. Church. An unsolvable problem of elementary number theory. **Annals of Mathematics, second series**, 33:346–366, 1936.

R. Cleve. A note on computing Fourier transforms by quantum programs. URL `http://pages.cpsc.ucalgary.ca/~cleve`. 1994.

C. Cohen-Tannoudji, B. Diu, e F. Laloë. **Quantum Mechanics**. Wiley, New York, 1977.

D. Coppersmith. An approximate Fourier transform useful in quantum factoring. Relatório técnico, IBM, New York, July 1994. IBM Research Report 19642.

A.S. Davidov. **Quantum Mechanics**. Pergamon, Oxford, 2 edição, 1976.

D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. **Proceedings of the Royal Society of London A**, 400:97–117, 1985.

D. Deutsch. Quantum computational networks. **Proceedings of the Royal Society of London A**, 425:73, 1989.

T. G. Draper, S. A. Kutin, E. M. Rains, e K. M. Svore. A logarithmic-depth quantum carry-lookahead adder. arXiv:quant-ph/0406142, June 2004.

Thomas G. Draper. Addition on a quantum computer. arXiv:quant-ph/0008033, August 2000.

E. Farhi, J. Goldstone, e S. Gutmann. A quantum algorithm for the hamiltonian nand tree. **Theory of Computing**, 4:169–190, 2008. MIT-CTP-3813, arXiv:quant-ph/0702144.

E. Farhi e S. Gutmann. Quantum computation and decision trees. **Physical Review A**, 58:915–928, 1998.

R. P. Feynman. Simulating physics with computers. **International Journal of Theoretical Physics**, 21:467, 1982.

L. Grover. A fast quantum mechanical algorithm for database search. In: **Proceedings of the 28th Annual ACM Symposium on the Theory of Computation**, páginas 212–219, New York, NY, 1996. ACM Press, New York.

S. Hallgren, A. Russell, e A. Ta-Shma. Normal subgroup reconstruction and quantum computation using group representations. In: **Proceedings of the 32nd Symposium on Theory of Computing**, páginas 627–635, Portland, Oregon, May 2000.

K. Hoffman e R. Kunze. **Linear Algebra**. Prentice Hall, Upper Saddle River, New Jersey, 1971.

N. Inui, Y. Konishi, e N. Konno. Localization of two-dimensional quantum walks. **Physical Review A**, 69:052323, 2004.

G. Ivanyos, F. Magniez, e M. Santha. Efficient quantum algorithms for some instances of the non-Abelian hidden subgroup problem. **International Journal of Foundations of Computer Science**, 14(5):723–739, 2003.

J. Kempe. Quantum random walks – an introductory overview. **Contemporary Physics**, 44(4):307–327, 2003a.

J. Kempe. Quantum random walks hit exponentially faster. In: **Proceedings of 7th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)**, páginas 354–369, 2003b.

V. Kendon. Decoherence in quantum walks – a review. **Mathematical Structures in Computer Science**, 17(6):1169–1220, 2007.

V. Kendon e B. Tregenna. Decoherence can be useful in quantum walks. **Physical Review A**, 67:042315, 2003.

A. Kitaev. Quantum measurements and the abelian stabilizer problem, 1995.

C. Kittel e P. McEuen. **Introduction to solid state physics**. Wiley New York, 1996.

N. Konno. Quantum random walks in one dimension. **Quantum Information Processing**, 1(5):345–354, 2002. arXiv:quant-ph/0206053.

R. Landauer. Irreversibility and heat generation in the computing process. **IBM Journal of Research and Development**, 5:183–191, 1961.

S.A. Lang. **Álgebra Linear**. Ciência Moderna, 2003.

Yun Li, Lei Ma, e Jie Zhou. Gate imperfection in the quantum random-walk search algorithm. **Journal of Physics A**, 39:9309–9319, 2006.

C. Van Loan. **Computational frameworks for the Fast Fourier Transform**. Frontiers in applied mathematics, 10. Society for Industrial and Applied Mathematics, Philadelphia, 1992.

C. Lomont. The hidden subgroup problem: review and open problems. arXiv:quant-ph/0411037, 2004.

Gui Lu Long, Yan Song Li, Wei Lin Zhang, e Chang Cun Tu. Dominant gate imperfection in grover's quantum search algorithm. **Physical Review A**, 61 (4):042305, Mar 2000.

F. Magniez, M. Santha, e M. Szegedy. Quantum algorithms for the triangle problem. **SIAM Journal on Computing**, 37(2):413–424, 2007.

O. Maloyer e V. Kendon. Decoherence vs entanglement in coined quantum walks. **New Journal of Physics**, 9:87, 2007.

Y. Manin. **Computable and Uncomputable**. Sovetskoye Radio, Moscow, 1980.

F.L. Marquezino e R. Portugal. The QWalk simulator of quantum walks. **Computer Physics Communications**, 179:359–369, 2008.

F.L. Marquezino, R. Portugal, G. Abal, e R. Donangelo. Mixing times in quantum walks on the hypercube. **Physical Review A**, 77(4):042312, 2008.

D. Meyer. From quantum cellular automata to quantum lattice gases. **Journal of Statistical Physics**, 85:551–574, 1996.

C. Moore e A. Russell. Quantum walks on the hypercube. In: J. D. P. Rolim e S. Vadhan (editores), **Proceedings of 6th International Workshop on Randomization and Approximation Techniques (RANDOM), Vol. 2483 of Lecture Notes in Computer Science (LNCS)**, páginas 164–178, Cambridge, MA, 2002. Springer-Verlag, Berlin, 2002.

M. Mosca. **Quantum Computer Algorithms**. Tese de Doutorado, University of Oxford, 1999.

R. Motwani e P. Raghavan. **Randomized algorithms**. Cambridge University Press, UK, 1995.

A. Nayak e A. Vishwanath. Quantum walk on a line, 2000. DIMACS Technical Report 2000-43.

M.A. Nielsen e I.L. Chuang. **Quantum Computation and Quantum Information**. Cambridge University Press, UK, 2000.

A.C. Oliveira. **Simulação de caminhos quânticos em redes bidimensionais**. Tese de Doutorado, Laboratório Nacional de Computação Científica, Petrópolis, RJ, Brasil, Junho 2007.

A.C. Oliveira, R. Portugal, e R. Donangelo. Decoherence in two-dimensional quantum walks. **Physical Review A**, 74:012312, 2006a.

A.C. Oliveira, R. Portugal, e R. Donangelo. Two-dimensional quantum walks with boundaries. In: **Proceedings of 1st WECIQ**, páginas 211–218, Pelotas, Brazil, Outubro 2006b. UCPel.

A.C. Oliveira, R. Portugal, e R. Donangelo. Simulation of the single- and double-slit experiments with quantum walkers. arXiv:0706.3181v1 [quant-ph], 2007.

R. Portugal, C.C. Lavor, L.M. Carvalho, e N. Maculan. **Uma Introdução à Computação Quântica**. SBMAC, São Paulo, 2004.

J. Preskill. **Lecture notes for Physics 229: Quantum information and computation**. California Institute of Technology, California, 1998.

A. Romanelli, R. Siri, G. Abal, A. Auyuanet, e R. Donangelo. Decoherence in the quantum walk on the line. **Physica A**, 347C:137–152, 2005.

D. Shapira, O. Biham, A.J. Bracken, e M. Hackett. One dimensional quantum walk with unitary noise. arXiv:quant-ph/0309063, 2003.

S.C. Shapiro. Church's thesis. In: **Encyclopedia of Artificial Intelligence**, páginas 99–100. John Willey & Sons, New York, 1990.

N. Shenvi, K. R. Brown, e K. B. Whaley. Effects of a random noisy oracle on search algorithm complexity. **Physical Review A**, 68:052313, 2003a.

N. Shenvi, J. Kempe, e K. B. Whaley. A quantum random walk search algorithm. **Physical Review A**, 67:052307, 2003b.

P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In: S. Goldwasser (editor), **Proceedings of the 35th Annual Symposium on the Foundations of Computer Science**, páginas 124–134, Los Alamitos, CA, 1994. IEEE Computer Society.

