UNIVERSITAT D'ALACANT
Facultat de Ciències
Facultad de Ciencias

FACULTAD DE CIENCIAS

GRADO EN FÍSICA

TRABAJO FIN DE GRADO

CURSO ACADÉMICO 2019-2020

TÍTULO:

IMPLEMENTACIÓN DEL ALGORITMO DE GROVER EN LOS ORDENADORES CUÁNTICOS DE IBM

AUTOR:

MARÍA LÓPEZ LÓPEZ

Resumen

La computación clásica forma parte de nuestro día a día y ha sido uno de los pilares que ha impulsado la innovación en las últimas décadas. Sin embargo, hay problemas que esta tecnología no es capaz de resolver o es muy poco eficiente hacerlo de esta manera. Aquí es donde entra en juego la computación cuántica, campo en el cual se están alcanzando progresos significativos. La principal diferencia entre la computación clásica y la cuántica es que en esta última se hace uso de fenómenos cuánticos, tales como la superposición y el entrelazamiento. Para ello la computación cuántica se basa en el uso de qubits o bits cuánticos, dejando de lado los bits clásicos y los sistemas lógicos empleados por los sistemas informáticos convencionales. Esta tecnología resulta de gran interés actualmente por su enorme potencial, que supone la posibilidad de revolucionar el mundo de la computación. Una misma tarea puede tener diferente complejidad en computación clásica y en computación cuántica, lo que ha dado lugar a una gran expectación. También permite la resolución de problemas intratables anteriormente. En general, los ordenadores cuánticos, debido a su gran capacidad de manejo de datos, nos ofrecen la posibilidad de impulsar grandes avances en diversos campos, desde la ingeniería hasta la investigación farmacéutica, por ejemplo.

En este trabajo se aborda el estudio del algoritmo cuántico de Grover, que resuelve el problema de encontrar una entrada en una base de datos desestructurada. En primer lugar se introducen los conceptos básicos para la comprensión del algoritmo, tales como el oráculo y la amplificación de amplitud. Posteriormente se tratan las mejoras respecto de los algoritmos de búsqueda clásicos y el hecho de que el algoritmo de Grover es óptimo. Esto completa la parte de desarrollo teórico. Posteriormente se presenta una componente de simulación en la plataforma *IBM Quantum Experience* para las diferentes situaciones propuestas. Más tarde se lleva a cabo la implementación real del algoritmo en los ordenadores cuánticos de IBM así como la implementación mitigando en cierta medida el error propio de los ordenadores cuánticos. Finalmente, se comentan algunas aplicaciones dentro del gran abanico de aplicaciones existentes a día de hoy.

Palabras clave: Algoritmo de Grover, algoritmo cuántico de búsqueda no estructurada, computación cuántica, IBM Q, Qiskit

Abstract

Classical computing is part of our daily lives and has been one of the pillars that has driven innovation in recent decades. However, some problems cannot either be solved by this technology or the above-mentioned approach is inefficient. Thereafter, quantum computing comes into play, a field of study in which significant progress is being made. The main difference between classical and quantum computing is that the latter makes use of quantum phenomena, such as superposition and entanglement. Thus, quantum computing is based on the use of qubits or quantum bits. Consequently classical bits and logical systems used by conventional computer systems are left aside. This technology is currently of great interest because of its enormous potential, which represents the possibility of revolutionizing the world of computing. A given task may present different level of complexity in classical and quantum computing, which carries a high level of expectation. This approach has allowed to achieve solutions to previously intractable problems in computational science. Overall, quantum computers, due to their great capacity of data management, allow us to trigger a breakthrough in several fields, from engineering to pharmaceutical research, for example.

This paper is focused on the study of Grover's quantum algorithm, which solves the problem of finding an input in an unstructured database. First, the basic concepts are entered so as to understand the algorithm, both the oracle and the amplification of amplitude. The improvements over classical search algorithms and the fact that Grover's algorithm is optimal are discussed. This completes the development of the theoretical part. After that, a simulation component is presented on the *IBM Quantum Experience* platform for the suggested scenarios. Later, the real implementation of the algorithm in IBM quantum computers is performed, as well as the implementation mitigating the error inherent in quantum computers. Finally, some applications are discussed within the wide range of applications existing nowadays.

Key words: Grover's algorithm, quantum unstructured search algorithm, quantum computation, IBM Q, Qiskit

Índice

1. Introducción

1.1. Conceptos básicos asociados a qubits

Un bit clásico es la unidad básica de información de la computación. Este puede tomar los valores 0 ó 1, es decir, se encuentra en un estado inequívoco. Por otra parte, un qubit o bit cuántico es la unidad unidad mínima y constitutiva de la teoría de la información cuántica. Se puede definir como un sistema cuántico de dos niveles o estados, siendo estos 0 y 1. Por tanto, el qubit es un concepto análogo al bit clásico con la diferencia fundamental de que, al tratase de un sistema cuántico, se puede definir un estado arbitrario como una superposición de estos dos niveles con diferentes pesos.

$$|\psi\rangle \ = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

La base computacional en el caso de trabajar únicamente con un qubit es:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2}$$

Cabe destacar que estos vectores que pertenecen a un espacio de Hilbert. Nótese también que estos vectores son ortonormales, formando así una base. Matemáticamente la interpretación de un estado es la de un vector de módulo unidad en un espacio vectorial complejo bidimensional, que indica que el estado está normalizado. La condición necesaria para la normalización es:

$$|\alpha|^2 + |\beta|^2 = 1 \tag{3}$$

A su vez, los coeficientes $|\alpha|^2$ y $|\beta|^2$ corresponden a la probabilidad de encontrar el sistema en uno de los estados de la base computacional. Naturalmente, la probabilidad total debe ser la unidad. La interpretación en cada caso es que, una vez medido el sistema, existe una probabilidad de $|\alpha|^2$ de encontrarlo en el estado fundamental y una probabilidad de $|\beta|^2$ de que se encuentre en el estado superior. Es decir, medir un qubit es forzar a que este se encuentre en un estado concreto.

Con el fin de comprender los conceptos anteriores y visualizar el estado de un único qubit usualmente se recurre a la representación geométrica de la figura 1. Atendiendo a dicha figura se puede reescribir la expresión 2 como:

$$|\psi\rangle \ = cos\theta|0\rangle + e^{i\varphi}sin\theta|1\rangle \tag{4}$$

donde los ángulos son números reales y definen un punto en una esfera tridimensional. Esta esfera recibe el nombre de esfera de Bloch y proporciona una sencilla manera de visualizar el estado de un qubit.

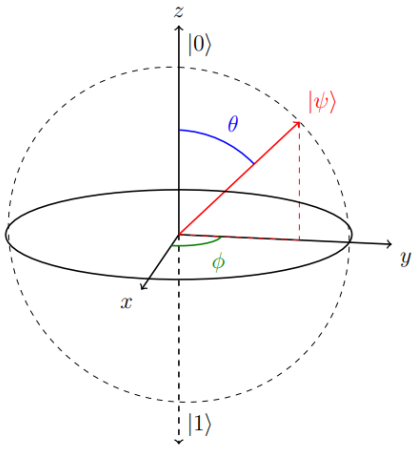


Figura 1: *Estado arbitrario de un qubit en la esfera de Bloch.*

Pese a tener infinitos puntos sobre la superficie de la esfera la información que un qubit contiene es finita. Esto se debe al propio comportamiento del qubit. Como se ha comentado anteriormente, al realizar la medida únicamente se encuentra al sistema en uno de los dos estados de la base computacional.

