Algoritmos Quânticos

Henrique Padovani Queiroz Machado
Orientadora: Sandra Denise Prado

Introdução

O intuito da computação quântica é aumentar a velocidade de processamento com a miniaturização dos processadores. A vantagem de se desenvolver um computador quântico decorre do fato do mesmo utilizar propriedades intrínsecas da Mecânica Quântica como a superposição de estados para resolver cálculos que a computação clássica poderia levaria anos ou nem mesmo solucionar.

Caminhada Aleatória 1D

A caminhada aleatória é o resultado de uma sucessão de passos aleatórios. Na versão clássica, o caminhante dá uma passo unitário para à direita com probabilidade p ou para à esquerda com probabilidade $(1-p)$. Na versão quântica, o caminhante vai simultaneamente para à esquerda e para à direita de acordo com as amplitudes de probabilidade.

Os resultados das simulações clássica e quântica com $p = 0.5$ são mostrados na Figura 1.

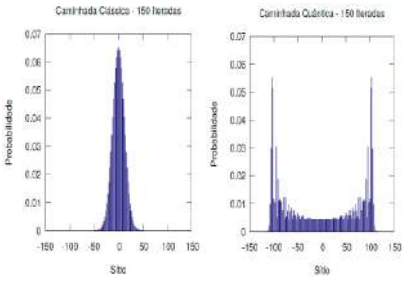

Figura 1 – Distribuição de probabilidade da caminhada aleatória clássica e quântica.

Observamos que enquanto o caminhante clássico não se afasta muito do ponto de partida, há uma maior probabilidade de se encontrar o caminhante quântico bem longe da posição inicial. Esse é um efeito da superposição quântica que traz velocidade de processamento aos algoritmos quânticos.

Na caminhada quântica definimos um estado inicial que evolui de acordo com o operador evolução temporal, que faz com que o caminhante, inicialmente localizado no sítio 0, dê um passo para à direita e para à esquerda com alguma probabilidade. Na Figura 1, o estado inicial é $|\varphi_0\rangle = 1/\sqrt{2}\,(|0\rangle|\uparrow\rangle - i\,|0\rangle|\downarrow\rangle)$. Observe que há igual probabilidade para os dois lados, por isso a distribuição de probabilidade é simétrica. Observe na Figura 2, a distribuição de probabilidade para dois estados iniciais que têm apenas um grau de liberdade inicialmente, para à esquerda (na esquerda) ou para à direita (na direita).

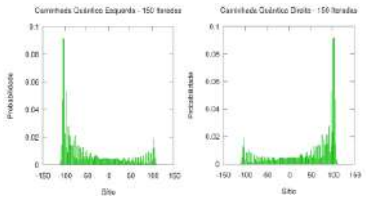

Figura 2 – Distribuição de probabilidade para os estados iniciais $|\psi_{dir}\rangle = |0\rangle|\uparrow\rangle$ e $|\psi_{esq}\rangle = |0\rangle|\downarrow\rangle$ respectivamente.

Na Figura 3 segue dois exemplos para diferentes $p = 0.93$ e $p = 0.07$.

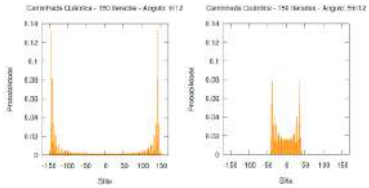

Figura 3 – Distribuição de probabilidade para diferentes operadores e estado inicial $|\varphi_0\rangle$.

Algoritmo de Grover

O algoritmo de Grover é um algoritmo de busca de um item em uma lista. Este algoritmo é inicializado com dois registradores, sendo o primeiro os q-bits necessários para armazenar os dados e o segundo registrador é um q-bit apenas.

O algoritmo inicia aplicando-se o operador Hadamard (rotação de $\pi/4$) sobre cada um dos dois registradores, para formar uma superposição de todos os estados da lista. A primeira etapa do algoritmo consiste em marcar a informação a ser pesquisada. Essa marcação é feita através do operador U_f, que é construído de acordo com a informação a ser procurada. A informação é marcada alterando-se o sinal da correspondente amplitude de probabilidade. Feito isso, a segunda etapa do algoritmo é aumentar a probabilidade de que, ao medir o primeiro registrador, o resultado seja a informação desejada. Para isso, utiliza-se um operador que reflete o estado resultante da primeira etapa em relação ao estado inicial. Estas duas etapas devem ser repetidas até que a informação desejada tenha maior probabilidade de ser obtida ao se realizar uma medida sobre o primeiro registrador.

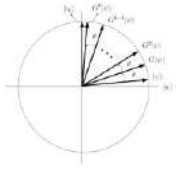

Figura 5 – Interpretação Geométrica do Algoritmo de Grover. Onde $|i_\alpha\rangle$ é a informação procurada, $|\psi\rangle$ é o primeiro registrador superposto, $|u\rangle$ são todas as outras informações da lista exceto a informação procurada, $G^k|\psi\rangle$ é o resultado da aplicação do algoritmo k vezes.

Vale ressaltar que para encontrar um dado em uma lista de N dados para N grande, algoritmos clássicos fazem, em média, N/2 operações. No algoritmo de Grover, a número procurado é obtido com alta probabilidade após $N^{1/2}$ operações.

Algoritmo de Shor

O algoritmo de Shor é um dos mais importantes algoritmos quânticos. Seu objetivo é encontrar fatores não triviais de um número inteiro N. Para isso é necessário encontrar a ordem r de um inteiro x que seja par e coprimo com N.

Ao encontrar o valor de da ordem r, existirá um valor inteiro y que satisfaz a seguinte relação:

$$x^{\frac{r}{2}} \equiv y \; mod \; N$$

Através de propriedades da aritmética modular, é possível provar que o $MDC(y-1, N)$ e $MDC(y+1, N)$ produzirá fatores não triviais de N.

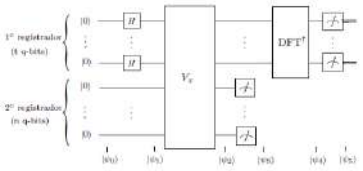

Figura 6 – Circuito do algoritmo de Shor.

Conclusão

Graças ao princípio da superposição e seu consequente paralelismo na computação quântica, os algoritmos quânticos têm larga vantagem sobre os algoritmos clássicos, quanto ao tempo de execução. Sua desvantagem é que ainda não há tecnologia que consiga manter um grande sistema de q-bits estável.

O algoritmo de Grover se torna extremamente eficiente quando se procura uma informação em uma lista muito grande. É possível provar que quando o número do elementos na lista, N, tende a infinito, o número de operações k necessárias é inferior ou igual a N. A desvantagem de se utilizar este algoritmo é que, o resultado final é uma distribuição de probabilidade em que a informação tem uma grande probabilidade de ser medida, mas não é equivalente a 100%, sendo possível obter um resultado errado.

Quanto ao algoritmo de Shor, é capaz de fatorar um número inteiro em tempo polinomial, enquanto o melhor algoritmo clássico resolve em tempo sub-exponencial. Enfim, o algoritmo de Shor torna possível decriptar todos os algoritmos de chave pública utilizados na moeda digital Bitcoin, por exemplo.

Referências

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

Uma Introdução à Computação Quântica - Renato Portugal, Carlile Campos Lavor, Luiz Mariano Carvalho e Nelson Maculan. São Carlos - SP, Brasil, 2004.