D. Simon. On the power of quantum computation. **SIAM Journal on Computing**, 26(5):1474–1483, 1997.

R. Solovay e V. Strassen. A fast Monte-Carlo test for primality. **SIAM Journal on Computing**, 6(1):84–85, 1977.

G. Strang. **Linear Algebra and its applications**. Brooks Cole, San Diego, 3rd edição, 1988.

F.W. Strauch. Connecting the discrete and continuous-time quantum walks. **Physical Review A**, 74:030301 (R), 2006.

A.S. Tanenbaum. **Organização Estruturada de Computadores**. LTC, Rio de Janeiro, 4 edição, 2001.

A. Tulsi. Faster quantum walk algorithm for the two dimensional spatial search. **Physical Review A**, 78(1):012310, 2008.

A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. **Proceedings of the London Mathematical Society**, 42:230–265, 1936.

J. Watrous. Quantum simulations of classical random walks and undirected graph connectivity. **Journal of Computer and System Sciences**, 62(2):376–391, 2001.

A. Yao. Quantum circuit complexity. In: **Proceedings of the 34th Annual Symposium on Foundations of Computer Science**, páginas 352–360, Los Alamitos, California, 1993. IEEE Press.

Apêndice A

Mecânica quântica e computação quântica

Alguns conflitos entre os resultados experimentais e as previsões teóricas da mecânica Newtoniana e do eletromagnetismo clássico, quando sistemas físicos em escala atômica ou subatômica eram considerados, tornaram-se evidentes por volta do início do século XX. Nessa época, a mecânica quântica começou a ser desenvolvida, passando a fornecer um arcabouço matemático bastante preciso para descrever muitos fenômenos que não podem ser descritos pela mecânica clássica. A física moderna apóia-se fortemente na mecânica quântica e na teoria da relatividade.

Neste capítulo será apresentada uma revisão da mecânica quântica adequada ao estudo da computação quântica. Uma revisão mais detalhada pode ser encontrada nos livros de Cohen-Tannoudji et al. (1977), Davidov (1976), dentre outros.

A.1 Notação de Dirac e álgebra linear

É conveniente revisar algumas notações utilizadas no estudo da mecânica quântica e da computação quântica (Nielsen e Chuang, 2000; Preskill, 1998).

Um vetor em um espaço de Hilbert costuma ser denotado por $|\psi_i\rangle$, ou simplesmente por $|i\rangle$. Este símbolo chama-se *ket*, e faz parte da notação de Dirac. O vetor dual é denotado por $\langle i|$. O produto interno entre $|\psi_i\rangle$ e $|\psi_j\rangle$ é igual a $\langle\psi_i|\,|\psi_j\rangle$, mas pode ser denotado abreviadamente por $\langle\psi_i|\psi_j\rangle$.

O produto tensorial ou produto de Kronecker Loan (1992); Portugal et al.

(2004), é uma forma de reunir espaços vetoriais para formar espaços maiores. É denotado por $|\psi_i\rangle \otimes |\psi_j\rangle$, ou abreviadamente por $|\psi_i\rangle\,|\psi_j\rangle$, ou ainda $|\psi_i\psi_j\rangle$. O produto tensorial de duas matrizes $A_{m\times n}$ e $B_{p\times q}$,

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \; B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1q} \\ b_{21} & b_{22} & \cdots & b_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pq} \end{pmatrix}, \tag{A.1}$$

é a matriz $(A \otimes B)_{mp\times nq}$ definida por

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

Há definições mais gerais para o produto tensorial, porém o tratamento matricial apresentado aqui é suficiente para os propósitos desta dissertação.

O produto de Kronecker satisfaz as seguintes propriedades:

(1) Seja M um espaço vetorial de dimensão m, e seja N um espaço vetorial de dimensão n. Se $|\mu\rangle \in M$ e $|\nu\rangle \in N$, então $|\mu\rangle \otimes |\nu\rangle \in M \otimes N$, e $M \otimes N$ é um espaço vetorial de dimensão mn.

(2) Para qualquer $|\mu\rangle \in M$ e $|\nu\rangle \in N$, e para qualquer escalar z, tem-se

$$z(|\mu\rangle \otimes |\nu\rangle) = (z\,|\mu\rangle) \otimes |\nu\rangle = |\mu\rangle \otimes (z\,|\nu\rangle),$$

(3) Para quaisquer $|\mu_1\rangle, |\mu_2\rangle \in M$ e para qualquer $|\nu\rangle \in N$, tem-se

$$(|\mu_1\rangle + |\mu_2\rangle) \otimes |\nu\rangle = |\mu_1\rangle \otimes |\nu\rangle + |\mu_2\rangle \otimes |\nu\rangle,$$

(4) Para qualquer $|\mu\rangle \in M$ e para quaisquer $|\nu_1\rangle, |\nu_2\rangle \in N$, tem-se

$$|\mu\rangle \otimes (|\nu_1\rangle + |\nu_2\rangle) = |\mu\rangle \otimes |\nu_1\rangle + |\mu\rangle \otimes |\nu_2\rangle.$$

(5) Se $|\psi_A\rangle \in M$ e $|\psi_B\rangle \in N$, e se A e B são operadores lineares definidos nos espaços vetoriais M e N, respectivamente, então $(A \otimes B)(|\psi_A\rangle \otimes |\psi_B\rangle) = A\,|\psi_A\rangle \otimes B\,|\psi_B\rangle$.

Convém lembrar que um operador N é normal se $NN^\dagger = N^\dagger N$; um operador U é unitário se $U^\dagger U = I$; e um operador H é Hermitiano se $H = H^\dagger$. Consequentemente, todo operador Hermitiano é normal. Um resultado muito importante associado aos operadores normais é o teorema espectral,

Teorema A.1 (Teorema espectral)**.** *Para todo operador normal N atuando em um espaço de Hilbert de dimensão finita $\mathcal{H}$, existe uma base ortonormal de $\mathcal{H}$ consistindo de autovetores $|N_i\rangle$ de N.*

Note que o operador N é diagonal em sua própria base de autovetores, ou seja, $N = \sum_i N_i\,|N_i\rangle\,\langle N_i|$. Chamamos de decomposição espectral de N a representação deste operador em sua própria base de autovetores.

Como boas referências em álgebra linear podemos citar, por exemplo, os livros de Hoffman e Kunze (1971), Lang (2003) e de Strang (1988).

A.2 Postulados

Na computação clássica a unidade de informação, chamada de bit, é codificada em um sistema físico com dois estados bem diferenciados. Este sistema pode ser, por exemplo, a voltagem de um sinal elétrico. Na computação quântica a unidade de informação, chamada de q-bit, é codificada em um sistema físico quântico de dois estados ortogonais. Este sistema pode ser, por exemplo, a direção de polarização de um fóton ou a orientação do spin de um elétron. Usualmente denotamos estes estados quânticos ortogonais por meio da notação de Dirac, como $|0\rangle$ e $|1\rangle$.

Podemos definir um q-bit simplesmente como um vetor normalizado no espaço de Hilbert $\mathcal{H}$ de dimensão 2, tomando $\{|0\rangle, |1\rangle\}$ como base ortonormal. Em computação quântica normalmente não precisamos da definição mais geral de espaço de Hilbert, sendo suficiente considerarmos simplesmente o espaço $\mathbb{C}^2$. Assim, um q-bit arbitrário pode ser escrito como

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{A.3}$$

onde $\alpha, \beta \in \mathbb{C}$ são chamadas amplitudes e satisfazem $|\alpha|^2 + |\beta|^2 = 1$. Ao vetor $|\psi\rangle$ dá-se o nome de superposição de estados $|0\rangle$ e $|1\rangle$, e sua interpretação física é a coexistência do q-bit nestes dois estados. Entretanto, ao medir o q-bit na base computacional obtemos o bit clássico $|0\rangle$ com probabilidade $|\alpha|^2$ ou o bit clássico $|1\rangle$ com probabilidade $|\beta|^2$. O processo de medição ficará claro mais adiante, porém deve-se ressaltar que ele é sempre irreversível e destrói a superposição. Além das medições, a descoerência também destrói as superposições quânticas, sendo portanto um dos maiores obstáculos à criação de computadores quânticos.