Si se trabaja con un número mayor de qubits se debe hacer uso de otra base computacional, una que sea adecuada. Análogamente al ejemplo anterior, si tenemos dos qubits la base es:

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{5}$$

Así, un estado arbitrario se puede expresar como superposición de estos 4 estados:

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \eta|11\rangle \tag{6}$$

con $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\eta|^2 = 1$. La interpretación es análoga al caso anterior pese a que no existe una generalización simple de la esfera de Bloch cuando la cantidad de qubits es superior.

En general, un ordenador cuántico con n qubits puede representar una superposición arbitraria de $N = 2^n$ estados simultáneamente. Es precisamente este comportamiento el que habilita una capacidad de procesamiento mucho mayor en computación cuántica. [6] [14]

1.2. Computación cuántica

Los ordenadores clásicos realizan cálculos y procesan información utilizando el modelo clásico de cómputo, donde la información se reduce a bits y el procesamiento se puede realizar a través de puertas lógicas simples. En cualquier punto del cálculo, el estado está completamente determinado por los estados de todos sus bits. En cambio, la computación e información cuántica es el estudio de las tareas de procesamiento de información que se pueden lograr utilizando sistemas cuánticos. Una computadora cuántica utiliza la mecánica cuántica para que pueda realizar el cómputo. ¿Se puede simular un circuito lógico clásico usando un circuito cuántico? La respuesta a esta pregunta es sí ya que los aspectos del mundo que nos rodea, incluidos los circuitos lógicos clásicos, pueden explicarse utilizando la mecánica cuántica. La razón por la que los circuitos cuánticos no se pueden usar para simular directamente los circuitos clásicos es que las puertas lógicas cuánticas unitarias son reversibles, propiedad que tan solo algunas puertas lógicas clásicas muestran. Por tanto, cualquier circuito clásico puede ser reemplazado por uno equivalente cuántico que contenga únicamente elementos reversibles.

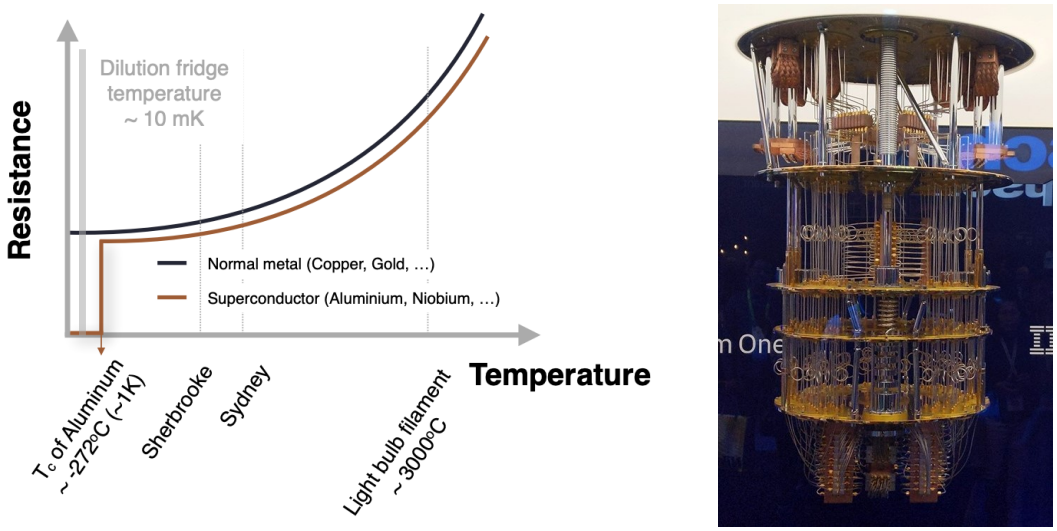
La ventaja de la computación cuántica es que pueden realizarse cierta clase de cálculos de manera más eficiente utilizando qubits y puertas cuánticas. El paralelismo cuántico es una característica fundamental en muchos algoritmos cuánticos y consiste en la superposición uniforme de los estados de la base. Así las computadoras cuánticas pueden evaluar una función para varios valores diferentes de entrada simultáneamente, lo que supone explotar así capacidad de un ordenador.

A día de hoy conocemos algoritmos cuánticos que suponen una mejora con respecto al caso clásico, como el algoritmo de Grover o la transformada de Fourier cuántica. ¿Qué otros problemas pueden resolver los ordenadores cuánticos más rápidamente que los ordenadores clásicos? La respuesta no la sabemos. El diseño de algoritmos para ordenadores cuánticos es complejo por diversos motivos y supone

un gran desafío. Primero, nuestra intuición humana está anclada al mundo clásico, por lo que para diseñar buenos algoritmos cuánticos uno debe dejar de lado dicha intuición clásica para al menos parte del proceso de diseño. Además, para que un algoritmo cuántico sea realmente interesante debe ser mejor que cualquier algoritmo clásico existente. [6]

Físicamente, en computación cuántica se utiliza el estado cuántico de un objeto para producir qubits, que generalmente son partículas subatómicas como electrones o fotones. Los ordenadores cuánticos de IBM, que son los que se utilizan posteriormente, están basados en circuitos superconductores de un tamaño del orden de micras. Se utilizan materiales como el aluminio a muy baja temperatura (del orden de mK habitualmente) que se convierten en superconductores. Los electrones entonces forman un condensado de Bose-Einstein de parejas de electrones y quedan descritos por una única función de onda. Cada qubit está formado por al menos dos superconductores con fases θ_1 y θ_2. Estos conductores, además, se encuentran separados por un material aislante de espesor atómico y esta unión túnel se conecta a un circuito LC superconductor. La energía electromagnética del circuito depende del factor $\theta_1 - \theta_2$, por lo que las fases juegan un importante papel en este contexto. Por otra parte, el Hamiltoniano efectivo se corresponde con el de un oscilador armónico con términos no lineales, por lo que el circuito presenta un espectro discreto con niveles no equidistantes. Los estados cuánticos asociados a dichos niveles describen la distribución de probabilidades de la fase de los superconductores y, por tanto, los valores de la corriente del circuito. Más concretamente, un qubit está formado por los dos estados de más baja energía del circuito superconductor. [3] [4] [14]

Generar y administrar qubits es un desafío científico a día de hoy. Otras estrategias no consisten en utilizar circuitos superconductores enfriados a muy bajas temperaturas sino que confinan átomos individuales en un chip de silicio en cámaras de ultra alto vacío, por ejemplo. En cualquier caso el objetivo es aislar los qubits para obtener un estado cuántico controlado. [3]



(a) (b)

Figura 2: *Resistencia en función de temperatura para algunos metales (a) [3], ordenador cuántico propiedad de IBM (b).*

1.3. Puertas cuánticas y símbolos de los circuitos

Los cambios que ocurren sobre un estado cuántico se pueden describir mediante el lenguaje de la computación cuántica. Un circuito cuántico es un conjunto de bits, qubits y puertas cuánticas que nos permiten transmitir y manipular la información cuántica. De esta manera, cualquier algoritmo cuántico se expresa como una secuencia de puertas lógicas cuánticas que actúan sobre uno o varios qubits. En la tabla 1 se presentan las puertas lógicas de un qubit utilizadas a lo largo del escrito.

