Universidade Federal de
Campina Grande
Centro de Ciências e Tecnologia

DSC
Departamento de Sistemas
e Computação

Av. Aprígio Veloso, 882 – Bodocongó
Caixa Postal 10.106
58.109-970 – Campina Grande – PB – Brasil
Fone: 3310-1119 — Fax: 3310-1273
email: dsc@dsc.ufcg.edu.br

Relatório Técnico

Nº DSC/XXX/05

O Algoritmo de Grover

Nigini A. Oliveira

UFCG/CCT/DSC
nigini@dsc.ufcg.edu.br

20 páginas

Novembro de 2005

Tutorial: O Algoritmo de Grover

Nigini A. Oliveira
E-mail: nigini@dsc.ufcg.edu.br

Departamento de Sistemas e Computação
Av. Aprígio Veloso, 882 — Bodocongó — Caixa Postal 10.106
CEP 58109-970 — Campina Grande — PB — Brasil
Fone: 3310-1119 — Fax: 3310-1273

Resumo

O algoritmo quântico de busca, também conhecido como algoritmo de Grover, é um pilar fundamental da Computação Quântica. É utilizado por vários agoritmos quânticos para melhorar suas soluções baseando-se na capacidade dele de fazer buscas em estruturas desordenadas em tempo $O(\sqrt{N})$. Neste trabalho estudamos as idéias fundamentais deste algoritmo, o simulamos e visitamos alguns casos de uso.

Palavras-chave: Computação Quântica, Informação Quântica, Algoritmos Quânticos, Grover

Abstract

The quantum search algorithm, well know as Grover's algorithm, is a Quantum Computer's fundamental structure. It is used by various quantum algorithms to improve their solutions based on its capacity of searching unsorted structures on $O(\sqrt{N})$ time. At this work we study the algorithm fundamental ideas, we simulate it and visite some using cases.

Keywords: Quantum Computation, Quantum Information, Quantum Algorithms, Grover

Sumário

Lista de Figuras

1 Introdução

O algoritmo de Grover é o algoritmo mais rápido para a busca de elementos em um conjunto desordenado. Na computação clássica, o algoritmo mais rápido para tal tarefa é da ordem $O(N)$, ou seja, em média, a busca é feita com sucesso ao examinar cada um dos elementos do conjunto.

Grover construiu um algoritmo quântico [2] que executa a tarefa com um ganho quadrático em relação ao computador clássico, ou seja, da ordem $O(\sqrt{N})$. Examinaremos neste tutorial os detalhes do algoritmo citado, como também a execução detalhada do circuito em um simulador, e faremos uma visita a exemplos de sua utilização como ferramenta na solução de problemas.

Desta forma, o que desejamos com este trabalho é, além de expor o funcionamento deste algoritmo, que é base fundamental da Computação Quântica, fazer o levantamento do estado da arte sobre o uso do mesmo.

1.1 O Problema da Busca em Estruturas Não Ordenadas

Definamos uma estrutura de dados **E** que está desordenada. A estrutura contém **N** elementos que, sem perda de generalidade, serão nomeados aqui com números de *0 a N-1*.

Para uma busca clássica em *E*, sabemos que é necessário testar todos os elementos até encontrar o(s) desejado(s). Desta forma, levamos um tempo médio de $\frac{N}{2}$ tentativas, e no pior caso N tentativas. Em termos de complexidade diríamos que o algoritmo é da ordem $O(N)$. Este mesmo problema tem solução quântica de ordem $O(\sqrt{N})$. *Como isso é possível?* Esta é uma questão que iremos examinar detalhadamente neste trabalho.

Gostaríamos de deixar claro neste ponto, que a formalização feita acima nos servirá apenas como abstração para vários problemas resumíveis a este tipo de busca. A real implementação de base de dados onde se possa aplicar esta técnica quântica não nos será pertinente por questões tecnológicas que estão fora do escopo deste trabalho. Desta forma, resumiremos nossos esforços no entendimento teórico da questão atacada e sua aplicação em problemas resumíveis a mesma.

2 O Algoritmo de Grover

O estudo do algoritmo de Grover feito neste tutorial é voltado para a montagem de um circuito que computa a resposta desejada. A estrutura de um circuito quântico não é diferente de seu equivalente clássico. Existem *qubits* de entrada, portas lógicas que computam operações unitárias, *qubits* de saída e eventualmente operações de leitura.

No circuito que montaremos na seção 3 consideraremos, por simplicidade, que o número de elementos $N = 2^n$, onde $n \in \mathbb{Z}$. Dois registradores são utilizados: um primeiro com n qubits (inicialmente no estado $|0\rangle$) e o segundo com apenas um qubit (inicialmente no estado $|1\rangle$).

Nesta seção estudaremos algebricamente as 3 principais partes do algoritmo separadamente. Desta forma a construção do circuito, partindo destes conceitos, torna-se trivial. Os 3 momentos do processo são: configuação inicial (seção 2.1.1), operações de transformação (seção 2.1.2) e rotação seletiva (seção 2.1.3). Uma visão simplista dos três passos pode ser:

1. Os estados iniciais são postos em sobreposição;
2. Marca-se o elemento desejado dentro da sobreposição;
3. Aumenta-se a amplitude do elemento marcado para que seja lido com maior probabilidade.

Os passos 2 e 3 são repetidos uma certa quantidade de vezes para a obtenção da probabilidade desejada. Esta repetição será estudada na seção 2.3.

2.1 Passo-a-Passo

Nas subseções abaixo faremos a explicação de cada um dos passos detalhadamente e mostraremos algebricamente sua funcionalidade e seus resultados. Uma observação importante é que consideramos, por questões de simplicidade, que apenas um elemento do conjunto é buscado. Mostraremos ainda, na seção 2.3, que uma generalização é possível.

2.1.1 Sobreposição dos Estados

Os estados iniciais são postos em sobreposição!

O primeiro passo é característico em algoritmos quânticos. A idéia é criar uma sobreposição de todos os estados da base com amplitudes iguais, ou seja, com a mesma probabilidade de obtenção. Este passo está intrinsecamente relacionado com a possibilidade quântica de se trabalhar com vários estados ao mesmo tempo.