Apesar de um q-bit aparentemente requerer quatro números reais para ser completamente descrito, pode-se aproveitar a restrição dos valores de suas amplitudes e reescrever a Equação (A.3) como

$$|\psi\rangle = e^{i\gamma}\left(\cos\frac{\theta}{2}\,|0\rangle + e^{i\phi}\sin\frac{\theta}{2}\,|1\rangle\right), \tag{A.4}$$

onde $\gamma, \phi, \theta \in \mathbb{R}$. A fase global $e^{i\gamma}$ pode ser ignorada, por não possuir efeito observável (Nielsen e Chuang, 2000). Assim, pode-se reescrever a equação como

$$|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + e^{i\phi}\sin\frac{\theta}{2}\,|1\rangle\,. \tag{A.5}$$

Desta forma, um q-bit pode ser representado por um ponto em uma esfera no $\mathbb{R}^3$, chamada esfera de Bloch (Figura A.1). No entanto, a mesma representação gráfica não pode ser utilizada para sistemas de múltiplos q-bits.

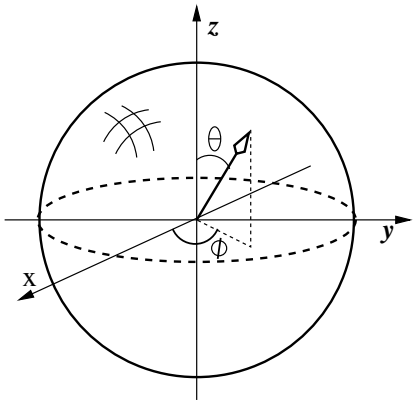

Figura A.1: Esfera de Bloch

Podemos agora generalizar essa discussão e formalizar o primeiro postulado da mecânica quântica, que trata da representação matemática de sistemas quânticos arbitrários.

Postulado 1 (Espaço de estados)**.** *O estado de um sistema físico quântico é descrito por um vetor unitário em um espaço de Hilbert.*

Todas as portas lógicas na computação quântica devem ser representadas por operações unitárias. A porta quântica de Hadamard (H), por exemplo, é uma operação unitária que atua nos estados da base da seguinte forma:

$$H\left|0\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle + \left|1\right\rangle), \qquad H\left|1\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle). \tag{A.6}$$

Note que esta porta produz superposições a partir dos estados de entrada. Representando os vetores da base computacional por

$$\left|0\right\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad \left|1\right\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{A.7}$$

temos que a representação matricial do operador H é dada por

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{A.8}$$

Podemos tratar as operações quânticas de modo mais geral, por meio do segundo postulado da mecânica quântica.

Postulado 2 (Evolução)**.** *A evolução temporal do estado de um sistema físico quântico fechado é descrita por um operador unitário. Isto é, para qualquer evolução do sistema fechado existe um operador unitário U tal que, se o estado inicial do sistema é $|\Psi_0\rangle$, então após a evolução o estado do sistema será $|\Psi_f\rangle = U|\Psi_0\rangle$.*

Neste trabalho também consideraremos espaços de Hilbert de dimensões maiores. Com frequência construiremos espaços de estados maiores a partir da concatenação de uma sequência de estados menores. Dizemos que um estado quântico de n q-bits, por exemplo, é um vetor unitário $|\psi\rangle$ no espaço de Hilbert $\mathcal{H}_n$, ou simplesmente no espaço $\mathbb{C}^{2^n}$. Estes espaços de Hilbert são obtidos a partir do produto tensorial dos espaços de um q-bit,

$$\mathcal{H}_n = \underbrace{\mathcal{H} \otimes \cdots \otimes \mathcal{H}}_{n \text{ vezes}}.$$

Estados quânticos com $n > 1$ q-bits são frequentemente chamados de registradores quânticos.

Quando consideramos estados de mais de um q-bit obtemos propriedades interessantes, por vezes sem nenhum análogo na Computação Clássica. Vamos ilustrar algumas dessas propriedades por meio de um exemplo simples. Consideremos um estado de dois q-bits, inicialmente no estado $|00\rangle$. Esta notação significa que temos um produto tensorial de dois q-bits no estado $|0\rangle$, ou seja, $|00\rangle = |0\rangle \otimes |0\rangle$. Aplicamos, então, a operação unitária $H \otimes H$ sobre o estado de dois q-bits $|00\rangle$, obtendo o estado

$$\begin{aligned}
(H \otimes H)|00\rangle &= H|0\rangle \otimes H|0\rangle \\
&= \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \\
&= \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right) \qquad \text{(A.9)}
\end{aligned}$$

Portanto, como resultado da aplicação de $H \otimes H$ ao estado $|00\rangle$ obtivemos uma superposição de todos os possíveis estados com dois q-bits. Este tipo de superposição é utilizado com frequência em algoritmos quânticos, a fim de explorar o chamado paralelismo quântico. Se temos um operador U_f que, de alguma forma, calcula uma função $f(x)$, então a aplicação de U_f a um estado como o da Equação (A.9) faz com que f seja calculada simultaneamente para muitos valores de x. No entanto, cabe ressaltar que apesar do resultado do cálculo ser obtido paralelamente, o mesmo não se torna totalmente disponível, já que o processo de medição provoca um colapso irreversível das superposições. É necessário processar de modo adequado a informação antes de medir o sistema, a fim de aproveitar eficientemente o paralelismo quântico.

Também podemos — e frequentemente iremos — representar o estado (A.9) usando uma notação mais compacta, como $\frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle)$. Caso o contexto não deixe claro se o conteúdo do ket está escrito em binário ou decimal, será feito algum comentário a fim de evitar ambiguidades.

Podemos agora generalizar essa discussão e formalizar o terceiro postulado, que trata da composição de sistemas físicos quânticos arbitrários.

Postulado 3 (Composição de sistemas)**.** *Quando dois sistemas físicos são tratados como um sistema combinado, o espaço de estados do sistema físico combinado é o espaço produto de Kronecker $\mathcal{H}_1 \otimes \mathcal{H}_2$ dos espaços de estados $\mathcal{H}_1, \mathcal{H}_2$ dos subsistemas componentes. Se o primeiro sistema está no estado $|\Psi_1\rangle$ e o segundo sistema está no estado $|\Psi_2\rangle$, então o estado do sistema combinado é $|\Psi_1\rangle \otimes |\Psi_2\rangle$.*

Aproveitando ainda o estado (A.9), vamos dar prosseguimento ao nosso exemplo, aplicando uma porta de Hadamard somente ao segundo q-bit. Ao primeiro q-bit será aplicada a identidade, ou seja, ele não será modificado. Desse modo, temos,

$$\begin{aligned}(I \otimes H)\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) &= \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes |0\rangle \\ &= \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \quad \text{(A.10)}\end{aligned}$$

Em seguida, vamos aplicar uma porta CNOT[1] ao estado resultante. Esta porta inverte o q-bit alvo se o q-bit de controle é igual a $|1\rangle$, e não faz nada caso contrário. Podemos descrever a porta CNOT de forma compacta, utilizando a operação de soma modulo 2, denotada por $\oplus$:

$$C\,|a,b\rangle = |a, a \oplus b\rangle\,, \tag{A.11}$$

onde $a, b \in \{0,1\}$, a é o q-bit de controle e b é o q-bit alvo. Outras portas quânticas também podem atuar de modo controlado, seguindo a mesma estrutura da porta CNOT. Além disso, é possível definir portas com múltiplos q-bits de controle. Por exemplo, podemos ter um Hadamard controlado, que atua sobre um q-bit alvo se e somente se dois q-bits de controle forem iguais a $|1\rangle$. Ao contrário das operações $H \otimes H$ e $I \otimes H$, apresentadas anteriormente, a porta CNOT não pode ser fatorada em produto tensorial de operadores atuando sobre um q-bit cada. Existe um resultado interessante na computação quântica, segundo o qual todas as portas quânticas podem ser fatoradas em CNOTs e portas sobre um q-bit (Barenco et al., 1995).