Nombre	Símbolo	Forma matricial
Pauli-X / NOT	X	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Z	Z	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard	H	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

Tabla 1: *Puertas cuánticas que actúan sobre un qubit.*

Debemos tener en cuenta que, por el hecho de trabajar sobre un único qubit, estas puertas quedan descritas por matrices de dimensión dos en la base computacional. Acorde a la definición trabajan de la siguiente manera:

$$X|0\rangle = |1\rangle \qquad X|1\rangle = |0\rangle \tag{7}$$

$$Z|0\rangle = |0\rangle \qquad Z|1\rangle = -|1\rangle \tag{8}$$

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \qquad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{9}$$

Podemos considerar las puertas cuánticas como rotaciones en la esfera de Bloch. Por ejemplo, la puerta Hadamard se trata de una rotación de 90º sobre el eje y seguida de otra de 180º sobre el eje x.

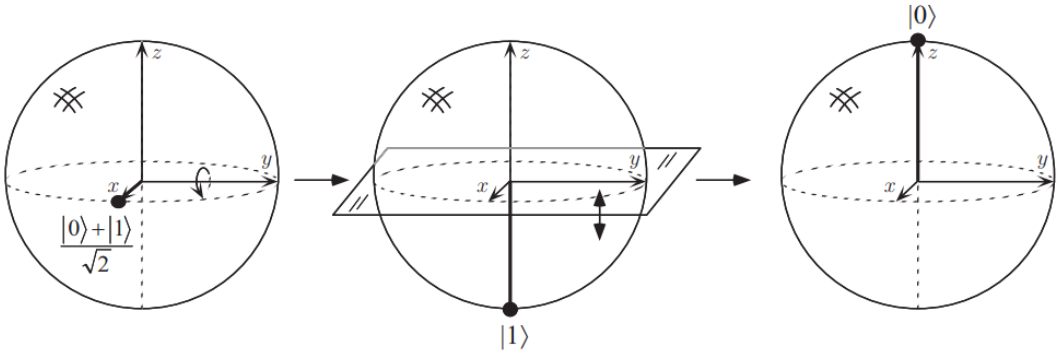


Figura 3: *Puerta Hadamard aplicada sobre el estado inicial* $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

Un requisito imprescindible para que las puertas cuánticas sean válidas es que conserven la normalización al actuar sobre un estado. Esto quiere decir que si una puerta cuántica actúa sobre un estado $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ que cumple $|\alpha|^2 + |\beta|^2 = 1$ y el resultado es $|\psi'\rangle = \alpha'|0\rangle + \beta'|1\rangle$ se debe verificar también $|\alpha'|^2 + |\beta'|^2 = 1$. Esto se traduce en que la puerta cuántica debe ser unitaria. La condición es:

$$U^\dagger U = UU^\dagger = I \tag{10}$$

donde I es la matriz unidad, U la representación matricial de la puerta y $U^\dagger$ matriz adjunta de U. Esto se puede demostrar fácilmente.

$$\langle a|b\rangle = U\langle a|U|b\rangle = \langle a|U^\dagger U|b\rangle \tag{11}$$

Como condición necesaria se obtiene la expresión 10. Esta es la única restricción sobre una puerta cuántica y conlleva que todas las operaciones que se pueden realizar en un ordenador cuántico son reversibles. Si existe una puerta U que transforma $|x\rangle$ en $|f(x)\rangle$ existe $U^{-1} = U^{\dagger}$ que devuelve el estado inicial $U^{-1}|f(x)\rangle = U^{-1}U|x\rangle = |x\rangle$. Se considera que una puerta lógica reversible se caracteriza por poder deducir cuál debe haber sido la entrada dada la salida. Por otro lado, una puerta es irreversible si dada la salida, la entrada no está totalmente determinada. Otra forma de entender la irreversibilidad es pensar en términos de información. Si una puerta lógica es irreversible se pierde parte de la información de entrada cuando la puerta funciona. Por el contrario, en un cálculo reversible, no se pierde información.

Existe además una clara conexión entre el consumo de energía y la irreversibilidad según el principio de Landauer, que enuncia que al borrar un bit de información la energía disipada es al menos $k_B\, T\, ln2$, donde k_B es la constante de Boltzmann y T la temperatura ambiente del entorno. Desde un punto de vista termodinámico la pérdida de información que se produce cuando el número de salidas es inferior al de entradas se traduce en un aumento de la entropía en la misma cantidad.

En este contexto, es importante destacar que la puerta Hadamard coincide con su adjunto, es decir, $H^{\dagger} = H$. Como consecuencia se cumple $HH = 1$, que quiere decir que para deshacer la acción de una puerta Hadamard sobre un qubit tan solo debe aplicarse de nuevo dicha puerta. Esto será de utilidad más adelante. Otra consecuencia de la propiedad anterior es la imposibilidad de implementar funciones irreversibles de un qubit con puertas que actúan únicamente sobre un qubit. Para ello debemos recurrir a puertas lógicas de dos qubits. En la tabla 2 se muestran las que se utilizan en secciones posteriores.

En este caso las puertas presentadas quedan descritas por matrices de dimensión cuatro acorde a la base computacional. El efecto de estas puertas es el siguiente:

$$CNOT|10\rangle = |11\rangle \qquad CNOT|11\rangle = |10\rangle \tag{12}$$

$$CZ|11\rangle = -|11\rangle \tag{13}$$

Al aplicar las puerta anteriores sobre los estados restantes de la base estos quedan inalterados. Nótese que estas puertas actúan sobre el qubit inferior (objetivo)

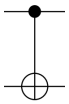
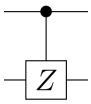
Nombre	Símbolo	Forma matricial
Controlled-NOT / CNOT		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled-Z / CZ		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$

Tabla 2: *Puertas cuánticas que actúan sobre 2 qubits.*

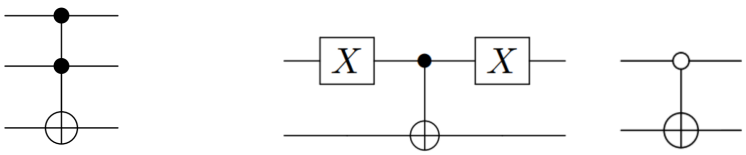
en función del estado del qubit superior (de control). Si el estado del qubit de control es el excitado sí actúan mientras que si es el fundamental el qubit objetivo queda inalterado. Como en el caso de puertas cuánticas de un qubit, las matrices son unitarias.

La única puerta de 3 qubits que se utiliza posteriormente es la puerta CCNOT. Esta es una puerta NOT doblemente controlada y recibe el nombre de puerta Toffoli. Su acción es aplicar sobre el qubit objetivo la puerta NOT si los dos qubits de control se encuentran en el estado $|1\rangle$.

