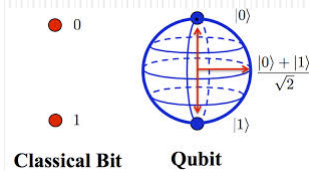


Algoritmos Cuánticos

Alfonsa García, Francisco García¹ y Jesús García¹

¹Grupo de investigación en Información y Computación Cuántica (GIICC)



Algoritmos cuánticos

1. Introducción
2. Primeros algoritmos cuánticos
 - 2.1 Problema de Deutsch
 - 2.2 Problema de Deutsch-Jozsa
 - 2.3 Problema de Simon
3. Búsqueda no estructurada. Algoritmo de Grover
4. Transformada Cuántica de Fourier
5. Algoritmo de Shor





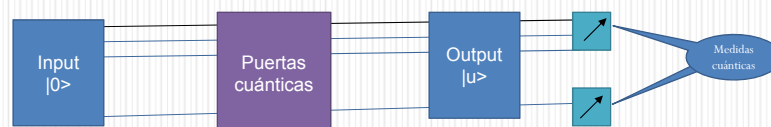
1. Introducción

- Un algoritmo es un proceso encaminado a realizar una tarea específica.
- Con frecuencia, las etapas de un algoritmo se pueden concretar en la evaluación de una función sobre distintos parámetros de entrada.
- El paralelismo cuántico permite evaluar una función simultáneamente sobre todas las posibles cadenas de n bits.
- El problema es que la información queda “oculta” en las amplitudes del estado cuántico resultante, y para acceder a ella se requiere una medición cuántica, que destruye parte de la información.



Algoritmos cuánticos → circuitos

- Un algoritmo cuántico se representa por un circuito que evoluciona de izquierda a derecha.
- El estado de entrada suele ser un n -qubit en estado $|0\rangle$ y las medidas cuánticas se hacen a la salida del circuito.





2. Primeros algoritmos cuánticos

2.1 Problema de Deutsch (ver si una función booleana de una variable es constante)

2.2 Problema de Deutsch-Jozsa (ver si una función booleana de n variables es balanceada)

2.3 Problema de Simon (determinar el periodo de una función $f: (\mathbb{Z}_2)^n \rightarrow (\mathbb{Z}_2)^n$)



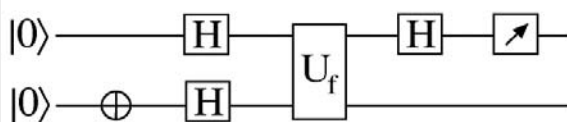
2.1 Problema de Deutsch

Dada una función booleana $f: \{0,1\} \rightarrow \{0,1\}$, se trata de ver si es constante ($f(0)=f(1)$) o no evaluándola una sola vez

Recordemos que la función f se implementa con la puerta cuántica:

$$U_f (|k\rangle \otimes |j\rangle) = |k\rangle \otimes |j \oplus f(k)\rangle$$

Resolución del problema:



Si al medir se obtiene 0 la función es constante

Una sola evaluación





2.2. Problema de Deutsch- Jozsa

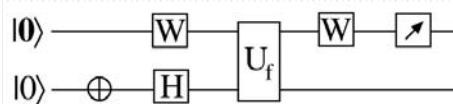
Dada $f: \{0,1\}^n \rightarrow \{0,1\}$ se trata de ver si es constante o balanceada con el menor número de evaluaciones de f

Computación clásica: Hay que evaluar f sobre la mitad más 1 de todas las posibles cadenas de n bits $\rightarrow 2^{n-1}+1$ evaluaciones

Para hacerlo cuánticamente, de nuevo usamos:

$$U_f (|k\rangle \otimes |j\rangle) = |k\rangle \otimes |j \oplus f(k)\rangle$$

Pero ahora $|k\rangle$ es un n -qubit



Si al medir el n -qubit se obtiene 0, la función es constante, si se obtiene un resultado distinto de 0 es balanceada.



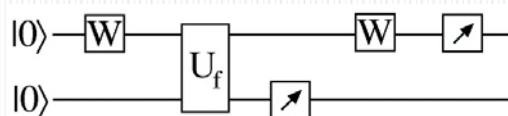
2.3. Problema de Simon

Dada $f: Z_2^n \rightarrow Z_2^n$, periódica de periodo $T \in Z_2^n$ y 2 a 1 , se trata de determinar T con el menor número posible de evaluaciones.