Na base computacional, a porta CNOT com alvo no segundo q-bit e controle no primeiro q-bit pode ser representada matricialmente como

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{A.12}$$

Aplicando a porta CNOT descrita acima ao estado (A.10), obtemos

$$C\left(\frac{1}{\sqrt{2}}\,|00\rangle + \frac{1}{\sqrt{2}}\,|10\rangle\right) = \frac{1}{\sqrt{2}}\,|00\rangle + \frac{1}{\sqrt{2}}\,|11\rangle\,. \tag{A.13}$$

O interessante nesse estado é o fato dele não poder ser decomposto. De fato, é

[1] *Controlled NOT*, ou NOT controlado.

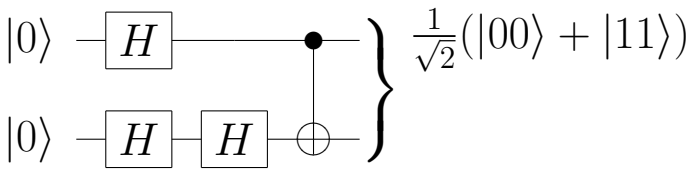

Figura A.2: Circuito representando a sequência de operações realizadas no exemplo.

bastante simples verificar que não existem $\alpha, \beta, \gamma, \delta \in \mathbb{C}$ tais que

$$(\alpha\,|0\rangle + \beta\,|1\rangle) \otimes (\gamma\,|0\rangle + \delta\,|1\rangle) = \frac{1}{\sqrt{2}}\,|00\rangle + \frac{1}{\sqrt{2}}\,|11\rangle\,.$$

Estados que não podem ser decompostos são chamados de estados emaranhados, e desempenham papel fundamental na computação quântica.

As operações quânticas também podem ser representadas por circuitos, análogos aos circuitos utilizados na computação clássica. Nesse modelo, as portas lógicas são representadas por caixas e os fios representam o fluxo dos dados de uma porta até a outra. O circuito é sempre lido da esquerda para a direita. O símbolo $\bullet$ é colocado sobre os fios correspondentes aos q-bits de controle, e uma caixa com a identificação da porta lógica é colocada sobre o fio correspondente aos q-bits alvo. A porta lógica quântica NOT pode ainda ser representada pelo símbolo $\oplus$ no modelo de circuitos.

Na Figura A.2 temos o circuito referente ao exemplo desta seção. Iniciamos com o estado $|00\rangle$. Em seguida aplicamos o operador H a cada um dos q-bits de entrada. Depois, aplicamos o operador H apenas ao segundo q-bit. Por fim, aplicamos uma porta CNOT com controle no primeiro q-bit e alvo no segundo q-bit. A saída é um estado emaranhado.

Vimos como a informação quântica é representada matematicamente e como os estados quânticos evoluem no tempo. Convém ressaltar que o postulado da evolução pressupõe que o sistema físico é fechado e, portanto, não interage com o meio. Em algum momento estaremos interessados em medir alguma propriedade do sistema a fim de obter informação para a computação realizada. Nesse momento

será necessário que o sistema interaja com o equipamento de medida, de modo que o postulado da evolução não pode ser usado para descrever o processo. A evolução do estado de um sistema físico quântico durante o processo de medição não é unitária. Portanto, é necessário introduzir mais um postulado.

Postulado 4 (Medição)**.** *As medições quânticas são descritas por operadores* $\{M_m\}$*, que atuam sobre o espaço de estados do sistema e satisfazem a relação de completitude,* $\sum_m M_m^\dagger M_m = I$*. O índice* m *se refere aos possíveis resultados da medida. Se o estado de um sistema quântico imediatamente antes da medida for* $|\Psi\rangle$*, então a probabilidade do resultado* m *ocorrer é*

$$p(m) = \langle \Psi | M_m^\dagger M_m | \Psi \rangle \tag{A.14}$$

e o estado do sistema após a medida será

$$|\Psi'\rangle = \frac{1}{\sqrt{p(m)}} M_m \, |\Psi\rangle \, . \tag{A.15}$$

A.3 Histórico do processamento quântico da informação

Duas importantes teorias desenvolvidas no início do século XX foram marcantes. Uma delas foi a mecânica quântica, um arcabouço matemático que permitiu a elaboração de teorias precisas para descrever sistemas físicos muito pequenos, tais como fótons e elétrons. Outra grande conquista foi o desenvolvimento da Teoria da Computação, com grande colaboração de Turing (1936). Em seu artigo, motivado pelo *Entscheidungsproblem* de Hilbert, ele descreve a noção abstrata de uma máquina capaz de executar algoritmos: a Máquina de Turing. Trata-se de uma máquina teórica composta de: um programa; um controle de estados finitos, consistindo de um conjunto finito de estados internos; uma fita, desempenhando o papel de memória do computador; e uma cabeça de leitura e gravação (Nielsen e Chuang, 2000). Turing criou ainda a noção de computador programável, demonstrando a existência de uma Máquina Universal de Turing, capaz de simular

qualquer outra Máquina de Turing. Outro passo importante para o estabelecimento da Ciência da Computação foi a tese de Church-Turing, — nomeada desta forma por ter sido desenvolvida originalmente por Church (1936) e posteriormente aprofundada pelo já mencionado matemático inglês — segundo a qual todo processo algorítmico que possa ser realizado na natureza pode ser descrito por uma Máquina de Turing (Shapiro, 1990).

Desde a invenção do transistor por Bardeen, Brattain e Shockley em 1948, um grande progresso foi observado no desenvolvimento dos computadores (Tanenbaum, 2001). Estes tornavam-se cada vez menores e mais velozes. Gordon Moore, em 1965, estabeleceu uma lei segundo a qual o número de transistores por unidade de área — e consequentemente, o poder de processamento dos computadores — dobraria aproximadamente a cada dois anos (Moore e Russell, 2002). Esta lei ficou conhecida como lei de Moore, e de fato conseguiu prever razoavelmente a evolução dos computadores até o presente. No entanto, para que a lei de Moore seja válida, faz-se necessária uma constante miniaturização dos componentes dos computadores. Naturalmente, chegará o dia em que os componentes alcançarão o limite de indivisibilidade da matéria, e a lei de Moore deixará de ser válida. Mesmo antes disso, surgirão problemas quando os componentes dos computadores forem se aproximando de dimensões atômicas, devido ao aparecimento de efeitos quânticos. A fim de vencer o obstáculo imposto pela natureza, e dar continuidade ao avanço dos computadores, existem ao menos duas possibilidades. Uma delas envolve o aperfeiçoamento da própria computação clássica, como a introdução de modificações na arquitetura dos computadores ou a utilização de computação paralela, por exemplo. A outra possibilidade consiste na utilização daquilo que, inicialmente, parecia o grande obstáculo: os efeitos quânticos da matéria. As duas alternativas representam grandes desafios da ciência da computação na atualidade, e neste trabalho iremos nos concentrar na segunda.

De fato, a teoria da computação não pode ser totalmente separada da física. Essa afirmação torna-se bastante clara através de Landauer, que em 1961 publi-

cou um importante trabalho onde é feito um estudo da relação entre consumo de energia e computação. Segundo Landauer, a energia dissipada por um computador quando este apaga um único bit de informação é maior ou igual a $k_B T \ln 2$, onde k_B é a constante de Boltzmann e T é a temperatura do ambiente do computador (Landauer, 1961). Outro resultado notável na física da computação foi o artigo de Bennett (1973), onde é demonstrada a possibilidade de realização de operações computacionais reversíveis. Como consequência deste trabalho, tem-se que é possível realizar operações computacionais sem dissipação de energia.

A partir da década de 1980 surgiram alguns trabalhos muito importantes para a computação quântica. Uma contribuição importante foi o conceito de Máquina de Turing quântica, desenvolvido por Benioff (1980), e posteriormente aprofundado por Deutsch (1985), Yao (1993) e Bernstein e Vazirani (1997). Destaca-se também o artigo de Feynman (1982), onde este físico norte-americano argumentava que a simulação de sistemas físicos quânticos por Máquinas de Turing seria um problema de complexidade exponencial, e que para simular eficientemente sistemas quânticos, seria necessário construir um computador baseado nos mesmos princípios da mecânica quântica. O trabalho desenvolvido independentemente por Manin (1980) também trazia conclusões semelhantes. Em 1989 o modelo de circuitos quânticos foi desenvolvido por Deutsch, e em 1993 foi aprofundado por Yao (Deutsch, 1989; Yao, 1993). Posteriormente, Barenco et al. (1995) demonstraram a universalidade das portas CNOT e portas atuando em um q-bit.