Cabe destacar que cada una de las puertas anteriores admite variaciones por el hecho de ser controladas. Si el qubit de control va acompañado de una puerta NOT a la derecha y otra a la parte izquierda significa que la puerta del qubit objetivo actúa si el qubit de control es $|0\rangle$ en vez de $|1\rangle$. La demostración se encuentra en el anexo 1. Utilizamos esta característica para evitar más cálculos de los necesarios en secciones posteriores pese a que no es posible implementar las puertas modificadas directamente.

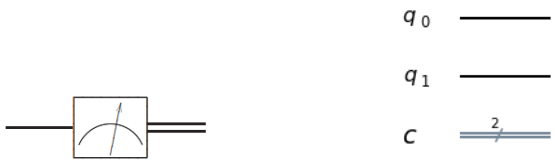
Otro de los símbolos de gran importancia, que marca el final de la mayoría de los algoritmos cuánticos, es el del operador medida. Este operador proyecta el estado cuántico sobre los estados de la base computacional. Esto significa que convierte un qubit en un bit probabilístico. Los últimos símbolos son los propios bits, en los cuales realizamos las operaciones anteriores. Tenemos el bit clásico, que se representa mediante un único cable, y el cuántico, que se presenta mediante

un par de cables. Un conjunto de n bits también se presenta simplificado como un doble cable indicando el número de bits agrupados. Cabe destacar que el tiempo representado de esta forma avanza hacia la derecha. [6]



(a) (b)

Figura 4: *Puerta Toffoli (a), equivalencia de puertas (b).*



(a) (b)

Figura 5: *Operador medida actuando sobre un qubit (a), circuito formado por dos qubits y dos bits (b).*

2. Algoritmo de Grover

2.1. Descripción del problema

El algoritmo de Grover es un algoritmo cuántico para resolver un problema de búsqueda en el que dados un total de N candidatos, únicamente $M < N$ elementos son solución del problema. Un ejemplo habitual para ilustrar el problema es el de encontrar una carta dentro de una baraja. Para trabajar se etiquetan los N candidatos con un índice en el rango $0 \leq x \leq N - 1$, de manera que no existe un patrón reconocible que permita encontrar una guía para reconocer las soluciones. Se trata por tanto de un problema desestructurado ya que no se obtiene información útil al hallar que ciertas posibilidades son incorrectas, aparte de la propia confirmación de que ese elemento no es solución. En otro casos, el hecho

de determinar que ciertos elementos no son solución permite eliminar otros y, por tanto, limitar la búsqueda de una solución. Estos problemas reciben el nombre de búsqueda estructurada.

El coste computacional de resolver el problema viene dado por el número de llamadas al oráculo, que es la diferencia más importante entre el caso clásico y el cuántico. Por tanto, cuantas menos llamadas al oráculo se necesiten más eficiente es el algoritmo de búsqueda teóricamente. Usando algoritmos clásicos, en el caso de tener N entradas se necesitan de media $N/2$ repeticiones del algoritmo para encontrar la solución. Esto se debe a la necesidad de comprobar cada una de las opciones individualmente. En el peor de los casos son necesarias N comprobaciones. El coste clásico es entonces $O(N)$. El algoritmo de Grover o algoritmo de búsqueda no estructurada permite agilizar este proceso de manera que el número de operaciones necesario es $O(\sqrt{N})$. Esto supone que el algoritmo de Grover introduce una mejora cuadrática, tal como se demuestra más adelante. En el caso de tener más de una solución clásicamente se tiene $O\left(\frac{M}{N}\right)$ mientras que cuánticamente es $O\left(\sqrt{\frac{M}{N}}\right)$, por lo que la mejora sigue siendo cuadrática. [2] [6] [9]

2.2. Estructura general del algoritmo de Grover

El algoritmo de Grover para resolver una búsqueda en una lista de elementos $x \in N = 2^n$ tiene la estructura dada por el circuito cuántico de la figura 6.

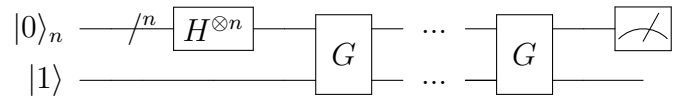


Figura 6: *Esquema del circuito del problema de búsqueda.*

En este circuito vemos dos tipos de qubits. Los n qubits de la parte de arriba, que permiten almacenar $N = 2^n$ entradas, y el qubit del espacio de trabajo. Este circuito está organizado en las siguientes etapas:

1. Preparación del estado cuántico inicial
2. Oráculo
3. Amplificación de amplitud

4. Repetición de los pasos 2 y 3 si es necesario.

5. Lectura

Los puntos 2 y 3 se engloban en llamado el operador de Grover, denotado G en la figura 6. Como se explica posteriormente, se aplican cuantas veces sea necesario. A continuación explico el funcionamiento del circuito, comenzando por el oráculo. [6]

2.3. Oráculo

Se comenta ahora uno de los puntos clave del algoritmo: el oráculo. El oráculo es una función cuyo comportamiento consiste en marcar el elemento solución del problema de búsqueda. A esta función se le llama f y toma como entrada los índices de los elementos de tal forma que devuelve un 1 si es solución y un 0 si no lo es. Se puede expresar de la siguiente manera:

$$f(x) = \begin{cases} 1 \quad si \quad |x\rangle_n = |x_w\rangle \\ \\ 0 \quad si \quad |x\rangle_n \neq |x_w\rangle \end{cases} \tag{14}$$

donde $|x_w\rangle$ hace referencia a la solución del problema y $|x_l\rangle$ al resto de elementos. Para comenzar podemos tratar al oráculo como una caja negra con la habilidad de reconocer la solución al problema. En principio se podría pensar que el oráculo es como se presenta en la figura 7.

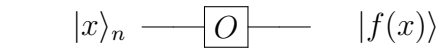


Figura 7: *Oráculo no válido.*

Sin embargo, esta propuesta para el oráculo no es válida ya que esta no es unitaria ni reversible. Una manera válida de plantear el oráculo es la que aparece en la figura 8.

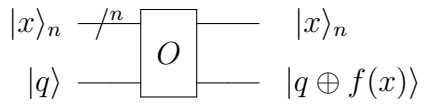


Figura 8: *Primera manera de implementar el oráculo (haciendo uso de un qubit del espacio de trabajo asociado a él).*

Así se puede establecer que el oráculo trabaja de la siguiente manera:

$$O(|x\rangle_n|q\rangle) = |x\rangle_n|q \oplus f(x)\rangle \tag{15}$$

donde $|x\rangle_n$ es el elemento del espacio de búsqueda, $\oplus$ representa la suma en módulo 2 y $|q\rangle$ es el qubit adicional de trabajo del oráculo. Como resultado de la expresión anterior, este último qubit es intercambiado si se evalúa la solución al problema y queda inalterado si no se da el caso.