Este resultado é obtido através da aplicação do operador unitário H, ou porta Hadamard, aos n qubits do primeiro registrador.

$$|\Psi\rangle = H^{\otimes n}|0,0,...,0\rangle = (H|0\rangle)^{\otimes n} = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes n} = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle \qquad (1)$$

Desta forma, enquanto os n primeiros qubits estão em sobreposição, o qubit do segundo registrador termina esta primeira fase no estado $|-\rangle$.

2.1.2 O Oráculo

O elemento desejado é marcado!

O oráculo é um operador unitário (denominado U_f) que age da seguinte forma:

$$U_f(|i\rangle\,|j\rangle) = |i\rangle\,|j \oplus f(i)\rangle\,. \tag{2}$$

O oráculo deve computar a função que identifica os elementos buscados. Podemos expressar esta função da seguinte forma:

$$f(i) = \begin{cases} 1, & \text{se i é o elemento procurado } (i_0) \\ 0, & \text{caso contrário} \end{cases}. \tag{3}$$

Se na equação 2, $|i\rangle$ é o estado do primeiro registrador e $|j\rangle$ o estado do segundo registrador temos o seguinte resultado:

$$\begin{array}{rcccl} U_f(|i\rangle\,|-\rangle) & = & U_f\left(\frac{|i\rangle|0\rangle-|i\rangle|1\rangle}{\sqrt{2}}\right) & = & \frac{U_f(|i\rangle|0\rangle)-U_f(|i\rangle|1\rangle)}{\sqrt{2}} \\ & = & \frac{1}{\sqrt{2}}(|i\rangle\,|f(i)\rangle - |i\rangle\,|1\oplus f(i)\rangle) & = & |i\rangle\left(\frac{|f(i)-|1\oplus f(i)\rangle\rangle}{\sqrt{2}}\right) \end{array} \tag{4}$$

Considerando agora as duas possibilidades de valor para a função $f(i) = \{0, 1\}$, concluimos que:

$$\begin{array}{rcl} U_f(|i\rangle\,|-\rangle) & = & \begin{cases} |i\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) = |i\rangle\,|-\rangle\,, & \text{Se f(i) = 0} \\ |i\rangle\left(\frac{|1\rangle-|0\rangle}{\sqrt{2}}\right) = |i\rangle\,(-1)\,|-\rangle\,, & \text{Se f(i) = 1} \end{cases} \\ U_f(|i\rangle\,|-\rangle) & = & (-1)^{f(i)}\,|i\rangle\,|-\rangle \end{array} \tag{5}$$

O que isso significa? Significa que apenas o elemento onde $f(i) = 1$ receberá uma fase, ou seja, o elemento buscado será marcado com um sinal negativo.

2.1.3 Rotação Seletiva

O elemento marcado tem a probabilidade de leitura aumentada!

Intuitivamente, dado que o elemento buscado já foi marcado, teríamos apenas que lê-lo. Porém o elemento encontrado ainda não será o resultado final do algoritmo, pois este possui a mesma amplitude dos outros estados. Neste passo do algoritmo, veremos como aumentar a amplitude do elemento para que sua leitura seja a de maior probabilidade dentre os estados possíveis.

A operação utilizada para tal é denominada *Transformação de Difusão* (nomeada aqui pela letra **D**). A repetição desta transformação um número bem definido de vezes (veja seção 2.3) executa a ampliação do elemento destacado, fazendo-o mais provável de ser lido.

A transformação D pode ser vista como uma *Inversão com Relação a Média* ($IRM = 2\,|\Psi\rangle\,\langle\Psi| - I$). Apesar de não ser explícito, Grover provou em [2] que IRM executa a operação D desejada. Um melhor entendimento desta questão é possível com a análise geométrica das trasformações sobre o vetor (seção 2.2).

A idéia é que o único componente negativo após o passo 2.1.2, passe a ser positivo com uma amplitude aumentada, já que, em relação a média, este elemento teria uma maior disparidade. Veja na figura 1 uma visão gráfica dos estados e perceba que apenas 1 tem amplitude negativa. Observe também que esta é uma visão simplista já que todos os outros estados estão sobre a média não são alterados. Em casos reais, todos os estados são modificados até chegarem a uma estabilidade como a exibida na figura.

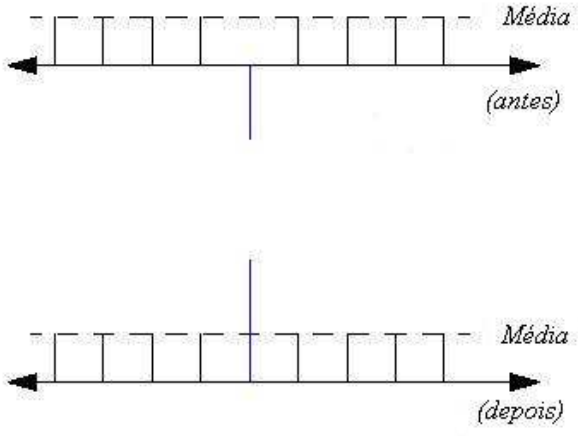


Figura 1: O funcionamento da inversão com relação a média.

2.2 Uma Visão Geométrica

Sabe-se bem que os estados quânticos e as operações sobre estes podem ser vistos como vetores e rotações no espaço de Hilbert. Nesta subseção, olharemos o comportamento do estado inicial $|\psi\rangle$ durante a execução do algoritmo de Grover, para melhor entendermos o que realmente acontece com o estado, dada a aplicação do algoritmo.

Na figura 2.1 vemos o estado inicial descrito como vetor no espaço gerado pelos vetores $|\alpha\rangle$ e $|\beta\rangle$. Consideremos o vetor $|\beta\rangle$ como sendo o vetor que define o valor a ser buscado, e $|\alpha\rangle$ o vetor ortogonal a $|\beta\rangle$, gerador do espaço. Assim, podemos descrever o estado inicial da seguinte forma: $|\Psi\rangle = a\,|\alpha\rangle + b\,|\beta\rangle$.