Para se melhor compreender as motivações da computação quântica, é interessante fazer uma breve digressão sobre a tese de Church-Turing. Sua versão forte dizia que qualquer processo algorítmico da natureza pode ser simulado **eficientemente** por uma Máquina de Turing. A tese de Church-Turing normalmente é aceita sem muita hesitação, porém a sua versão forte já foi desafiada pela existência de algoritmos probabilísticos que não parecem ter solução eficiente em Máquinas de Turing determinísticas[2] . Uma saída paliativa seria modificar a ver-

[2] Segundo Nielsen e Chuang (2000), o primeiro destes algoritmos foi o teste de primalidade desenvolvido por Solovay e Strassen (1977).

são forte, passando a dizer que qualquer processo algorítmico da natureza pode ser simulado eficientemente por uma Máquina de Turing **probabilística**. No entanto, dentro deste contexto, uma questão levantada pelo físico David Deutsch é se existe um modelo computacional, baseado nas leis da física, com o qual seja possível estabelecer uma versão ainda mais forte da tese de Church-Turing. Este modelo computacional deveria ser capaz de simular **eficientemente um sistema físico arbitrário**. Feynman (1982), em um artigo seminal, já argumentava que a simulação de sistemas quânticos de n partículas por equipamentos clássicos é um problema aparentemente de complexidade exponencial. Em outras palavras, sistemas físicos quânticos parecem não obedecer a versão forte da tese de Church-Turing. Portanto, esse mesmos sistemas quânticos, se utilizados como dispositivos computacionais, talvez possam ser mais eficientes que dispositivos computacionais clássicos.

Já em 1985, Deutsch utilizou a teoria da mecânica quântica e conseguiu desenvolver um algoritmo quântico mais rápido que qualquer algoritmo possível de ser implementado em um computador clássico (Deutsch, 1985). Para uma função $f(x) : \{0,1\} \rightarrow \{0,1\}$ o algoritmo de Deutsch verifica se $f(0) = f(1)$ ou $f(0) \neq f(1)$, avaliando $f(x)$ apenas uma vez. O algoritmo de Deutsch não possui nenhuma aplicação prática. No entanto, vários pesquisadores conseguiram resolver eficientemente, utilizando o formalismo da mecânica quântica, alguns problemas computacionais de grande importância que não possuem solução eficiente conhecida em Máquinas de Turing — mesmo probabilísticas. Shor (1994) desenvolveu algoritmos para fatoração de inteiros grandes e cálculo de logaritmo discreto, utilizando para isso um algoritmo para cálculo de DFT em computadores quânticos, eficiente quando N é menor que $\log N$, onde N é a quantidade de elementos do vetor a ser transformado. No mesmo ano, motivado pelo trabalho de Shor, Coppersmith (1994) desenvolveu uma versão quântica da FFT quando N é potência de dois. Esta importante sub-rotina quântica é a QFT. A mesma sub-rotina foi desenvolvida independentemente por Cleve (1994), através de uma abordagem re-

cursiva.

Kitaev (1995) utiliza a QFT para calcular ordem de elementos de um grupo. Simon (1997) desenvolveu um algoritmo quântico para resolver uma instância do Problema do Subgrupo Escondido (HSP[3]), quando $G = \mathbb{Z}_2^n$. Há ainda resultados mais recentes em algoritmos para Teoria de Grupos como, por exemplo, o trabalho de Hallgren et al. (2000), onde se apresenta uma solução do HSP para um subgrupo normal através da QFT, e os trabalhos de Mosca (1999); Watrous (2001); Cheung e Mosca (2001) e de Ivanyos et al. (2003) onde se discutem problemas como decomposição de grupos Abelianos e cálculo de ordem de grupos solúveis.

Todos os algoritmos mencionados nos parágrafos anteriores, que apresentam ganho exponencial de complexidade em relação aos algoritmos clássicos, utilizam a QFT. A transformada de Fourier quântica ainda foi utilizada no desenvolvimento de um algoritmo para soma, sem no entanto alcançar ganho exponencial de complexidade em relação ao algoritmo clássico (Draper, 2000; Draper et al., 2004). Como já mencionado neste trabalho, Grover (1996) desenvolveu um algoritmo quântico para busca em listas não-ordenadas com ganho quadrático em relação ao melhor algoritmo clássico.

Desta forma, a presença de efeitos quânticos nos computadores pode não ser um problema. Ao contrário, pode ser a oportunidade de desenvolver um modelo computacional muito mais eficiente que os atuais baseados na mecânica Newtoniana.

[3] *Hidden Subgroup Problem.* Dado um grupo G, por meio de um conjunto gerador, o HSP consiste em encontrar os geradores de um subgrupo H. Para isso, utiliza-se uma função, chamada oráculo, que diz se um elemento em G também pertence a H ou a um coset de H (Lomont, 2004).

Apêndice B

Análise do algoritmo abstrato de busca

Neste apêndice, apresentamos uma análise do algoritmo abstrato de busca, tendo como principal referência o artigo de Tulsi (2008). No algoritmo abstrato de busca aplica-se o operador $U_A = U \cdot R_t$ repetidas vezes sobre o estado $|\Psi_0\rangle$, que é o único autovetor de U associado ao autovalor 1. Aqui, U é um operador real e R_t é um operador de reflexão sobre o estado alvo, $|t\rangle$. Podemos encontrar propriedades importantes da caminhada modificada, U_A, por meio da decomposição espectral de U e dos coeficientes da expansão de $|t\rangle$ na base de autovetores de U.

Como U é uma matriz unitária real, seus autovalores diferentes de ± 1 existem em pares de complexos conjugados, $e^{\pm i\theta}$. O autovalor correspondente ao autovalor 1, ou seja, $\theta = 0$, é o estado diagonal, que iremos denotar por $|\Phi_0\rangle$. Os autovetores correspondentes ao autovalor -1, ou seja, $\theta = \pi$, são denotados por $|\Phi_k\rangle$. Os autovetores correspondentes aos autovalores $e^{\pm i\theta_j}$, diferentes de ± 1, são denotados por $\left|\Phi_j^{\pm}\right\rangle$. Portanto, como U é uma matriz unitária real, temos

$$\begin{aligned} U\left|\Phi_j^{-}\right\rangle &= e^{-i\theta_j}\left|\Phi_j^{-}\right\rangle, \\ U\left|\Phi_j^{-}\right\rangle^{*} &= e^{i\theta_j}\left|\Phi_j^{-}\right\rangle^{*}, \end{aligned} \tag{B.1}$$

e portanto, $\left|\Phi_j^{-}\right\rangle^{*} = \left|\Phi_j^{+}\right\rangle$.

Sejam $a_0 = \langle\Phi_0|t\rangle$, $a_j^{\pm} = \langle\Phi_j^{\pm}|t\rangle$ e $a_k = \langle\Phi_k|t\rangle$ os coeficientes do estado procurado $|t\rangle$ na base de autovetores de U. Como $|t\rangle$ é real, temos que $a_j^{+} = (a_j^{-})^{*}$. Além disso, a menos de uma fase global, pode-se escolher $\left|\Phi_j^{\pm}\right\rangle$ de modo que

$a_j^+ = a_j^- = a_j$. Também é possível escolher $|\Psi_0\rangle$ e $|\Psi_k\rangle$ de modo que a_0, a_k sejam reais. Então,

$$|t\rangle = a_0 |\Phi_0\rangle + \sum_j a_j(\left|\Phi_j^+\right\rangle + \left|\Phi_j^-\right\rangle) + \sum_k a_k |\Phi_k\rangle . \tag{B.2}$$

Para um número real λ, definimos o vetor não-unitário $|\omega_\lambda\rangle$, cuja expansão na base de autovetores de U é dada por $\langle\Phi_l|\omega_\lambda\rangle = a_l F_\lambda(\theta_l)$, em que $F_\lambda(\theta_l) = \cot\left(\frac{\lambda-\theta_l}{2}\right)$. Existem algumas propriedades importantes, que serão utilizadas mais adiante:

$$e^{i\theta}(-1 + iF_\lambda(\theta)) = e^{i\lambda}(1 + iF_\lambda(\theta)), \tag{B.3}$$

$$F_\lambda(\theta) + F_\lambda(-\theta) = \frac{2\sin\lambda}{\cos\theta - \cos\lambda}, \tag{B.4}$$

$$F_\lambda(0) = \cot\frac{\lambda}{2}, \tag{B.5}$$

$$F_\lambda(\pi) = -\tan\frac{\lambda}{2}. \tag{B.6}$$

As demonstrações dessas propriedades serão omitidas neste trabalho, porém elas podem ser encontradas no artigo de Ambainis et al. (2005).