Al aplicar el oráculo tal y como se ha definido es útil hacerlo habiendo inicializado $|q\rangle$ al estado $|1\rangle$. Posteriormente se aplica una puerta Hadamard, por lo que en el espacio de trabajo del oráculo se tiene:

$$H|q\rangle = H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{16}$$

Se aplica entonces el siguiente paso del oráculo, que denotamos O'. Si se trabaja con una solución al problema de búsqueda el resultado es:

$$O'|x_w\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |x_w\rangle\frac{1}{\sqrt{2}}(-|0\rangle + |1\rangle) = -|x_w\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{17}$$

De lo contrario quedaría:

$$O'|x_l\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |x_l\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{18}$$

Por tanto, se puede expresar la acción del oráculo hasta el momento de la siguiente manera:

$$O'|x\rangle_n\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = (-1)^{f(x)}|x\rangle_n\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{19}$$

Por último se aplica de nuevo una puerta Hadamard al espacio de trabajo del oráculo, completando la acción del oráculo completo O. Queda entonces:

$$(-1)^{f(x)}|x\rangle_n H\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = (-1)^{f(x)}|x\rangle_n|1\rangle \tag{20}$$

Nótese que para construir O simplemente se acompaña a O' de dos puertas Hadamard, una antes de actuar y otra después, siempre en el espacio de trabajo del oráculo. La primera puerta H se puede explicar por como se ha inicializado el qubit y la segunda devuelve el qubit al estado $|1\rangle$ para que no sufra cambios. Por tanto, la acción completa del oráculo es la siguiente:

$$O|x\rangle_n|1\rangle = (-1)^{f(x)}|x\rangle_n|1\rangle \tag{21}$$

El qubit del espacio de trabajo del oráculo no sufre modificación alguna, por lo que se puede omitir de la discusión.

$$O|x\rangle_n = (-1)^{f(x)}|x\rangle_n \tag{22}$$

La anterior descripción del oráculo es análoga a la que aparece en la figura 9.

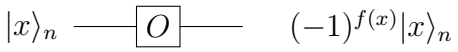


Figura 9: *Segunda manera de implementar el oráculo (en ausencia de qubits del espacio de trabajo asociado a él).*

Al implementar el oráculo de esta manera no es necesario trabajar con qubits extra en el espacio del oráculo y sí se trata de un operador válido.

Una vez realizada la discusión anterior sobre el propio funcionamiento del oráculo queda claro que se puede tratar con como una caja negra. Se podría llegar a pensar que se necesita saber cuál es el elemento solución al problema para marcarlo, lo cual no tendría sentido. El concepto clave está en que que el oráculo no sabe qué elemento es la solución pero sí es capaz de reconocerlo. A la hora de construir el oráculo sí se necesita saber la solución al problema pero no qué elemento es dicha solución. [6] [7]

2.4. Procedimiento

Se empieza aplicando aplicando una puerta Hadamard a cada uno de los n qubits del espacio de trabajo principal, que inicialmente estaban preparados en el estado $|0\rangle$. El estado inicial se denota entonces como:

$$|\psi_0\rangle \ = |0\rangle_n \tag{23}$$

mientras que el siguiente se puede escribir como superposición uniforme de los elementos de la base computacional:

$$|\psi_1\rangle \ = H^{\otimes n}|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle_n \tag{24}$$

Esta última expresión se puede obtener a partir de la definición general de la puerta de Hadamard, que es:

$$H|x\rangle = \frac{1}{\sqrt{2}} \sum_{y=0}^{1} (-1)^{xy}|y\rangle \tag{25}$$

Por tanto:

$$\begin{aligned} H^{\otimes n}|x\rangle_n = H^{\otimes n}|x_{n-1}x_{n-2}...x_o\rangle &= H|x_{n-1}\rangle...H|x_o\rangle \\ &= \frac{1}{\sqrt{2}} \sum_{y_{n-1}=0}^{1} (-1)^{y_{n-1}x_{n-1}}|y_{n-1}\rangle ... \frac{1}{\sqrt{2}} \sum_{y_o=0}^{1} (-1)^{y_o x_o}|y_o\rangle \\ &= \frac{1}{2^{n/2}} \sum_{y_{n-1}=0}^{1} ... \sum_{y_0=0}^{1} (-1)^{\sum_{j=0}^{1} x_j y_j}|y_{n-1}y_{n-2}...y_o\rangle \\ &= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} (-1)^{xy}|y\rangle_n \end{aligned} \tag{26}$$

La expresión 26 es equivalente a la expresión 24. Con ello se consigue partir de una función de onda que tiene la misma probabilidad de colapsar en cada uno de los estados de la base. Este concepto se conoce como paralelismo cuántico, ya comentado anteriormente.

Ahora hacemos una observación que resulta crucial para entender el funcionamiento del algoritmo. Es posible representar el estado de la expresión 24 como superposición lineal de dos estados ortogonales, tal y como se indica la figura 10. Matemáticamente se escribe:

$$|\psi_1\rangle = sin\theta|x_w\rangle + cos\theta|x_l\rangle \tag{27}$$

donde $|x_w\rangle$ es el estado solución (suponemos $M = 1$ para mayor claridad) y $|x_l\rangle$ es la combinación lineal uniforme de los estados no solución. Tal como se han definido los conceptos se tiene que el ángulo es:

$$\theta = arcsin\langle x_l|\psi_1\rangle = arcsin\frac{1}{\sqrt{N}} \tag{28}$$

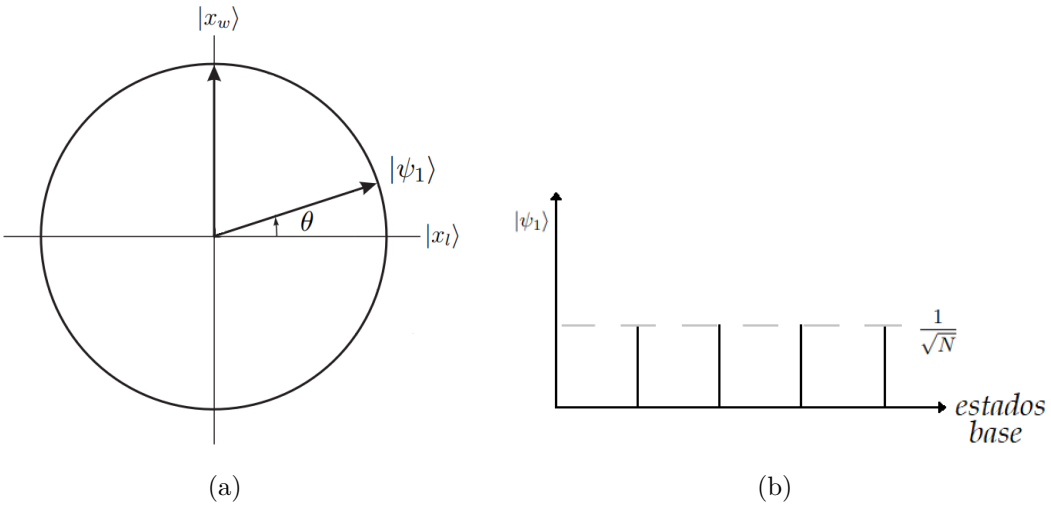


Figura 10: *Estado $|\psi_1\rangle$ en la base de los estados solución y no solución al problema de búsqueda (a), amplitudes de dichos estados (b).*

A continuación se aplica el operador de Grover tantas veces como sea necesario. Esta subrutina está compuesta por varias puertas. En primer lugar se aplica el oráculo. Tal como se ha definido su función es cambiar el signo de elemento solución al problema. Ahora no nos centramos en cual de las dos maneras de implementar el oráculo utilizar ya que el funcionamiento es análogo. El estado resultado de esta operación es:

$$|\psi_2\rangle \ = O|\psi_1\rangle \ = -sin\theta|x_w\rangle + cos\theta|x_l\rangle \tag{29}$$

Geométricamente, el oráculo realiza una reflexión del vector $|\psi_1\rangle$ alrededor de $|x_l\rangle$, como indica la figura 11. La amplitud del estado $|x_w\rangle$ es ahora negativa. Como consecuencia, la media de las amplitudes disminuye pese a no afectar a los demás valores.