Na figura 2.2 observamos a rotação executada pela primeira execução do operador de Grover sobre o estado inicial. Primeiro, a execução do oráculo gera uma reflexão de $|\Psi\rangle$ com relação ao eixo $|\alpha\rangle$. Isso acontece porque o oráculo, como já foi visto, transforma o estado inicial em $O\,|\Psi\rangle = a\,|\alpha\rangle - b\,|\beta\rangle$. No segundo momento, a aplicação do terceiro passo do algoritmo ($G = (2\,|\Psi\rangle\,\langle\Psi| - I)O$) executa uma segunda reflexão, agora com relação ao estado inicial. É observado que a composição destas duas reflexões dão origem a uma rotação do estado inicial de x graus.

Por fim, na figura 2.3 vemos uma segunda execução da iteração. Com esta execução, o estado foi novamente rotacionado de mais x graus. O que queremos mostrar, lembrando do algebrismo da seção anterior, é a real ampliação do componente buscado, ou seja, a aproximação do estado inicial do eixo definido pelo vetor $|\beta\rangle$.

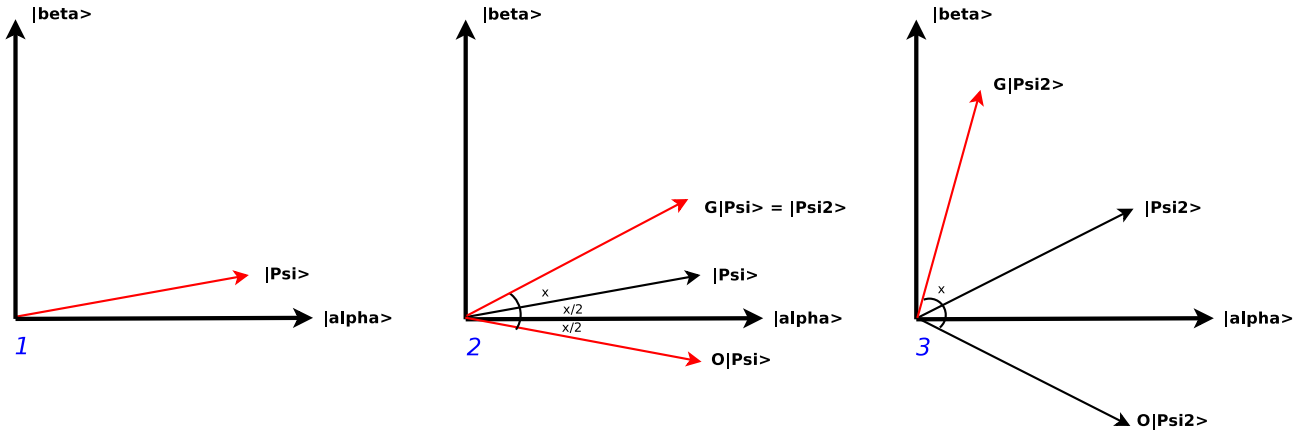


Figura 2: A evolução vetorial do estado inicial dada a aplicação iterativa do algoritmo de Grover.

2.3 Generalização e Performance do Algoritmo

Como já foi dado a entender anteriormente, o algoritmo quântico de busca é iterativo. Esta observação mostra que, dependendo da configuração inicial, será necessário mais de uma aplicação do *kernel* do algoritmo, o qual chamaremos de operação unitária **G** ou ainda, iteração de Grover. Nesta seção responderemos a seguinte pergunta: Quantas execuções são necessárias para obtermos sucesso com uma probabilidade desejada?

Para uma análise mais completa da performance do algoritmo precisamos fazer algumas generalizações. Até o momento falamos sobre a busca de um único elemento dentro do espaço possível, mas se o nosso oráculo é capaz de identificar vários elementos por vez, a generalização é trivial. Façamos então um exercício desta abstração:

- Adotemos a seguinte notação: $\sum_x^{'}$ indicando a soma dos **M** elementos que *são* solução para a busca e $\sum_x^{''}$ no caso contrário (**N-M** elementos).
- Podemos então escrever o estado inicial da seguinte forma: $|\Psi\rangle = \sqrt{\frac{N-M}{N}}\,|\alpha\rangle + \sqrt{\frac{M}{N}}\,|\beta\rangle$. Para isso precisamos escrever os estados da base considerando os somatórios do item anterior: $|\alpha\rangle = \frac{1}{\sqrt{N-M}}\sum_x^{''}|x\rangle$ e $|\beta\rangle = \frac{1}{\sqrt{M}}\sum_x^{'}|x\rangle$.
- Por fim, geometricamente falando, se rotacionarmos o estado $|\Psi\rangle$ de arccos $\sqrt{M/N}$ radianos, levaremos o sistema a $|\beta\rangle$.

Analisando esta rotação em radianos chegaremos à performance desejada que, de forma geral, pode ser expressa por:

$$R = IMP\left(\frac{arccos\sqrt{M/N}}{\theta}\right) \tag{6}$$

onde IMP significa *inteiro mais próximo* e θ é o ângulo rotacionado por cada aplicação da iteração G. Ou seja, repetindo-se o algoritmo de Grover R vezes, rotaciona-se $|\Psi\rangle$ até um ângulo interno $\theta/2 \leq \pi/4$ com o eixo $|\beta\rangle$.

