Una aplicación del algoritmo de Grover: búsqueda de dos elementos en una base de datos

Introducción a la Computación Cuántica

Bruno E. Ramírez Galindo
Facultad de Ciencias, Universidad Nacional Autónoma de México

Junio de 2020

Introducción y marco teórico

Los algoritmos de búsqueda son utilizados a diario en el manejo de bancos de datos de todos tamaños. En una base de datos con N elementos, un algoritmo clásico podrá encontrar un elemento realizando $O(N)$ operaciones. En 1996 Lov Grover introdujo un algoritmo cuántico que permite la búsqueda de un elemento realizando solamente $O(\sqrt{N})$ operaciones, presentando una ventaja del cómputo cuántico ante el clásico [1]. Este algoritmo puede ser modificado para realizar una búsqueda de más de un elemento. Por ejemplo, todos los elementos que satisfagan tener cierto atributo en una base de datos.

Para búsqueda de un número M de elementos el algoritmo de Grover comienza por generar un estado de superposición de los N estados posibles, al aplicar una compuerta $H^{\otimes n}$ al estado inicial $|0\rangle^{\otimes n}$, donde n es el número qubits necesario para representar los N elementos en la base[1]. Posteriormente, al estado en superposición

$$|\Psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{N-1} |x\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

Sea A el conjunto de estados que satisfacen tener un atributo y B el conjunto de estados que no satisfacen la búsqueda, es evidente que $\#A = M, \#B = N - M$. Dado lo anterior definimos los estado normalizados:

$$|\alpha\rangle \equiv \sum_{x\in A} |x\rangle,\ |\beta\rangle \equiv \sum_{x\in B} |x\rangle,$$

Lo anterior es útil para reescribir el estado de superposición de la siguiente forma

$$|\Psi\rangle = \frac{1}{\sqrt{N}}|\alpha\rangle + \frac{1}{\sqrt{N}}|\beta\rangle = k_0|\alpha\rangle + l_o|\beta\rangle$$

Notemos que la norma cuadrada del estado $|\Psi\rangle$ es

$$Mk_0^2 + (N-M)l_0^2 = 1$$

El algoritmo continúa aplicando dos compuertas. La primera de ellas, conocida como un oráculo, es de la forma $U_\alpha = I - 2|\alpha\rangle\langle\alpha|$, cuyo efecto será invertir el signo del estado $|\alpha\rangle$, que es la solución a la búsqueda[2]. Después se aplica la compuerta $U_S = 2|\rangle\Psi\langle\Psi| - I$. Tras aplicar j iteraciones del algoritmo, se prueba que las amplitudes para los estados $|\alpha\rangle, |\beta\rangle$ están dadas por:

$$k_j = \frac{1}{\sqrt{M}} \sin((2j+1)\theta)$$
$$l_j = \frac{1}{\sqrt{M}} \cos((2j+1)\theta)$$

donde θ es tal que $\sin^2\theta = \frac{M}{N}$. Puesto que la amplitud asociada al estado $|\alpha\rangle$ es la que se desea maximizar, se busca j tal que $l_j \approx 0$. Esto sucede si $j = (\frac{\pi}{4\theta} - \frac{1}{2})$, y si $N >> t$ entonces $\theta \approx \sin\theta = \sqrt{M/N}$ de modo que:

$$j = \frac{\pi}{4}\sqrt{\frac{N}{M}} - \frac{1}{2}$$

Se tiene que la probabilidad máxima de obtener la solución $|\alpha\rangle$ sucede después de $O(\sqrt{\frac{N}{M}})$ iteraciones. No obstante, para el caso en el que $N = 4M$ se requerirá hacer una sola iteración del algoritmo, esto ya que en este caso $\sin^2\theta = \frac{1}{4}$, por lo tanto $\theta = \pi/6$ y tenemos que la amplitud de probabilidad del estado $|\beta\rangle$ se anula [2]

$$l_1 = \frac{1}{\sqrt{N-M}} \cos(\pi/2) = 0$$

[1] El valor de n estará dado explícitamente por: $n = \lfloor \log_2(N) \rfloor + 1$
[2] En este sentido se dice que la compuerta U_α «marca la solución»

En este trabajo se utilizará lo anterior para realizar búsquedas en una base de datos de $N = 8$ entradas, donde cada búsqueda tiene $M = 2$ soluciones. Según lo presentado anteriormente, se llevará a cabo una sola iteración de $U_S U_\alpha$ para hallar las soluciones. El principal reto de construir el algoritmo de búsqueda será la implementación de las compuertas U_α que marcan a las soluciones de la búsqueda invirtiendo su signo. Adelante se presenta la implementación de dichas compuertas utilizando el software Qiskit.