A continuación se debe introducir una puerta Hadamard en cada uno de los n qubits del espacio de búsqueda. Seguidamente se aplican el operador cambio de fase y, de nuevo, una puerta Hadamard a cada uno de los n qubits. El operador cambio de fase introduce una fase $e^{i\pi} = -1$ a cada estado excepto al fundamental.

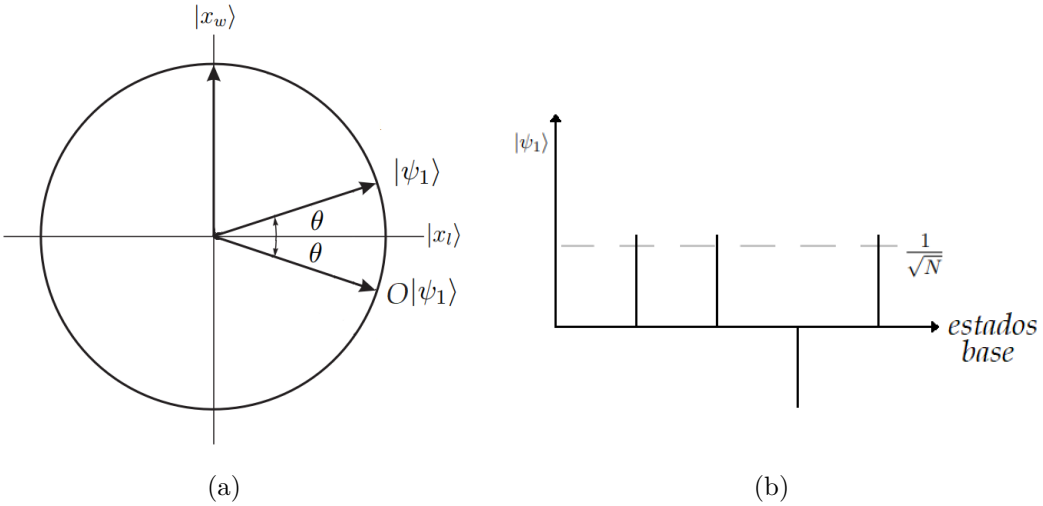


Figura 11: *Estado $|\psi_2\rangle$ en la base de los estados solución y no solución al problema de búsqueda (a), amplitudes de dichos estados (b).*

La función del operador cambio de fase es la siguiente:

$$S|0\rangle_n = |0\rangle_n \qquad S|x\rangle_n = -|x\rangle_n \tag{30}$$

donde $|x\rangle_n$ hace referencia al resto de estados de la base computacional en este contexto. Una sencilla manera de expresar esta puerta es la siguiente:

$$S = 2|0\rangle_n\langle 0|_n - I \tag{31}$$

Con las puertas Hadamard:

$$H^{\otimes n}SH^{\otimes n} = H^{\otimes n}(2|0\rangle_n\langle 0|_n - I)H^{\otimes n} = 2|\psi_1\rangle\langle\psi_1| - I \tag{32}$$

Aplicando $J = 2|\psi_1\rangle\langle\psi_1| - I$ sobre un estado general $|\psi\rangle = \sum_k \alpha_k|k\rangle_n$ se obtiene:

$$\begin{aligned}(2|\psi_1\rangle\langle\psi_1| - I)\left(\sum_k \alpha_k|k\rangle_n\right) &= 2\sum_k \alpha_k|\psi_1\rangle\langle\psi_1|k\rangle_n - \sum_k \alpha_k|k\rangle_n \\ &= 2\sum_k \frac{\alpha_k}{N}\sum_x\sum_{x'}|x\rangle_n\langle x'|_n|k\rangle_n - \sum_k \alpha_k|k\rangle_n \\ &= 2\sum_k \frac{\alpha_k}{N}\sum_x|x\rangle_n - \sum_k \alpha_k|k\rangle_n \\ &= \sum_k[-\alpha_k + 2\langle\alpha\rangle]|k\rangle_n\end{aligned} \tag{33}$$

Anteriormente se ha tenido en cuenta que el valor medio de α_k viene dado por la expresión $\langle\alpha\rangle = \sum_k \frac{\alpha_k}{N}$. La operación $J = (2|\psi_1\rangle\langle\psi_1| - I)$ es la inversión sobre el promedio y se conoce como como operador inversión. Para facilitar la visualización geométrica se desarrolla la acción de este operador sobre el estado $|\psi_2\rangle$.

$$\begin{aligned} |\psi_3\rangle &= J|\psi_2\rangle \\ &= (2|\psi_1\rangle\langle\psi_1| - I)|\psi_2\rangle \\ &= 2(-sin^2\theta + cos^2\theta)|\psi_1\rangle - |\psi_2\rangle \\ &= (-4sin^3\theta + 3sin\theta)|x_w\rangle + (4cos^3\theta - 3cos\theta)|x_l\rangle \\ &= sin(3\theta)|x_w\rangle + cos(3\theta)|x_l\rangle \end{aligned} \tag{34}$$

Esto es análogo a decir que J realiza una reflexión sobre el vector $|\psi_1\rangle$, como se indica en la figura 12.

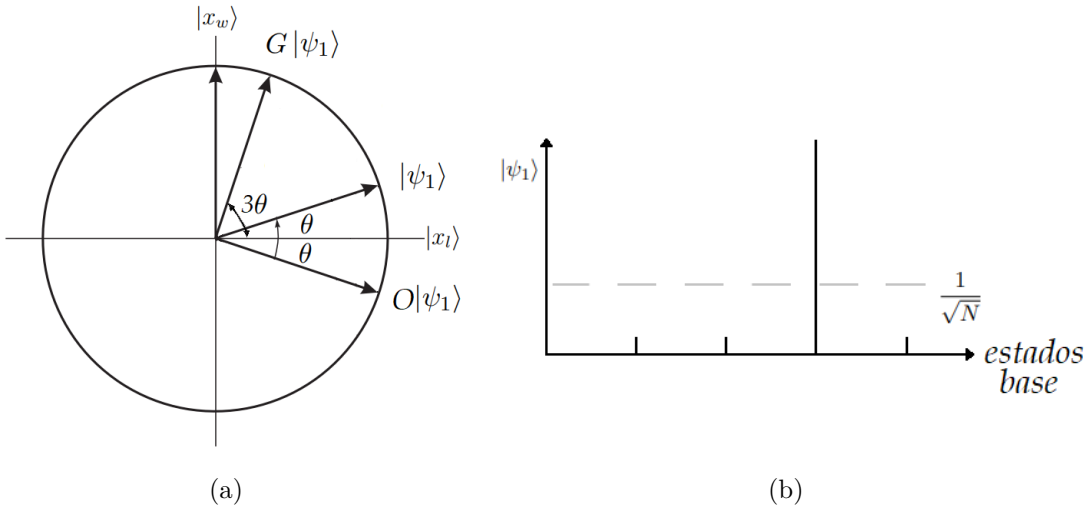


Figura 12: *Estado $|\psi_3\rangle$ en la base de los estados solución y no solución al problema de búsqueda (a), amplitudes de dichos estados (b).*

Todos estos pasos corresponden al operador de Grover, que se puede expresar como:

$$G = JO = (2|\psi_1\rangle\langle\psi_1| - I)O \tag{35}$$

La representación del operador de Grover en el circuito se muestra en la figura 13.