Computación clásica: $2^{n-1}+1$ evaluaciones

Algoritmo cuántico:

1. Inicializar el $2n$ -qubit $\Psi = |0\rangle \otimes |0\rangle$.
2. Aplicar la transformada de Walsh-Hadamard W_n a los n primeros qubits.
3. Aplicar U_f .
4. Medir los n últimos qubits: j_1, \dots, j_n (resultado $j = j_1 \dots j_n$).
5. Aplicar de nuevo W_n a los n primeros qubits.
6. Medir los n primeros qubits: k_1, \dots, k_n . Devolver $k = k_1 \dots k_n$.



Ganancia exponencial frente a los algoritmos clásicos





Seguimiento del algoritmo

$$\begin{aligned}
 |0\rangle \otimes |0\rangle &\rightarrow \frac{1}{\sqrt{2^n}} \sum_{0 \leq k < 2^n} (|k\rangle \otimes |0\rangle) \rightarrow \frac{1}{\sqrt{2^n}} \sum_{0 \leq k < 2^n} (|k\rangle \otimes |f(k)\rangle) \\
 &\rightarrow \frac{1}{\sqrt{2}} (|l\rangle + |l \oplus T\rangle) \otimes |j\rangle \quad (\text{medida } j, \Rightarrow \{l, l \oplus T\} = f^{-1}(j)) \\
 &\rightarrow \frac{1}{\sqrt{2^{n+1}}} \sum_{0 \leq k < 2^n} \left((-1)^{l \cdot k} + (-1)^{(l \oplus T) \cdot k} \right) |k\rangle \otimes |j\rangle \\
 &= \frac{1}{\sqrt{2^{n+1}}} \sum_{0 \leq k < 2^n} (-1)^{l \cdot k} \left(1 + (-1)^{T \cdot k} \right) |k\rangle \otimes |j\rangle \\
 &\rightarrow k \text{ tal que } T \cdot k = 0
 \end{aligned}$$

Obtenemos una ecuación lineal homogénea en \mathbb{Z}_2 , $T \cdot k = 0$, con n incógnitas. Hay que repetir el algoritmo hasta tener un sistema de rango $n-1$. La solución no nula del sistema será el periodo de la función



3. Búsqueda no estructurada

- Problema de búsqueda no estructurada: Hallar x en un conjunto de N posibles soluciones (no ordenadas) tal que la sentencia P sea cierta.
- Ejemplo de búsqueda no estructurada: Encontrar en una guía telefónica el titular de un número concreto.
- Se puede formular como buscar el entero x , entre 0 y $N-1$, de modo que para una función booleana f , sea $f(x)=1$.
- Computación clásica \rightarrow complejidad $O(N)$

La idea es evaluar f simultáneamente sobre los N números enteros.

Completando la lista si es preciso, supondremos $N=2^n$ y tendremos 2^n cadenas de n bits de las que sólo una verifica $f(x)=1$.





Algoritmo cuántico de Grover (1996)

- El algoritmo de Grover resuelve el problema de búsqueda no estructurada en una lista de N datos, con solución única, con $O(N^{1/2})$ evaluaciones.
- Se aprovecha el paralelismo cuántico para evaluar simultáneamente f sobre todos los posibles x de $[0, N-1]$, construyendo, con la transformación de Walsh-Hadamard, el estado:

$$\psi = W_n(|0\rangle) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

Después se va modificando este estado de modo que se incremente la probabilidad de que al medir se obtenga el x tal que $f(x)=1$



Descripción del algoritmo

Paso 1: A partir de una lista de $N=2^n$ datos ($x=0..N-1$), de modo que solo 1 verifica $f(x)=1$, se construye el estado superposición de todas las palabras de n bits: $W_n(|0\rangle)$

Paso 2 (Oráculo): Cambio de signo en la amplitud de los x tales que $f(x)=1$
 $U(|x\rangle) = (-1)^{f(x)}|x\rangle$

Se implementa por medio de:

$$U_f(|x\rangle \otimes |b\rangle) = |x\rangle \otimes |b \oplus f(x)\rangle \quad \text{con} \quad b = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Paso 3 (Inversión sobre el promedio): Si A es el promedio de las amplitudes, se transforma

$$\sum_{x=0}^{N-1} a_x |x\rangle \quad \text{en} \quad \sum_{x=0}^{N-1} (2A - a_x) |x\rangle$$