Base de datos

Supongamos que se tiene la siguiente tabla en la que se muestran ocho materiales y sus correspondientes estructuras cristalinas.

# Material	Binario	Estructura
0	000	FCC
1	001	BCC
2	010	FCC
3	011	SC
4	100	BCC
5	101	SC
6	110	HCP
7	111	HCP

Cuadro 1: Base de datos con estructura cristalina por número de material, se incluye representación binaria de cada número.

Las estructuras cristalinas representan el arreglo de los átomos de un material en una celda unitaria, en este caso se presentan siguientes: cúbica simple (SC), cúbica centrada en las caras (FCC), cúbica centrada en el cuerpo (BCC) y hexagonal simple (HCP). Podemos reescribir la tabla anterior de la siguiente forma en la que se indica si un material satisface tener cierta estructura (1) o no lo hace (0).

Material	SC	FCC	BCC	HCP
000	0	1	0	0
001	0	0	1	0
010	0	1	0	0
011	1	0	0	0
100	0	0	1	0
101	1	0	0	0
110	0	0	0	1
111	0	0	0	1

Cuadro 2: Base de datos presentada por columnas indicando que material tiene alguna estructura (1) o no (0).

Esta última representación de la base de datos es útil para comenzar a construir el algoritmo de búsqueda. Esto ya que indica de manera más clara a que estados se les hará una inversión de signo según la estructura que se busque. Construiremos cuatro compuertas que hagan lo anterior, y denotaremos a cada una de estas compuertas como: $U_{CS}, U_{FCC}, U_{BCC}, U_{HCP}$.

Construyendo las compuertas U_X

Adelante se comenta a grandes rasgos el procedimiento a seguir para invertir el signo de un estado que se encuentre en superposición, en este caso $|\alpha\rangle = |A\rangle + |B\rangle$, los estados A, B son los que se buscan en cada caso. Se comienza por aplicar las compuertas X necesarias para llevar el estado A al $|111\rangle$, el estado B cambiará.

$$|A\rangle + |B\rangle \to |111\rangle + |B'\rangle$$

Posteriormente se aplica una compuerta CCZ. Dicha compuerta cambiará de signo el primero de los estados.

$$|A'\rangle + |B'\rangle \to -|111\rangle + |B'\rangle$$

Ahora se aplican las compuertas X necesarias para llevar el estado B' a $|111\rangle$

$$-|111\rangle + |B'\rangle \to -|A'\rangle + |111\rangle$$

Nuevamente se aplica la la compuerta CCZ, esta vez cambiará de signo al segundo término de la superposición

$$-|A'\rangle + |111\rangle \to -|A'\rangle - |111\rangle$$

Finalmente se aplican las dos series de compuertas X anteriormente aplicadas para llegar al estado original

$$-|A'\rangle - |111\rangle \to -|A\rangle - |B\rangle$$

Compuerta para los materiales CS

En este caso el primero de los qubits a invertir es $|011\rangle$, y el segundo $|101\rangle$. Dado lo anterior el primer bloque de X que se aplica ese solamente una sobre el qubit q_2, lo cual deja el segundo qubit en el estado $|B'\rangle = |001\rangle$. Dado lo anterior, el segundo bloque de compuertas X que se aplica actúa sobre los qubits q_1, q_2. Ambos bloques se aplican al final para regresar tener el resultado deseado

$$|011\rangle + |101\rangle \to -|011\rangle - |101\rangle$$

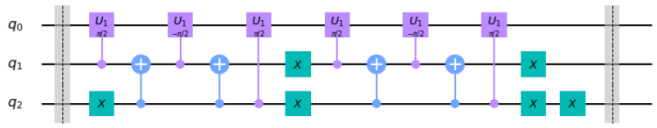

Figura 1: Circuito equivalente a la compuerta U_{CS} que invierte el signo de los estados $|011\rangle, |101\rangle$.