Proposição B.1. *O vetor $|\omega_\lambda\rangle$ é ortogonal a $|t\rangle$ se e somente se o vetor (não-normalizado) $|\lambda\rangle = |t\rangle + i|\omega_\lambda\rangle$ é um autovetor do operador $U_A = U \cdot R_t$ com autovalor $e^{i\lambda}$.*

Demonstração. Primeiramente, notamos que

$$\begin{aligned}\langle\Phi_l|\lambda\rangle &= \langle\Phi_l|t\rangle + i\langle\Phi_l|\omega_\lambda\rangle \\ &= a_l + ia_l F_\lambda(\theta_l) \\ &= a_l(1 + iF_\lambda(\theta_l)).\end{aligned} \tag{B.7}$$

Também podemos notar que $R_t|\lambda\rangle = -|t\rangle + i|\omega_l\rangle$, se e somente se $|\omega_\lambda\rangle \perp |t\rangle$, pois

nesse caso R_t não altera $|\omega_\lambda\rangle$. Assim, temos,

$$\begin{aligned}\langle\Phi_l|R_t|\lambda\rangle &= -\langle\Phi_t|t\rangle + i\langle\Phi_t|\omega_l\rangle \\ &= -a_l + ia_l F_\lambda(\theta_l) \\ &= a_l(-1 + iF_\lambda(\theta_l)),\end{aligned} \tag{B.8}$$

se e somente se $|\omega_\lambda\rangle \perp |t\rangle$. Como $|\Phi_l\rangle$ são autovetores de U, temos

$$\begin{aligned}\langle\Phi_l|U \cdot R_t|\lambda\rangle &= e^{i\theta_l}\langle\Phi_l|R_t|\lambda\rangle \\ &= a_l e^{i\theta_l}(-1 + iF_\lambda(\theta_l)).\end{aligned} \tag{B.9}$$

Usando as propriedades mencionadas anteriormente neste apêndice, temos

$$\begin{aligned}\langle\Phi_l|U \cdot R_t|\lambda\rangle &= a_l e^{i\lambda}(1 + iF_\lambda(\theta_l)) \\ &= e^{i\lambda}\langle\Phi_l|\lambda\rangle.\end{aligned} \tag{B.10}$$

Como a equação anterior vale para todo vetor $|\Phi_l\rangle$ da base, segue-se que $|\lambda\rangle$ é autovetor de $U_A = U \cdot R_t$, com autovalor associado $e^{i\lambda}$, se e somente se $|\omega_\lambda\rangle$ é ortogonal a $|t\rangle$. □

Verifica-se facilmente que a condição de $|\omega_\lambda\rangle$ ser ortogonal a $|t\rangle$ é equivalente a $\sum_l a_l^2 F_\lambda(\theta_l) = 0$. Expandindo esse somatório e usando a Equação (B.4), temos

$$\begin{aligned}&a_0^2 F_\lambda(\theta_0) + \sum_j a_j^2 F_\lambda(\theta_j) + a_j^2 F_\lambda(-\theta_j) + \sum_k a_k^2 F_\lambda(\theta_k) = 0 \\ &a_0^2 \cot\frac{\lambda}{2} + \sum_j \frac{2a_j^2 \sin\lambda}{\cos\theta_j - \cos\lambda} + \sum_k a_k^2 \cot\frac{\lambda - \pi}{2} = 0.\end{aligned} \tag{B.11}$$

Portanto, a condição é equivalente a

$$a_0^2 \cot\frac{\lambda}{2} = \sum_j \frac{2a_j^2 \sin\lambda}{\cos\lambda - \cos\theta_j} + \sum_k a_k^2 \tan\frac{\lambda}{2}$$
$$a_0^2 \frac{\cot\frac{\lambda}{2}}{\sin\lambda} = \sum_j \frac{2a_j^2}{\cos\lambda - \cos\theta_j} + A_k^2 \frac{\tan\frac{\lambda}{2}}{\sin\lambda}, \tag{B.12}$$

em que $A_k = \sqrt{\sum_k a_k^2}$. Note que A_k é a projeção do estado $|t\rangle$ sobre o autoespaço de U associado ao autovalor -1. É fácil verificar que a Equação (B.12) também é válida para $-\lambda$.

Seja θ_{min} o menor ângulo dentre os θ_j. Ambainis et al. (2005) demonstraram que a Equação (B.12) tem exatamente duas soluções, $\lambda = \pm\alpha$, tais que $|\alpha| < \frac{\theta_{min}}{2}$. Como veremos mais adiante, os autovetores correspondentes aos autovalores $e^{\pm i\alpha}$ geram de modo aproximado o vetor $|\Phi_0\rangle$ e, portanto, a iteração de U_A sobre $|\Phi_0\rangle$ pode ser analisada considerando-se somente estes autovetores.

Tipicamente, θ_{min} é muito pequeno, de modo que α também é muito pequeno. Avaliando a Equação (B.12) em α obtemos

$$\frac{a_0^2}{\alpha^2} = \sum_j \frac{a_j^2}{\cos\alpha - \cos\theta_j} + \frac{A_k^2}{4}. \tag{B.13}$$

Como mostrado no artigo de Ambainis et al. (2005),

$$\sum_j \frac{a_j^2}{\cos\alpha - \cos\theta_j} = \Theta\left(\sum_j \frac{a_j^2}{1 - \cos\theta_j}\right), \tag{B.14}$$

de modo que

$$\alpha = \Theta\left(\frac{a_0}{\sqrt{\sum_j \frac{a_j^2}{1-\cos\theta_j} - \frac{A_k^2}{4}}}\right). \tag{B.15}$$

Agora, sejam $|\pm\alpha\rangle = |t\rangle + i\,|\omega_{\pm\alpha}\rangle$ autovetores não-normalizados de U_A correspondendo aos autovalores $e^{\pm i\alpha}$. Seja $|\alpha_u^-\rangle = |\alpha\rangle - |-\alpha\rangle = i(|\omega_\alpha\rangle - |\omega_{-\alpha}\rangle)$ um estado não-normalizado e seja $|\alpha^-\rangle = |\alpha_u^-\rangle / \| |\alpha_u^-\rangle \|$ o estado normalizado correspondente.

Proposição B.2. *O estado inicial* $|\Phi_0\rangle$ *é gerado pelos autovetores* $|\pm\alpha\rangle$.