Se implementa con:

$$G = \begin{pmatrix} \frac{2}{N} - 1 & \frac{2}{N} & \dots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \dots & \frac{2}{N} \\ \dots & \dots & \dots & \dots \\ \frac{2}{N} & \frac{2}{N} & \dots & \frac{2}{N} - 1 \end{pmatrix}$$





Un ejemplo:

Partimos de una lista de 64 elementos de la que sólo 1 (x_s) satisface $f(x_s)=1$.
Construimos

$$W_n(|0\rangle) = \frac{1}{8} \sum_{x=0}^{63} |x\rangle$$

Tras cambiar el signo de la amplitud x_s el promedio de amplitudes es: $A = (63\frac{1}{8} - \frac{1}{8})/64 \approx 0.12109$

Con la inversión en el promedio la nueva amplitud de x_s es:

$$2A + \frac{1}{8} \approx 0.367817$$

Todas las demás amplitudes quedan en 0.117187. Por tanto la probabilidad de obtener x_s al medir es la más alta.

Si repetimos el proceso 6 veces, el coeficiente de x_s es 0.998291 y la probabilidad de que al medir se obtenga la solución es 0.998291².



Complejidad del algoritmo de Grover

Para una probabilidad de fallo $< 1/N \rightarrow$ número de iteraciones $O(\sqrt{N})$

Coste de implementación de $G = W_n R W_n$

$$R = 2|0\rangle\langle 0| - I$$

R es una matriz con todos sus elementos iguales a 0, salvo $R_{11}=2$, que se puede implementar con $\log(N)$ puertas de Toffoly

Coste de $W_n \rightarrow \log(N)$

Complejidad total del algoritmo: $O(\sqrt{N} \log(N))$



4. Transformada Cuántica de Fourier

La transformada cuántica de Fourier (QFT) es una transformación unitaria sobre H_n , que para un n-qubit $|j\rangle$, con $0 \leq j < n$, se define:

Sobre el n-qubit $|0\rangle$ actúa igual de la transformación de Walsh-Hadamard

$$F_n|j\rangle = \frac{1}{\sqrt{Q}} \sum_{k=0}^{Q-1} e^{2\pi i j k / Q} |k\rangle \quad Q=2^n$$

Shor (1996) introdujo un algoritmo cuántico para calcular la transformada de Fourier con $O(\log^2(Q))$ puertas cuánticas elementales

El cálculo clásico de la transformada discreta de Fourier, de una lista de tamaño Q , requiere Q^2 operaciones o $O(Q^2)$ si se usa el algoritmo FFT

GIEMATIC
Grupo de Innovación Educativa
E.U. Informática, Datos, Matemática Aplicada

La QFT y el periodo de una función

Si f es una función entera y periódica de periodo T su QFT se anula en todos los elementos del dominio salvo en los múltiplos de la frecuencia w tal que $wT=Q$


Esta propiedad permite obtener la frecuencia, y por tanto el periodo, de la función f .

$$F_n \left(\sum_{j=0}^{Q-1} f_j |j\rangle \right) = \sum_{k=0}^{T-1} \widehat{f}_{wk} |wk\rangle$$

Se aplica la QFT a f y se miden todos los qubits \rightarrow se obtiene wk , tal que $0 \leq k < T$ y tomamos $w = \text{mcd}(wk, Q)$, que será múltiplo de w .

Aplicación: El problema de factorización se puede reducir al cálculo del período de una función entera y para ello se puede usar la QFT

GIEMATIC
Grupo de Innovación Educativa
E.U. Informática, Datos, Matemática Aplicada


POLITÉCNICA
Universidad de Cádiz

5. Algoritmo de Shor

La dificultad computacional de resolver el problema de la factorización de enteros es la base del criptosistema RSA. El **algoritmo de Shor** es un algoritmo cuántico que permite factorizar un número N en tiempo $O(\text{poly}(\log(N)))$


Dado N impar y con al menos dos factores primos, el problema de encontrar un factor primo de N se puede reducir al de encontrar el orden de un entero positivo a tal que $\text{mcd}(a, N) = 1$. Es decir encontrar t tal que $a^t \equiv 1 \pmod{N}$


Algoritmo para obtener un factor propio de N :

1. Elegir aleatoriamente a entre 1 y $N-1$
2. Si $\text{mcd}(a, N) \neq 1$, devolver $\text{mcd}(a, N)$
3. Determinar t , tal que $a^t \equiv 1 \pmod{N}$
4. Si t es impar devolver fallo
5. Si $\text{mcd}(a^{t/2} + 1, N) \neq N$, devolver $\text{mcd}(a^{t/2} + 1, N)$
6. Devolver fallo