Compuerta para los materiales FCC

En este caso $|A\rangle = |000\rangle, |B\rangle = |010\rangle$. Esto indica que el primer bloque de compuertas X actúa sobre todos los qubits. Esto generará el estado $|B'\rangle = |101\rangle$, por lo que el segundo bloque de compuertas X consiste solamente de una compuerta actuando sobre el qubit q_1. Se aplican ambas secuencias al final para tener:

$$|000\rangle + |010\rangle \rightarrow -|000\rangle - |010\rangle$$

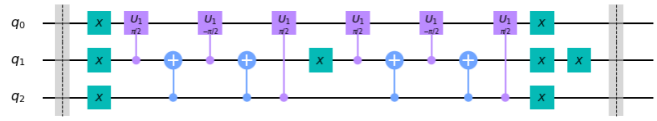

Figura 2: Circuito equivalente a la compuerta U_{FCC} que invierte el signo de los estados $|000\rangle, |010\rangle$.

Compuerta para los materiales BCC

Ahora $|A\rangle = |001\rangle, |B\rangle = |100\rangle$. Entonces el primer bloque de compuertas por aplicar actuará sobre los qubits q_2, q_1 Esto generará el estado $|B'\rangle = |010\rangle$, por lo que en el segundo bloque se aplican compuertas X sobre q_0, q_2. Se aplican las mismas secuencias de compuertas X al final y se tiene

$$|001\rangle + |100\rangle \rightarrow -|001\rangle - |100\rangle$$

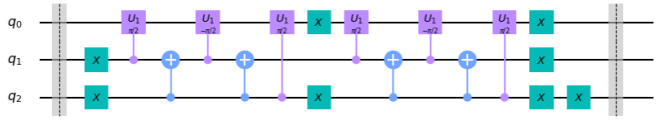

Figura 3: Circuito equivalente a la compuerta U_{BCC} que invierte el signo de los estados $|001\rangle, |100\rangle$.

Compuerta para los materiales HCP

En este último y sencillo caso $|A\rangle = |110\rangle, |B\rangle = |111\rangle$. Dado lo anterior, en el primer bloque de compuertas X solamente se aplica una sobre q_0, lo cual produce $|B'\rangle = |110\rangle$. Esto indica que el segundo bloque de compuertas X es nuevamente una compuerta X sobre el mismo qubit q_0. Al aplicar ambas secuencias al final, estas se anulan, pues se aplican dos X consecutivas. Se tiene entonces:

$$|110\rangle + |111\rangle \rightarrow -|110\rangle - |111\rangle$$

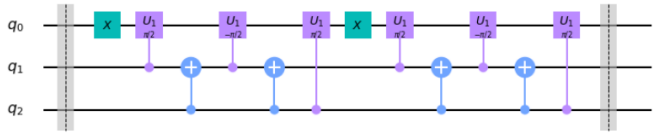

Figura 4: Circuito equivalente a la compuerta U_{HCP} que invierte el signo de los estados $|110\rangle, |111\rangle$.

Resultados

Se corrió el algoritmo de Grover utilizando las compuertas antes definidas. Éste se corrió en 1000 ocasiones para cada estructura. El número de iteraciones de la secuencia $U_S U_X$ se fijó en uno, considerando la teoría mencionada en la sección introductoria.

En cada uno de los casos, es decir para cada una de las compuertas U_X, se obtuvo satisfactoriamente que el estado final era una superposición de los estados buscados, anulando la probabilidad de los seis estados restantes. Se obtuvo en cada ocasión el estado

$$|\Psi_1\rangle \approx \frac{1}{\sqrt{2}}(|A\rangle + |B\rangle)$$

para los estados A, B correspondientes para cada estructura. Se puede confirmar el comportamiento aquí comentado revisando el documento `.ipynb` adjunto. Se adjuntan en un apéndice los resultados tras correr el algoritmo, en el que se muestran las probabilidades en cada caso.