Se observarmos que podemos impor limites na equação 6 também criamos limites para o número de rotações necessárias. Para isto, comecemos percebendo que $R \leq \lceil \pi/2\theta \rceil$, então um limite inferior para θ nos dará um limite superior para R. Para valores específicos de M e N saberemos as proporções do ângulo $\theta/2$ que tanto é o ângulo da primeira reflexão, como será o erro angular com relação ao eixo β após as iterações de Grover. Assumirmos uma relação média $M \leq N/2$ teremos:

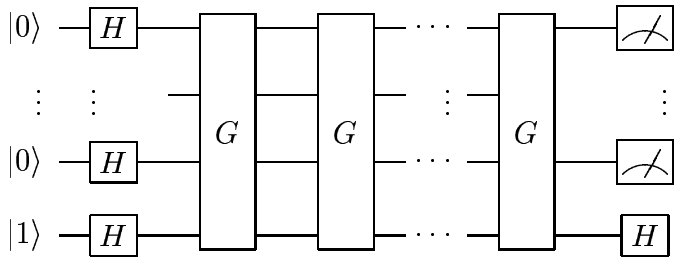
$$\frac{\theta}{2} \geq sin\frac{\theta}{2} = \sqrt{\frac{M}{N}} \Longrightarrow R \leq \left\lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rceil \tag{7}$$

Da última parte da equação 7 temos que $R = O(\sqrt{N/M})$, ou seja, o número de execuções de G do algoritmo de Grover tem um ganho quadrático com relação ao $O(N/M)$ (número de execuções do oráculo) do algoritmo clássico. Este resultado por sua vez, é provado ser o maior ganho possível para o problema de busca em dados desordenados (veja seção 6 de [2]).

3 O Circuito

Agora que já estudamos a fundamentação teórica, iremos montar o circuito quântico que computa o algoritmo quântico de busca.

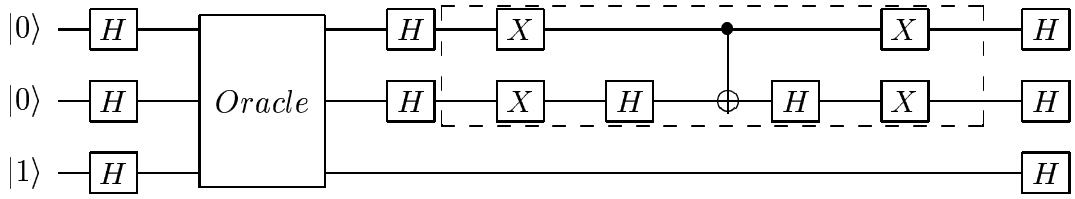
O circuito montado abaixo expressa exatamente o que foi estudado na seção 2. No primeiro instante é feita a sobreposição dos **N** estados e logo inicia-se a repetição iterativa do operador **G** que nada mais é que a composição dos passos 2.1.2 e 2.1.3.



Mas para que se possa confirmar a possibilidade de implementação deste circuito, é necessária a montagem do mesmo com portas mais básicas. Esta é a tarefa da seção 3.1. Por fim, na seção 3.2 executaremos o mesmo no simulador ZENO [1].

3.1 Um Exemplo

Por ser bastante simples e expressar todas as idéias estudadas, analisaremos o exemplo exposto no capítulo 6 do livro [4]. O exemplo faz a busca no espaço mínimo de N = 4, e para tal temos o seguinte circuito:

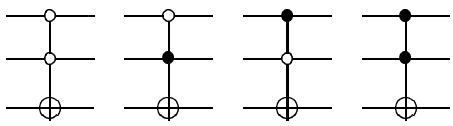


Três observações são necessárias:

1. Somente uma iteração está sendo executada, como pode ser calculado a partir das discussões da seção 2.3. Apesar de simples, este circuito já mostra o ganho com relação a um número médio de execuções de 2.25 classicamente.

2. O circuito que está destacado implementa a operação *IRM* detalhada na seção 2.1.3, responsável pela rotação seletiva do estado marcado.

3. O oráculo não está implementado pois ele varia de acordo com o elemento buscado. Veja as possíveis implementações abaixo.



Acima vemos as possíveis implementações para o oráculo, sendo $x_0 = 0, 1, 2, 3$ da esquerda para a direita.

3.2 Execução

Uma forma elegante de finalizarmos esta inspeção no algoritmo quântico de busca é fazendo a simulação do mesmo. Para esta tarefa utilizamos o simulador Zeno [1]. Abaixo, seguem alguns *snapshots* da execução com suas devidas explicações.

A simulação executada foi do mesmo circuito da seção anterior, ou seja, com N = 4. Para completar o mesmo, inserimos o oráculo para fazer a busca do elemento 3, o que é mostrado na imagem 3.

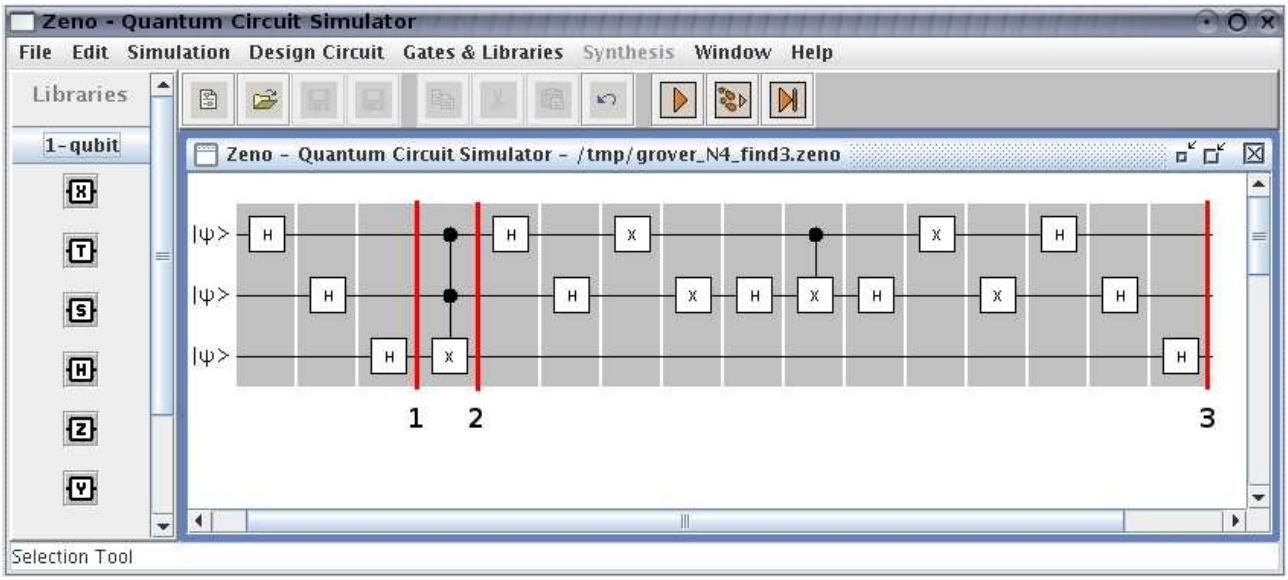


Figura 3: Circuito no simulador Zeno para o algoritmo de busca de Grover. O oráculo implementado é para a busca do elemento 3.