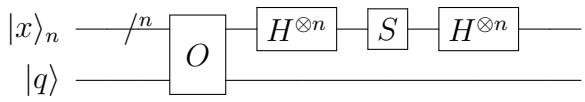


Figura 13: *Esquema del operador de Grover.*

Cabe destacar que en el caso de utilizar el qubit extra del espacio de trabajo del oráculo debe inicializarse de la manera comentada.

De acuerdo a la definición del operador de Grover tenemos que se trata de una rotación en el plano definido por $|x_w\rangle$ y $|x_l\rangle$ al ser el producto de dos reflexiones en ese mismo plano. Dicha rotación es de 2θ radianes (ver figura 12a) y tiene como resultado desplazar el estado inicial de superposición hacia el vector solución. Esto se traduce en una mayor probabilidad de medir el estado solución una vez aplicado el operador G al aumentar la amplitud del estado correspondiente (ver figura 12b). Puede darse el caso de que con una única iteración no se obtenga un resultado satisfactorio, por lo que convendría aplicar el operador de Grover repetidas veces. Tras k aplicaciones del operador de Grover el estado resultante es:

$$G^k|\psi_1\rangle = sin\Big((2k+1)\theta\Big)|x_w\rangle + cos\Big((2k+1)\theta\Big)|x_l\rangle \tag{36}$$

La deducción de esta expresión se encuentra en el anexo 6.2. Teóricamente, para obtener el resultado deseado con total probabilidad se debe cumplir $sin\Big((2k+1)\theta\Big) = 1$, lo que implica que el número de repeticiones óptimo es:

$$k = \frac{\pi}{4}\frac{a}{\theta} - \frac{1}{2} \tag{37}$$

con $a \in \mathbb{Z}$. Este es el número de iteraciones que maximiza la amplitud del estado solución y un valor menor o mayor de este parámetro disminuye la probabilidad de éxito, es decir, de medir el estado deseado.

En el caso general ($N > M \geq 1$) se escribe:

$$|x_w\rangle = \frac{1}{\sqrt{M}}\sum_{x'}|x'\rangle \tag{38}$$

$$|x_l\rangle = \frac{1}{\sqrt{N-M}}\sum_{x''}|x''\rangle \tag{39}$$

donde $|x_w\rangle$ es una combinación lineal uniforme de los estados solución de la base y $|x_l\rangle$ una combinación lineal uniforme del resto de estados. Por tanto:

$$|\psi_1\rangle = \sqrt{\frac{N-M}{N}}|x_w\rangle + \sqrt{\frac{M}{N}}|x_l\rangle \tag{40}$$

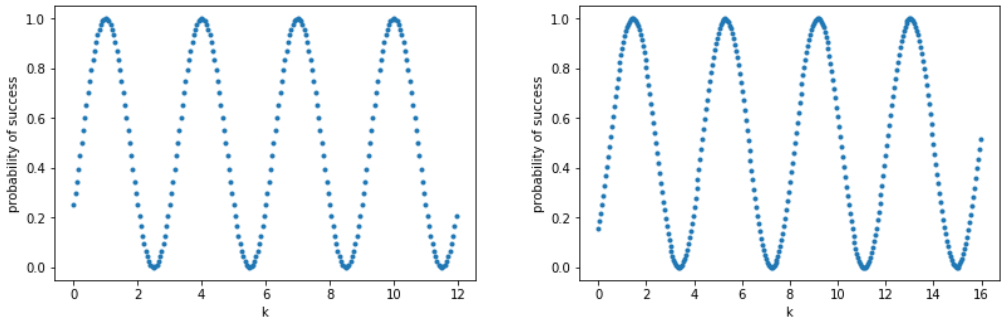
Siguiendo los mismos pasos que en el caso $M = 1$ se llega de nuevo a la expresión 36. El número de iteraciones queda inalterado pero se ha de tener en cuenta que $\theta = arcsin\langle x_l|\psi_1\rangle = arcsin\sqrt{\frac{M}{N}}$. Con esta expresión y la fórmula 37 se puede escribir el límite superior del número de repeticiones óptimo:

$$k \leq \frac{\pi}{4}\sqrt{\frac{N}{M}} \tag{41}$$

al tener

$$\theta \geq sin\theta = \sqrt{\frac{N}{M}} \tag{42}$$

Como resultado se tiene entonces que el número de veces que se debe llamar al operador de Grover (y con ello al oráculo) es $O(\sqrt{\frac{N}{M}})$. Esto supone una mejora cuadrática respecto de las $O(\frac{N}{M})$ llamadas del algoritmo clásico.



(a) (b)

Figura 14: *Probabilidades de éxito en el caso de $N = 2^2$ con $M = 1$ (a) y en el caso $N = 2^5$ con $M = 5$ (b).*

Con todo lo anterior se construye el circuito que permite resolver el problema de búsqueda tratado [2] [6] [9]. El esquema básico se presenta de nuevo a continuación.

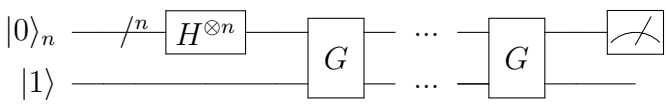


Figura 15: *Esquema de la solución al problema de búsqueda.*

2.5. Algoritmo óptimo

El algoritmo de Grover no es simplemente un primer ejemplo de un algoritmo cuántico que permite resolver el problema de búsqueda no estructurada. Este algoritmo es óptimo, es decir, que cualquier otro que resuelva el mismo problema debe llamar al oráculo como mínimo el mismo número de veces. A continuación se demuestra que ningún otro algoritmo cuántico puede realizar la misma tarea usando menos de $\Omega(\sqrt{N})$ llamadas al oráculo, demostrando así que el algoritmo de Grover es óptimo.