Demonstração. Vamos calcular a sobreposição de $|\Phi_0\rangle$ com o vetor $|\alpha^-\rangle$. Os coeficientes da expansão de α_u^- na base de autovetores de U são dados por

$$\begin{aligned} |\langle\Phi_l|\alpha_u^-\rangle| &= \langle\Phi_l|\omega_\alpha\rangle - \langle\Phi_l|\omega_{-\alpha}\rangle \\ &= a_l(F_\alpha(\theta_l) - F_{-\alpha}(\theta_l)). \end{aligned} \tag{B.16}$$

Fazendo $l = 0$ na equação acima, obtemos

$$\begin{aligned} |\langle\Phi_0|\alpha_u^-\rangle| &= a_0(F_\alpha(0) - F_{-\alpha}(0)) \\ &= a_0(\cot\frac{\alpha}{2} - \cot\frac{-\alpha}{2}) \\ &= 2a_0\cot\frac{\alpha}{2}. \end{aligned} \tag{B.17}$$

Um vez que $|\langle\Phi_0|\alpha^-\rangle| = \dfrac{|\langle\Phi_0|\alpha_u^-\rangle|}{\| \, |\alpha_u^-\rangle \, \|}$, resta-nos somente limitar $\| \, |\alpha_u^-\rangle \, \|$ para que possamos limitar $|\langle\Phi_0|\alpha^-\rangle|$. Temos que

$$\begin{aligned} \| \, |\alpha_u^-\rangle \, \| &= \sqrt{\sum_l |\langle\Phi_l|\alpha_u^-\rangle|^2} \\ &= \sqrt{\sum_l |a_l(F_\alpha(\theta_l) - F_{-\alpha}(\theta_l))|^2}. \end{aligned} \tag{B.18}$$

No somatório em l, o termo T_0 correspondente a $l = 0$ é

$$\begin{aligned} T_0 &= |\langle\Phi_0|\alpha_u^-\rangle| \\ &= 4a_0^2\cot^2\frac{\alpha}{2} \\ &= \Theta\left(\frac{a_0^2}{\alpha^2}\right). \end{aligned} \tag{B.19}$$

Calculando os termos do somatório correspondentes a $l \in k$, temos

$$\begin{aligned} |\langle \Phi_k | \alpha_u^- \rangle| &= \langle \Phi_k | \omega_\alpha \rangle - \langle \Phi_k | \omega_{-\alpha} \rangle \\ &= a_k (F_\alpha(\pi) - F_{-\alpha}(\pi)) \\ &= -2a_k \tan \frac{\alpha}{2}. \end{aligned} \tag{B.20}$$

Portanto, o termo T_k do somatório, correspondente a $l \in k$, é

$$\begin{aligned} T_k &= \sum_k |\langle \Phi_k | \alpha_u^- \rangle|^2 \\ &= 4 \tan^2 \frac{\alpha}{2} \left(\sum_k a_k \right)^2 \\ &= \Theta \left(A_k^2 \alpha^2 \right). \end{aligned} \tag{B.21}$$

Finalmente, o termo T_j do somatório, correspondente a $l \in j$, foi calculado por Ambainis et al. (2005), que encontraram

$$T_j = \Theta \left(\alpha^2 \sum_j \frac{a_j^2}{(1 - \cos \theta_j)^2} \right). \tag{B.22}$$

Além disso, no caso da busca espacial, eles mostraram que T_0 é muito maior que T_j e T_k quando N é grande. Portanto, como $\| \, |\alpha_u^-\rangle \, \| = \sqrt{T_0 + T_j + T_k}$, temos que

$$\begin{aligned} |\langle \Phi_0 | \alpha^- \rangle| &= \frac{\sqrt{T_0}}{\| \, |\alpha_u^-\rangle \, \|} \\ &= 1 - \frac{T_j + T_k}{2T_0}. \end{aligned} \tag{B.23}$$

Mais explicitamente,

$$|\langle \Phi_0 | \alpha^- \rangle| \geq 1 - \Theta \left(\alpha^4 \sum_j \frac{a_j^2}{a_0^2} \frac{1}{(1 - \cos \theta_j)^2} \right) - \Theta \left(\frac{A_k^2 \alpha^4}{a_0^2} \right). \tag{B.24}$$

Logo, o estado inicial do algoritmo, $|\Phi_0\rangle$, é muito próximo de $|\alpha^-\rangle = c(|\alpha\rangle - |-\alpha\rangle)$, em que c é o fator de normalização. □

Como $|\pm\alpha\rangle$ são autovetores de U_A associados a autovalores $e^{\pm i\alpha}$, temos que $U_A^q |\alpha^-\rangle = c(e^{iq\alpha}|\alpha\rangle - e^{-iq\alpha}|-\alpha\rangle)$. Após $T = \left\lceil \frac{\pi}{2\alpha} \right\rceil$ iterações de U_A, chega-se perto do estado $|\alpha^+\rangle = c(|\alpha\rangle + |-\alpha\rangle)$. Para finalizar a análise, mostraremos que o estado $|\alpha^+\rangle$ é uma boa aproximação para o elemento procurado.

Seja $|\alpha_u^+\rangle$ um estado não-normalizado. Podemos normalizá-lo fazendo $|\alpha_u^+\rangle = \frac{|\alpha_u^+\rangle}{\| |\alpha_u^+\rangle \|}$, e portanto $|\langle t|\alpha^+\rangle| = \frac{|\langle t|\alpha_u^+\rangle|}{\| |\alpha_u^+\rangle \|}$. Como $|\alpha_u^+\rangle = 2|t\rangle + i(|\omega_\alpha\rangle + |\omega_{-\alpha}\rangle)$ e $|\omega_{\pm\alpha}\rangle$ são ortogonais a $|t\rangle$, temos $|\langle t|\alpha_u^+\rangle| = 2$. Também temos que

$$\begin{aligned} \| \left|\alpha_u^+\right\rangle \|^2 &= \|2|t\rangle + i(|\omega_\alpha\rangle + |\omega_{-\alpha}\rangle)\|^2 \\ &= 4 + \| |\omega_\alpha\rangle + |\omega_{-\alpha}\rangle \|^2. \end{aligned} \tag{B.25}$$

A expansão do vetor $|\omega_\alpha\rangle + |\omega_{-\alpha}\rangle$ na base de autovetores de U é dada por $\langle\Phi_l| (|\omega_\alpha\rangle + |\omega_\alpha\rangle) = a_l(F_\alpha(\theta_l) + F_{-\alpha}(\theta_l))$, e portanto,

$$\| |\omega_\alpha\rangle + |\omega_{-\alpha}\rangle \|^2 = \sum_l |a_l(F_\alpha(\theta_l) + F_{-\alpha}(\theta_l))|^2. \tag{B.26}$$

Os temos correspondentes a $l \in \{0, k\}$ desaparecem. Já os termos correspondentes a $l \in j$ somam

$$\| |\omega_\alpha\rangle + |\omega_{-\alpha}\rangle \|^2 = \Theta\left(\sum_j a_j^2 \cot^2 \frac{\theta_j}{4}\right), \tag{B.27}$$

conforme calculado por Ambainis et al. (2005).

Substituindo este resultado na Equação (B.25), obtemos

$$|\langle t|\alpha^+\rangle| = \left(1 + \Theta\left(\sum_j a_j^2 \cot^2 \frac{\theta_j}{4}\right)\right)^{-\frac{1}{2}}. \tag{B.28}$$

Quando o valor do somatório é grande em comparação com 1, obtemos

$$|\langle t|\alpha^+\rangle| = \Theta\left(\min\left(\frac{1}{\sqrt{\sum_j a_j^2 \cot^2 \frac{\theta_j}{4}}}, 1\right)\right). \tag{B.29}$$

Portanto, o estado final do algoritmo, $|\alpha^+\rangle$, é uma boa aproximação para

o elemento procurado. Assim, completamos a análise do algoritmo abstrato de busca.

Isto implica em $\frac{1}{|\langle t|\alpha^+\rangle|^2}$ reptições (em média) do algoritmo para encontrar o estado buscado com probabilidade $O(1)$. Junto com a Equação (B.15), isto implica que a ordem do algoritmo de busca é $\frac{1}{\alpha} \cdot \frac{1}{|\langle t|\alpha^+\rangle|^2}$ e sua dependência com N pode ser calculada a partir do conhecimento do problema de autovalores do operador U não-modificado. Este formalismo, assim, se torna uma ferramenta poderosa para a análise de novos algoritmos de busca espacial.

Apêndice C

Simulador QWalk: instalação e comandos adicionais

A instalação do simulador QWalk em ambiente Linux (ou semelhantes) é bastante simples. Basta descarregar o código-fonte em *http://qubit.lncc.br/qwalk*, descompactá-lo em qualquer diretório, entrar no diretório *src* e finalmente usar o comando `make` para compilar o código. O código-fonte do simulador também foi compilado com sucesso no sistema operacional Microsoft Windows, usando o compilador Dev-C++ 4.9.9.2, que pode ser descarregado gratuitamente na Internet, no sítio *http://www.bloodshed.net*. Usando o comando `make doc` é possível gerar uma listagem do código-fonte dentro do diretório *doc*. Informações detalhadas de como compilar o simulador, assim como versões pre-compiladas do mesmo, também estão disponíveis no web-site. Juntamente com os arquivos descarregados, o usuário que possua conhecimento em programação C também encontra informações sobre como mudar o código-fonte.