As barras vermelhas marcadas no circuito da figura 3 são os principais pontos do algoritmo, cujos resultados examinaremos abaixo.

Na primeira barra, temos o momento de sobreposição dos estados iniciais. Este resultado pode ser examinado na figura 4. Gostaríamos de ressaltar a pequena dificuldade que temos para ver o real resultado da busca, já que nas imagens disponibilizadas teremos também o estado do qubit auxiliar. Mas, já que sabemos que os 4 estados possíveis para o primeiro registrador são $|00\rangle$, $|01\rangle$, $|10\rangle$ e $|11\rangle$, podemos interpretar as informações das imagens da seguinte forma: vendo que o adicionamento do terceiro qubit tem apenas o efeito de dividir estes estados em dois, um positivo (pois recebeu a parte positiva do estado $|-\rangle$, que era o estado do terceiro qubit) e outro negativo (pois recebeu a parte negativa do estado $|-\rangle$).

Na figura 5 está o resultado após a execução do oráculo, ou seja, o elemento buscado neste momento deve estar marcado. Na mesma figura, marcamos as informações referentes ao estado desejado. Se compararmos com a imagem anterior veremos facilmente que o sinal do mesmo foi modificado, ou seja, o elemento foi realmente marcado.

Na figura 6 vemos claramente que apenas o estado $|111\rangle$ tem amplitude, ou seja, a leitura final terá 100% de chances de ler o estado $|11\rangle$ no primeiro registrador, ou seja, o elemento desejado de valor 3 foi encontrado.

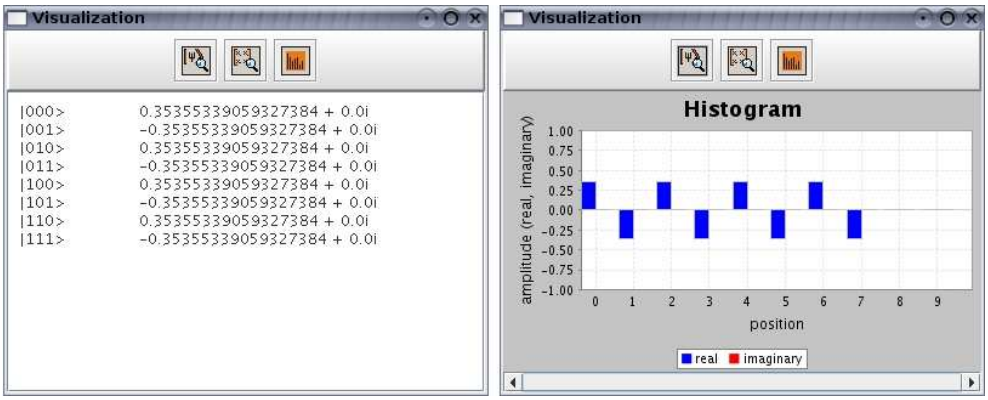


Figura 4: O resultado do primeiro passo da execução do circuito de busca no simulador Zeno.

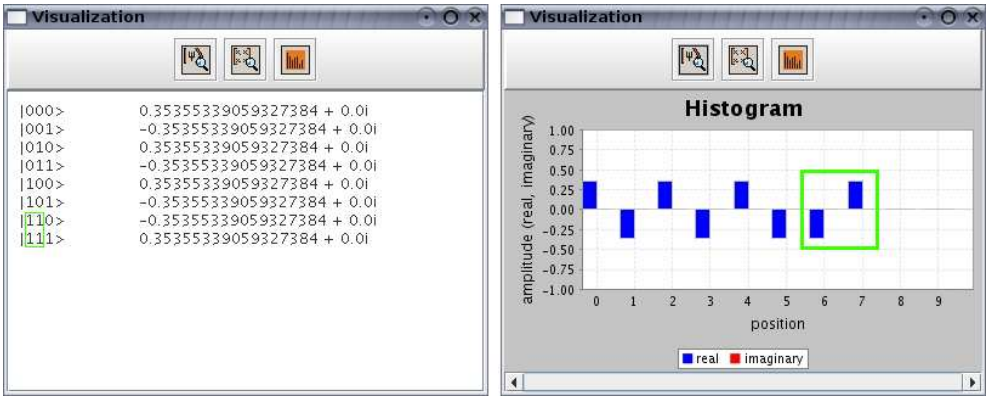


Figura 5: O resultado do segundo passo da execução do circuito de busca no simulador Zeno.

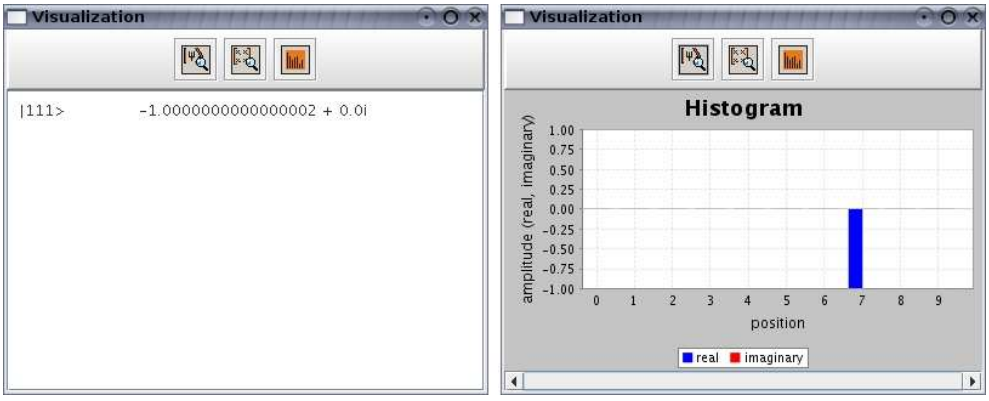


Figura 6: O resultado final da execução do circuito de busca no simulador Zeno.