Este es el paso costoso y equivale a calcular el periodo de $f(k) = a^k \pmod{N}$

Si el algoritmo no falla devuelve un factor propio de N


Grupo de Innovación Educativa
E.U. Informática, Data, Matemática Aplicada


POLITÉCNICA
Universidad de Cádiz


Ejemplo

Supongamos que se quiere factorizar el número $N=77$

1. Elegimos un número entre 0 y 76. Por ejemplo 3.
2. Como $\text{mcd}(3, 77) = 1$, seguimos el proceso.
3. El orden de 3, en el grupo de unidades módulo 77 es 30, porque $3^{30} \pmod{77} = 1$
4. Salida: $\text{mcd}(3^{15} + 1, 77) = 7$

7 es un factor propio de 77

El paso 3, equivale a hallar el periodo de la función $f(k) = 3^k \pmod{77}$, es la parte del algoritmo en la que Shor propuso usar computación cuántica


Grupo de Innovación Educativa
E.U. Informática, Data, Matemática Aplicada



Algoritmo de Shor usando la QFT

Notación: Se busca un factor propio de N .

Se toma n tal que $N^2 \leq 2^n < 2N^2$, m tal que $N \leq 2^m < 2N$ y $Q=2^n$.

Algoritmo:

1. Elegir aleatoriamente a entre 1 y $N-1$
2. Si $\text{mcd}(a,N) \neq 1$, **devolver** $\text{mcd}(a,N)$.
3. Determinar el período T de la función $f(k)=a^k \text{ mod } N$:
 - (a) Inicializar el (n,m) -qubit: $|0\rangle \otimes |0\rangle$
 - (b) Aplicar la QFT, F_n al primer registro.
 - (c) Aplicar el operador U_f asociado a la función f .
 - (d) Aplicar nuevamente F_n al primer registro.
 - (e) Obtener la medida k y calcular la fracción continua de k/Q
 - (f) Tomar como posibles valores de T los denominadores de las convergentes de la fracción continua.
4. Para cada T , hacer:
 - (a) Si T es impar **devolver** fallo.
 - (b) Si T es par y $\text{mcd}(a^{T/2}+1,N) \neq N$, **devolver** $\text{mcd}(a^{T/2}+1,N)$
 - (c) En otro caso **devolver** fallo

Para conseguir una probabilidad de acierto independiente de T y de N es suficiente repetir el algoritmo $O(\log\log(N))$



Ejemplo Supongamos que se quiere factorizar el número $N=15$

Datos: $n=8$, $m=4$, $Q=2^n=256$

1. Elegimos un número aleatorio entre 1 y 14. Por ejemplo 13.
2. Como $\text{mcd}(13,15)=1$, seguimos el proceso.
3. Tomamos el $(8,4)$ -qubit: $|0\rangle \otimes |0\rangle$
 - (b) Aplicamos, F_n al primer registro $\rightarrow \frac{1}{16} \sum_{j=0}^{255} |j\rangle \otimes |0\rangle$
 - (c) Aplicamos $U_f \rightarrow \frac{1}{16} \sum_{j=0}^{255} |j\rangle \otimes |f(j)\rangle$
 - (d) Aplicamos F_n al primer registro $\rightarrow \frac{1}{256} \sum_{k=0}^{255} |k\rangle \otimes |A(k)\rangle$, $A(k) = \sum_{j=0}^{255} e^{2\pi i j k / Q} |f(j)\rangle$
 - (e) Supongamos que al medir el primer registro se obtiene $k=192$, consideramos la fracción continua $\frac{192}{256} = \frac{1}{1 + \frac{1}{3}}$
 - (f) Como las convergentes son $1/1$ y $3/4$, tomamos $T=4$.
4. Salida $\rightarrow \text{mcd}(13^2+1, 15)=5$



Complejidad del Algoritmo de Shor

1. Aritmética básica $\rightarrow O(\log^3(N))$
2. Elección aleatoria de un número entre 1 y $N-1 \rightarrow O(\log(N))$
3. Transformada cuántica de Fourier $O(\log^2(N))$
4. Exponenciación modular cuántica $U_f \rightarrow O(\log^3(N))$
5. Medida cuántica del primer registro $\rightarrow O(\log(N))$

El algoritmo de Shor es polinomial respecto al número de dígitos de N