Como vimos na Seção 5.1, o simulador QWalk consiste de três ferramentas: *qw1d* simula caminhadas quânticas em malhas unidimensionais; *qw2d*, em malhas bidimensionais; e *qwamplify* melhora a visualização dos gráficos gerados por *qw2d*, amplificando algumas regiões. Para usar o *qw1d* ou o *qw2d* é necessário escrever um arquivo de entrada em qualquer editor de texto ASCII — sem formatação. Esse arquivo de entrada consiste de palavras-chave que definem as opções de simulação. A maioria das palavras-chave importantes foram explicadas através dos exemplos

da Seção 5.1. As demais, são explicadas neste apêndice.

Após criar um arquivo de entrada — digamos que ele se chame *file.in* — basta que se digite *qw2d file.in* ou *qw1d file.in* no prompt de comando do sistema operacional utilizado, dependendo se a simulação é referente a uma caminhada unidimensional ou bidimensional. Convém ressaltar que usuários do sistema operacional Windows também devem executar o programa a partir do prompt de comando, já que um duplo-clique no nome do executável **não** faz abrir uma interface gráfica.

Os resultados da simulação ficam armazenados em alguns arquivos de saída. Iremos citar os arquivos gerados pela simulação de caminhadas bidimensionais (usando *qw2d*), mas o usuário deve ter em mente que o conjunto de arquivos gerados para simulações unidimensionais é menor.

O arquivo de saída *file.dat* contém a distribuição de probabilidades final. O arquivo de saída *file-wave.dat* contém as amplitudes complexas da função de onda no final da simulação. O arquivo de saída *file-pb.dat* contém a distribuição estacionária aproximada, quando esta é requisitada. O arquivo de saída *file-screen.dat* contém os dados observados no anteparo, quando este tipo de simulação é requisitado. O arquivo de saída *file.sta* contém certas estatísticas como variância, desvio padrão, média e a variação total da distância para uma distribuição estacionária aproximada e para a distribuição uniforme. O arquivo de saída *file.plt* é um script do gnuplot. Os arquivos postscript que ele gera dependem das opções usadas. Tipicamente esses arquivos são: *file-3d.eps*, o gráfico 3D; *file-2d.eps*, o gráfico de contorno; *file-screen.eps*, o padrão observado no anteparo; *file-pb.eps*, o gráfico da distribuição estacionária aproximada. Gráficos adicionais podem ser gerados por usuários que possuam algum conhecimento em gnuplot ou em alguma ferramenta semelhante.

O simulador QWalk possui algumas opções que não foram usadas nos exemplos citados na Seção 5.1. Passamos a descrevê-las neste apêndice. A palavra-chave `CHECK` faz com que testes de consistência sejam efetuados ao longo da simulação.

Uma sub-opção é esperada após essa palavra-chave: a sub-opção `STATEPROB` declara que a unitariedade do estado deve ser verificada após cada passo da simulação; e a sub-opção `SYMMETRY`, a simetria da distribuição de probabilidades deve ser verificada. Evidentemente, esta sub-opção somente deve ser usada quando há suspeita da simetria da distribuição de probabilidades *a priori*, a fim de confirmar ou descartar essa suspeita. Nas simulações bidimensionais, podemos selecionar dois tipos de verificação de simetria. A sub-opção `XSYMMETRY` verifica se há simetria em torno do eixo x, ou seja, verifica se a probabilidade no sítio (x, y) é a mesma probabilidade do sítio $(-x, y)$, para todo x, y na malha. Analogamente, temos as sub-opção `YSYMMETRY`.

A fim de definir uma moeda genérica, precisamos ter o comando `COIN CUSTOM` na seção principal do arquivo de entrada, além da descrição da moeda em uma seção separada do mesmo arquivo. Essa seção deve ser delimitada pelas palavras-chave `BEGINCOIN` e `ENDCOIN`, e precisa conter todas as entradas da matriz da moeda, da esquerda para a direita e de cima para baixo, com partes real e imaginária separadas por um espaço em branco. Por exemplo, o código

```
BEGINCOIN
 0.707106781186  0.0
 0.0  0.707106781186
 0.0  0.707106781186
 0.707106781186  0.0
ENDCOIN
```

define, em caminhadas unidimensionais, a moeda

$$C = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}. \tag{C.1}$$

Em simulações bidimensionais a descrição da moeda é análoga.

A fim de definir um estado genérico, precisamos ter o comando `STATE CUSTOM` na seção principal do arquivo de entrada, além da descrição do estado em uma seção separada do mesmo arquivo. Essa seção deve ser delimitada pelas palavras-chave

`BEGINSTATE` e `ENDSTATE`, e precisa conter todas as amplitudes diferentes de zero do estado inicial, com partes real e imaginária separadas por um espaço em branco. Em simulações unidimensionais, cada uma destas amplitudes precisa ser precedida por dois inteiros: o primeiro indicando a moeda e o segundo indicando a posição do caminhante. Por exemplo, o código

```
BEGINSTATE
 1 0 0.0 1.0
ENDSTATE
```

define, em caminhadas unidimensionais, o estado inicial $|\Psi_0\rangle = i\,|1\rangle\,|0\rangle$. Em caminhadas bidimensionais a descrição do estado é análoga, exceto por requerer dois inteiros para a moeda e dois inteiros para a posição do caminhante.

A palavra-chave `SEED` define manualmente uma semente para o gerador de números aleatórios. Por padrão, o simulador define essa semente a partir do relógio do sistema e geralmente o usuário não deve alterá-la.

O comando `AFTERMEASURE` define o número de iterações que serão executadas após um resultado de medição não-trivial, ou seja, após o resultado de uma medição colapsar o estado para um detector em vez de seu complemento.

A palavra-chave `LATTEXTRA` é usada raramente. Ela define um espaço adicional reservado para a malha a fim de evitar acesso a regiões inválidas de memória durante a simulação. Seu valor padrão é 1 e normalmente este valor não deve ser alterado pelo usuário. Quando este comando é usado juntamente com `LATTSIZE` e `STEPS`, existe uma ordem que deve ser respeitada: primeiro `LATTEXTRA`, depois `STEPS` e finalmente `LATTSIZE`. O uso da palavra-chave `LATTEXTRA` pode ocasionar a geração de *scripts* de gnuplot ruins. Nesse caso, o usuário deve fazer pequenas correções ao *script*, fornecendo *ranges* e *tics* adequados.

Na Tabela C.1, temos um resumo de todos os comandos permitidos pelo QWalk, tanto para malhas unidimensionais como bidimensionais.

Tabela C.1: Comandos do QWalk

Comando	Sub-opções	qw1d	qw2d
AFTERMEASURE		sim	sim
BEGINBL/ENDBL	LINE	não	sim
	POINT	não	sim
BEGINCOIN/ENDCOIN		sim	sim
BEGINSTATE/ENDSTATE		sim	sim
BLPERMANENT		não	sim
BLPROB		sim	sim
CHECK	STATEPROB	sim	sim
	SYMMETRY	sim	não
	XSYMMETRY	não	sim
	YSYMMETRY	não	sim
COIN	CUSTOM	sim	sim
	FOURIER	não	sim
	GROVER	não	sim
	HADAMARD	sim	sim
DETECTORS		sim	sim
DTPROB		sim	sim
EXPERIMENTS		sim	sim
LATTEXTRA		sim	sim
LATTYPE	CYCLE	sim	sim
	DIAGONAL	não	sim
	LINE	sim	não
	NATURAL	não	sim
	SEGMENT	sim	não
MIXTIME		sim	sim
SCREEN		não	sim
SEED		sim	sim
STATE	CUSTOM	sim	sim
	FOURIER	não	sim
	GROVER	não	sim
	HADAMARD	sim	sim