4 Aplicação

O algortimo de Grover juntamente com a *Transformada Quântica de Fourier* (QFT), são os pilares fundamentais da chamada Computação Quântica. São fundamentais pois são o cerne de vários algoritmos que utilizam-se dos fundamentos da Mecânica Quântica para obter ganhos computacionais sobre as máquinas clássicas. Examinaremos aqui alguns problemas que utilizam-se de Grover para soluções mais eficientes e ainda, como este pode interagir com a QFT .

Iremos visitar o problema da fatoração, em seguida a aplicação de Grover em união com a QFT no problema da contagem de soluções, e ainda o uso deste conjunto na solução de problemas NP-Completos. Veremos por fim, um problema chamado *MDGP* (Molecular Distance Geometry Problem) que estuda a construção da geometria molecular dado as distâncias entre os átomos da mesma.

Desejamos mostrar com estes exemplos, algumas vezes de forma mais aprofundada e outra mais superficial, a importância deste algoritmo nos fundamentos da Computação Quântica.

4.1 Fatoração

Um primeiro exemplo para compreendermos melhor a estrutura do algoritmo de Grover é o da fatoração. Neste problema precisamos dizer se existem dois fatores primos (p e q) para um número, geralmente grande, m.

O método mais obvio para a solução deste problema é a busca de um primeiro número primo **p** no intervalo $[2, \sqrt{m}]$ e logo que encontrado, podemos então descobrir um valor de **q** onde $q = m/p$. Em um computador clássico este método é $O(\sqrt{m})$, mas utilizando-se do algoritmo de Grover podemos melhorar este tempo.

Para vermos isso, precisamos que o leitor lembre que o algoritmo de Grover resume-se a busca de soluções dentro de um certo conjunto. Considerando que temos este conjunto (o intervalo $[2, \sqrt{m}]$), precisamos então implementar um oráculo para o reconhecimento das soluções. Na verdade, a aplicação do algoritmo quântico de busca geralmente se resume a construção do circuito que implementa o oráculo.

Não iremos entrar no mérito da construção deste circuito aqui, mas desejamos sim que o leitor perceba dois pontos:

- a construção do circuito quântico do oráculo é um exercício de computação reversível, ou seja, dado que resolvemos o problema classicamente de forma eficiente, então, através de técnicas bem conhecidas, podemos montar o circuito quântico para computar uma função equivalente;
- o ponto chave da função computada pelo oráculo é que, mesmo sem saber a solução (no caso os fatores de m), a função pode reconhecê-la quando apareça.

Neste caso, como já sabemos que temos um ganho quadrático com a utilização da busca quântica, assintoticamente, o que é $O(m^{\frac{1}{2}})$ na computação clássica passa a ser resolvido com $O(m^{\frac{1}{4}})$ consultas ao oráculo.

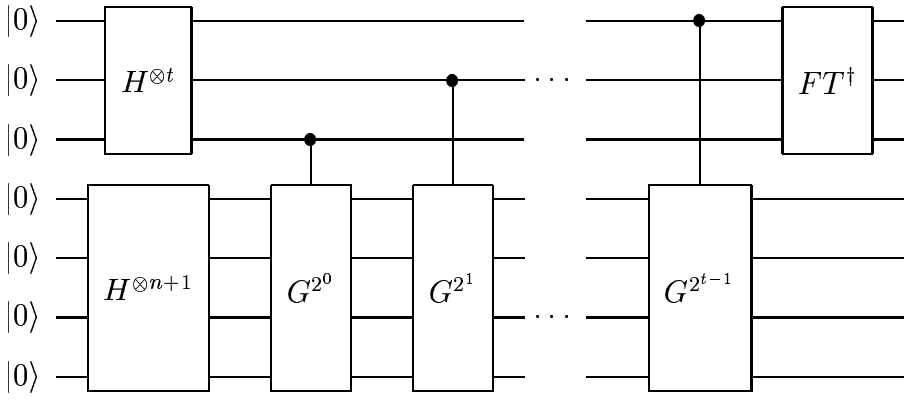
4.2 A Contagem Quântica

A contagem de soluções de um problema pode ser vista como uma importante ferramenta, e um bom exemplo do seu uso é na solução de problemas *NP-Completos*. Esta classe de problemas, no estudo da complexidade computacional, pode ser vista exatamente em termos de *buscar* a existência de uma solução. Nesta seção mostraremos, em linhas gerais, como a QFT pode ser utilizada em conjunto com o algoritmo de Grover para computar esta contagem.

A QFT está fora do escopo deste trabalho, mas uma visão simplista desta tranformada é necessária. Ela não traz ganhos diretos com relação a sua versão clássica, mas foi mostrado que ela permite que a computação da *estimação de fase* seja feita de forma mais eficiente. Esta por sua vez, é usada para resolver vários problemas como: fatoração, encontrar a ordem de funções, subgrupos escondidos e a contagem de soluções de problemas.

A estimação de fase é na verdade o cálculo de uma aproximação dos autovalores de uma transformação unitária em dadas circunstâncias. A estimação dos autovalores da iteração de Grover $\mathbf{G}$, permite a descoberta do número de soluções M para o problema da busca. O algortimo quântico de busca só pode ser aplicado com eficiencia garantida quando se sabe o valor de M. Um ganho claro da utilização da contagem é exatamente permitir a execução do algoritmo de Grover quando não conhecemos o valor de M. Isto torna o uso de Grover mais natural, já que comumente fazermos buscas sem saber a quantidade de soluções possíveis.

O circuito abaixo é um esboço do circuito que computa a contagem. Os 3 primeiros qubits representam os t qubits necessários para o cálculo da estimação de fase (função retirada do circuito QFT). Os 4 qubits seguintes representam os $n+1$ qubits necessários para a execução de Grover.