Conclusión

De manera general, se puede establecer que las compuertas aquí implementadas permiten al algoritmo de búsqueda de Grover encontrar dos elementos en una base de datos. El funcionamiento del algoritmo depende fuertemente de la implementación de las compuertas U_X para cada caso. Inicialmente se realizaron implementaciones de dichas compuertas utilizando qubits auxiliares. Al usar dichas implementaciones se reducía la probabilidad de éxito al correr el algoritmo de Grover con una sola iteración. Se concluye que las compuertas aquí mostradas son una buena opción para la implementación del algoritmo de Grover al buscar dos elementos en una base de datos.

Bibliografía

[1] Grover, L. K. (1996, Julio). *A fast quantum mechanical algorithm for database search.* In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (pp. 212-219).

[2] Nielsen, M. A., Chuang, I. (2002). *Quantum computation and quantum information.*

[3] Boyer, M., Brassard, G., Høyer, P., Tapp, A. (1998). *Tight bounds on quantum searching* Fortschritte der Physik: Progress of Physics, 46(4-5), 493-505.

Apéndice A: Resultados del algoritmo.

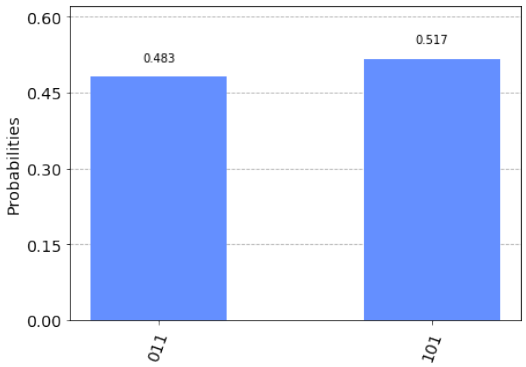

Figura 5: Histograma de estados medidos al ejecutar el algoritmo de Grover con el oráculo U_{CS}.

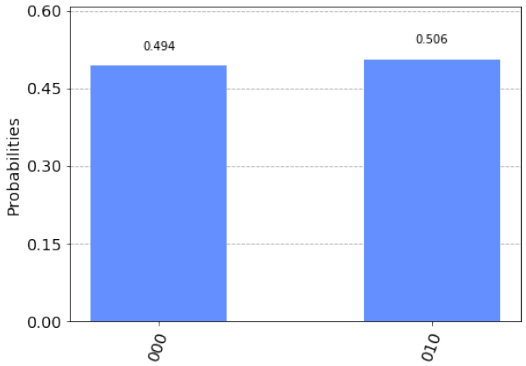

Figura 6: Histograma de estados medidos al ejecutar el algoritmo de Grover con el oráculo U_{FCC}.

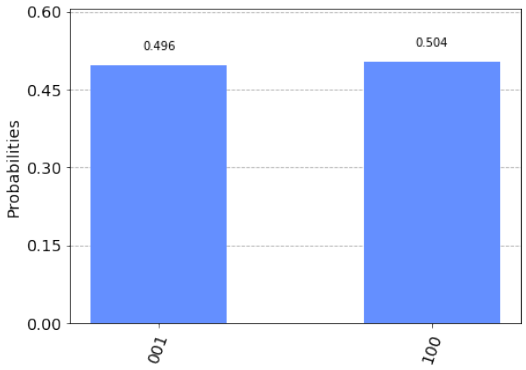

Figura 7: Histograma de estados medidos al ejecutar el algoritmo de Grover con el oráculo U_{BCC}.

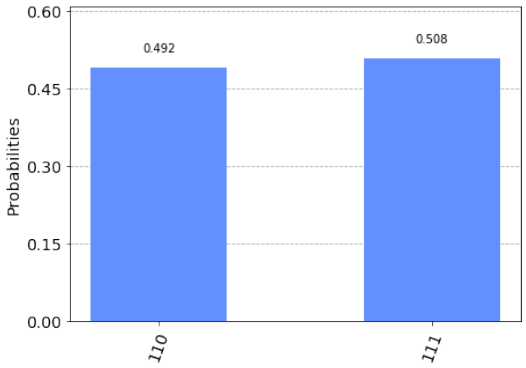

Figura 8: Histograma de estados medidos al ejecutar el algoritmo de Grover con el oráculo U_{HCP}.