O circuito estima o ângulo de rotação das iterações do algoritmo de Grover (θ). Verificamos que, com a estimação deste ângulo, a equação 7 irá então prover-nos um valor para a quantidade de soluções M.

Nesta seção pretendíamos fazer uma exposição superficial deste problema. Para maiores informações sugerimos fortemente a leitura da seção 6.3 de [4].

4.3 Problemas NP-Completos

Um outro ganho possível com o algoritmo quântico para o problema da contagem de soluções será visto aqui. As vantagens ocorrem porque problemas NP-Completos podem ser formulados com a pergunta: Existe uma solução para o problema? Nesta seção estudaremos a solução para o denominado *Problema do Ciclo Hamiltoniano*(HC).

O problema HC é encontrar um ciclo hamiltoniano num dado grafo. Um ciclo hamiltoniano é um caminho fechado que visite todos os vértices do grafo sem repetir nenhum deles.

Um algoritmo simples para esta tarefa é:

1. Gerar todas as ordenações possíveis dos vértices $(v_1, v_2, \ldots, v_n)$;
2. Para toda ordenação, verificar se é um caminho; caso não, continue a busca.

Já que temos $n^n = 2^{n \log n}$ ordenações possíveis é óbvio que classicamente este algoritmo precisa executar $2^{n \log n}$ consultas ao oráculo no pior caso. Usaremos o algoritmo da seção 4.2 para determinar se a solução existe. Grover pode ser usado, num segundo momento, quando desejamos encontrar os possíveis ciclos.

Mas para que estudemos problemas sobre grafos, precisamos formalizar uma notação para os mesmos dentro do domínio dos algoritmos/circuitos quânticos. Podemos escrever a base computacional como o vetor $|v_1, \ldots, v_n\rangle$. Mas para representarmos cada vértice precisaremos de $m = \lceil \log n \rceil$ qubits. Desta forma, precisaremos de $m \cdot n$ qubits onde cada bloco de m qubits representará um vértice.

Podemos então expressar a função do oráculo para o algoritmo de Grover da seguinte forma:

$$O\,|v_1, \ldots, v_n\rangle = \begin{cases} |v_1, \ldots, v_n\rangle\,, & \text{se } (v_1, \ldots, v_n) \text{ não é um ciclo hamiltoniano} \\ -\,|v_1, \ldots, v_n\rangle\,, & \text{se } (v_1, \ldots, v_n) \text{ é um ciclo hamiltoniano} \end{cases} \tag{8}$$

A implementação deste circuito classicamente têm um tamanho polinomial e pode ser convertido em um circuito reversível também com tamanho polinomial. Daí, para um oráculo quântico, necessitaremos também de um circuito com um número de portas polinomial em n.

Aplicando então o algoritmo da seção 4.2 teremos $O(2^{mn/2})$ aplicações do oráculo para determinar a existência do ciclo. Desta forma, percebemos novamente um ganho quadrático com relação ao algoritmo clássico.

4.4 O Problema da Distância Molecular

O trabalho [3] mostra uma nova aplicação do algoritmo de Grover para a solução do problema MDGP (Molecular Distance Geometry Problem). Novamente é conseguido um ganho quadrático em velocidade sobre o equivalente clássico. Iremos nesta seção analisar esta aplicação.

O *MDGP* pode ser formalizado como sendo a busca de uma estrutura tridimensional de um grafo não direcionado. Levando em consideração este espaço $\mathbb{R}^3$, o problema pode

ser expresso como encontrar as coordenadas cartesianas $x_1, x_2, \ldots, x_n$ dos n átomos da molécula (onde x_n é um vetor de coordenas).

A idéia é encontrar a posição dos átomos através das informações disponíveis. Aqui, consideramos que os átomos são postos em uma lista, onde átomos vizinhos possuem uma ligação. Listados desta forma os átomos da molécula possuem no máximo 2 conexões. Apesar desta simplificação o modelo usado abaixo pode ser expandido para qualquer tipo de molécula. Os dados em questão são listados a seguir e podem ser melhor vistos na figura 7.

- S = {(i,j), todo par de átomo com distância (d_{ij}) conhecida};
- distância ente dois átomos conectados $r_{i-1,i}$; para $i = 2, \cdots, n$;
- ângulo entre conexões $\theta_{i-2,i}$ é o ângulo entre conexões atômicas $(i-2, i-1)$ e $(i-1, i)$; para $i = 3, \cdots, n$;
- ângulo de torção $\omega_{i-3,i}$ é o ângulo entre planos $(i-3, i-2, i-1)$ e $(i-2, i-1, i)$; para $i = 4, \cdots, n$.

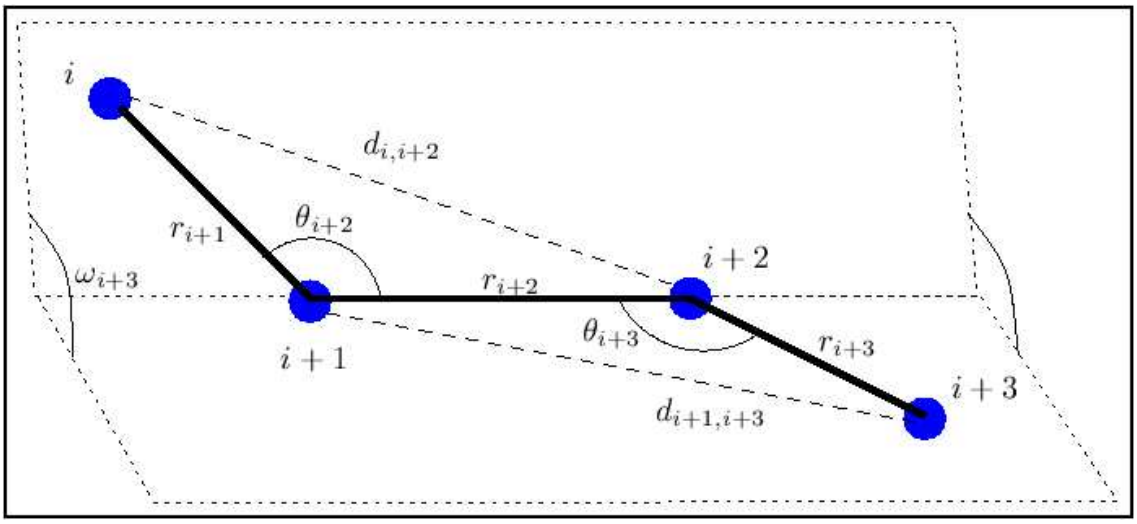


Figura 7: Uma estrutura molecular espacial e as caraterísticas descritoras do posicionamento de seus componentes.

4.4.1 Formulação Discreta de MDGP

O MDGP pode ser formulado por um problema contínuo denominado *least-squares minimization* da seguinte forma:

$$g(x_1, \cdots, x_n) = \sum (i,j) \in S(\|x_i - x_j\|^2 - d_{ij}^2)^2 \tag{9}$$

É obvio que uma sequência $(x_1^*, \cdots, x_n^*)$ é a solução quando $g(x_1^*, \cdots, x_n^*) = 0$.

O trabalho fonte define uma maneira de encontrarmos uma sequência de coordenadas, obtidos todos os valores de $\mathbf{r}$, θ e ω da sequência de átomos, procedimento este que denominaremos h. É da natureza do problema que todas as distâncias $\mathbf{r}$ e ângulos θ sejam conhecidos. Além disso, também é mostrado no trabalho, que para 4 átomos subsequentes da lista, os 3 primeiros podem ser fixados e o quarto encontrado a partir de $\omega_1 4$. Isso é feito pois, através dos r e θ relacionados, é possível encontrar o $cos(\omega_1 4)$. Para finalizar os cálculos faz-se necessário encontrar ainda o $sen(\omega_1 4)$ e então a posição do quarto átomo.

É fácil ver que, dado um $cos(\omega)$, existem apenas duas possibilidades para o $sen(\omega) = \pm\sqrt{1 - cos^2(\omega)}$. Seguindo esta idéia, para o i-ésimo átomo teremos 2^{i-3} possibilidades e ainda que, pra uma molécula de n átomos existirão 2^{n-3} possíveis estruturas tridimensionais. Só então, utilizando a função g acima citada, poder-se-á identificar a solução para o problema.

Com o cálculo dos cossenos e senos acima descritos, o MDGP pode ser considerado um problema de busca discreto (Discrite MDGP - DMDGP) e deste modo, poderemos utilizar o algoritmo de Grover para resolver o problema.

4.4.2 Aplicação de Grover

Sabendo como é o funcionamento do algoritmo de Grover (estudado na seção 2), precisamos definir apenas os dados do algoritmo no problema em questão.

O primeiro passo é definir quantos qubits são necessários. Precisamos associar um estado do conjunto de estados da base (primeiro registrador) a cada solução possível para o MDGP. Desta forma, temos $N = 2^{n-3}$, onde N é a quantidade de qubits no primeiro registrador e n a quantidade de átomos na molécula.

O segundo passo é então encontrar uma função para o oráculo equivalente à função g que nos diz qual das estruturas tridimensionais é a correta. A função segue abaixo:

$$f(i) = 1 - \left\lfloor \left(\frac{g(h(i))}{p_1} \right)^{\frac{1}{p_2}} + 0.5 \right\rfloor \tag{10}$$

O numerador $g(h(i))$ executa o cálculo da função g de verificação da sequência (least-squares minimization), sobre as coordenadas produzidas pela função h que foi definida pelo trabalho como o processo que dá a posição dos átomos dados as distâncias e ângulos. Os dois parâmetros p_1 e p_2 devem ser suficientemente grandes dependendo do número de átomos da molécula. Desta forma:

$$\frac{g(h(i))^{\frac{1}{p_2}}}{p_1} \in [0, 1] \tag{11}$$

que tomará valores muito próximos de 1, exceto quando i for associado a uma solução de DMDGP, e $f(i) = 1$ apenas quando i for solução para o problema.

Com esta aplicação de Grover para DMDGP teremos um ganho quadrático com relação a solução clássica. Os autores do trabalho [3] afirmam que há possibilidade de melhorias do algoritmo através da busca de outras propriedades estruturais do problema.

5 Conclusão

Conhecido como o algoritmo quântico de busca, desejamos ter mostrado neste trabalho porquê o algoritmo de Grover é um dos pilares fundamentais da computação quântica. Atuando muito mais como auxiliar na composição de soluções para vários problemas que como um algoritmo de busca em bases de dados reais, este algoritmo é de fundamental importântica no estudo desta área científica.

Desejamos ainda com este tutorial, que iniciantes ou até pessoas de um conhecimento intermediário em Computação Quântica possam ter expandido seus conhecimentos no estado da arte desse algoritmo.

Imaginamos que este trabalho venha a merecer novas versões, principalmente no que diz respeito a inserção de novos casos de uso, mas esperamos que esta primeira leitura seja bastante instrutiva.

6 Agradecimentos

Gostaríamos de agradecer ao CNPq e mais diretamente ao povo brasileiro que, sem saber ou poder, destina renda para que estudos como este sejam realizados.

Queremos ainda agradecer a Cheyenne Ribeiro por haver ajudado na revisão do mesmo.

Agradecemos antecipadamente a todos os leitores que desejem enviar correções ou ainda indicações de melhorias para este trabalho.

Referências

[1] Gustavo Eulalio M. Cabral. Uma ferramenta para projeto e simulação de circuitos quânticos. Master's thesis, UFCG - Universidade Federal de Campina Grande, 2004.

[2] Lov K. Grover. A fast quantum mechanical algorithm for database search. pages 212–219, 1996.

[3] Carlile Lavor, Leo Liberti, and Nelson Maculan. Grover's algorithm applied to the molecular distance geometry problem, 2005.

[4] Michael A. Nielsen and Issac L. Chuang. *Quantum Computation and Quantum Information*. 2000.