Por simplicidad se realiza la demostración en el caso de tener una única solución ($M = 1$), que se denota simplemente como $|x\rangle$ para simplificar la notación. Recordemos que para determinar el estado solución se debe aplicar el oráculo, que se puede expresar como $O = I - 2|x\rangle\langle x|$. Partiendo del estado $|\psi_1\rangle$ se definen:

$$|\psi_k^x\rangle = J_k O J_{k-1} O ... J_1 O |\psi_1\rangle \tag{43}$$

$$|\psi_k\rangle = J_k J_{k-1} J_1 |\psi_1\rangle \tag{44}$$

De este modo, $|\psi_k^x\rangle$ es el estado resultante al aplicar el oráculo k veces y operaciones unitarias intercaladas $J_1, J_2, ..., J_k$ entre los oráculos. Por otra parte, $|\psi_k\rangle$ es el resultado de aplicar únicamente los operadores J. Se define también una medida de la desviación producida por el oráculo tras k aplicaciones como:

$$D_k = \sum_x ||\psi_k^x - \psi_k||^2 \tag{45}$$

Para simplificar la notación a partir de este momento se omiten los kets de los estados. En primer lugar se demuestra:

$$D_k \leq 4k^2 \tag{46}$$

por inducción. Para $k = 0$:

$$D_0 = \sum_x ||\psi_0^x - \psi_0||^2 = 0$$

Se considera entonces que la expresión 46 es cierta para k. Para $k + 1$:

$$D_{k+1} = \sum_x ||\psi_{k+1}^x - \psi_{k+1}||^2 = \sum_x ||U_{k+1} O \psi_k^x - U_{k+1} \psi_k||^2 = ||U_{k+1}|| \sum_x ||O\psi_k^x - \psi_k||^2$$

$$= \sum_x ||O\psi_k^x - \psi_k||^2 = \sum_x ||O(\psi_k^x - \psi_k) + (O - I)\psi_k||^2$$

Por otra parte:

$$(O - I)|\psi_k\rangle = |\psi_k\rangle - 2|x\rangle\langle x|\psi_k\rangle - |\psi_k\rangle = -2\langle x|\psi_k\rangle|x\rangle \tag{47}$$

Con la expresión 47 y haciendo uso de la desigualdad $||a+b||^2 \leq ||a||^2 + ||b||^2 + 2||a||\ ||b||$ con a $\equiv O(\psi_k^x - \psi_k)$ y b $\equiv (O-I)\psi_k$ queda:

$$D_{k+1} \leq \sum_x (||\psi_k^x - \psi_k||^2 + 4||\psi_k^x - \psi_k||\ |\langle x|\psi_k\rangle| + 4|\langle x|\psi_k\rangle|^2)$$

Sobre el segundo término se aplica la desigualdad de Cauchy-Schwarz, que se formula como $\Big(\sum_i a_i b_i\Big)^2 \leq \Big(\sum_i a_i^2\Big)\Big(\sum_i b_i^2\Big)$. El resultado es:

$$D_{k+1} \leq D_k + 4\Big(\sum_x (||\psi_k^x - \psi_k||^2\Big)^{1/2}\Big(\sum_x |\langle\psi_k|x\rangle|\Big)^{1/2} + 4$$

Haciendo uso de la hipótesis de inducción y la propiedad $\sum_x |\langle x|\psi_k\rangle|^2 = 1$ se obtiene:

$$D_{k+1} \leq D_k + 4\sqrt{D_k} + 4 \leq 4k^2 + 8k + 4 = 4(k+1)^2$$

Esta última expresión completa la inducción, por lo que podemos tomar la expresión 46 como válida. Con esto termina la primera parte de la demostración de que el algoritmo de Grover es óptimo. Queda probar que D_k debe ser $\Omega(N)$ para que la probabilidad de éxito sea aceptable. A continuación se trabaja con una probabilidad de éxito de al menos un 50 %, por lo que se toma $|\langle x|\ \psi_k^x\rangle|^2 \geq \frac{1}{2}$. Reemplazar $|x\rangle$ por $e^{i\theta}|x\rangle$ no altera la probabilidad de éxito, por lo que se asume $|\langle x|\ \psi_k^x\rangle| = \langle x|\ \psi_k^x\rangle$ sin pérdida de generalidad. Se puede escribir:

$$||\psi_k^x - x||^2 = 2 - 2|\langle x|\ \psi_k^x\rangle| \leq 2 - \sqrt{2} \tag{48}$$

Se define ahora $E_k = \sum_x ||\psi_k^x - x||^2$, por lo que se cumple:

$$E_k \leq (2 - \sqrt{2})N \tag{49}$$

Se define también $F_k = \sum_x ||x - \psi_k||^2$. Partiendo de la definición de D_k y desarrollando, se obtiene:

$$D_k = \sum_x ||\psi_k^x - \psi_k||^2 = \sum_x ||(\psi_k^x - x) + (x - \psi_k)||^2$$

$$\geq \sum_x ||(\psi_k^x - x)||^2 - 2\sum_x ||x - \psi_k||\ ||\psi_k^x - x|| + \sum_x ||x - \psi_k||^2$$
$$= E_k + F_k - 2\sum_x ||x - \psi_k||\ ||\psi_k^x - x||$$
$$\geq E_k + F_k - 2\sqrt{E_k F_k} = \left(\sqrt{F_k} - \sqrt{E_k}\right)^2 \tag{50}$$

donde se han aplicado la desigualdad de Cauchy-Schwarz y la propiedad $||a + b||^2 \geq ||a||^2 + ||b||^2 - 2||a||\ ||b||$ con a $\equiv (\psi_k^x - x)$ y b $\equiv (x - \psi_k)$. Se pretende ahora hallar un límite inferior para F_k.

$$F_k = \sum_x ||x - \psi_k||^2 = \sum_x \left(||x||^2 + ||\psi_k||^2 - 2Re(\langle x|\psi_k\rangle)\right)$$
$$\geq \sum_x \left(||x||^2 + ||\psi_k||^2 - 2|\langle x|\psi_k\rangle|\right) = 2N - 2N|\langle x|\psi_k\rangle|$$
$$= 2N - \frac{2N}{\sqrt{N}} = 2N - 2\sqrt{N} \tag{51}$$

Combinando las expresiones 49, 50 y 51 se obtiene:

$$D_k \geq N\left(\sqrt{2 - \frac{2}{\sqrt{N}}} - \sqrt{2 - \sqrt{2}}\right) \tag{52}$$

Por tanto, se puede escribir:

$$D_k \geq cN \tag{53}$$

para N suficientemente grande, donde c es una constante que cumple $c < \left(\sqrt{2} - \sqrt{2 - \sqrt{2}}\right)^2$. Recuperando 46 y 53 se obtiene finalmente:

$$k \geq \sqrt{\frac{cN}{4}} \tag{54}$$

Este resultado indica que cualquier algoritmo de búsqueda que realice su función con probabilidad de éxito de al menos 50 % debe hacer $\Omega(\sqrt{N})$ llamadas al oráculo. Este es el caso del algoritmo de Grover, por lo que se puede afirmar es óptimo. [6]

3. Implementación en ordenadores cuánticos

En este capítulo discuto los circuitos cuánticos para implementar del algoritmo de Grover en el caso específico de $N = 4$ entradas, tanto en el caso con $M = 1$ (una única solución) como en el caso de tener varias soluciones. Finalmente también muestro un ejemplo con $N = 8$ y $M = 2$.

3.1. Implementación para una única solución

En el caso de trabajar con 2 qubits en el espacio de búsqueda se tienen $N = 2^2 = 4$ estados posibles. Cada uno puede tomarse como solución, por lo que existen diversas elecciones del oráculo según el elemento escogido. A continuación se presenta una tabla con diversas maneras de realizar la implementación correspondiente. [6] [9]

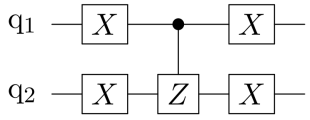
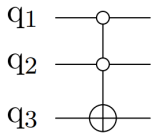
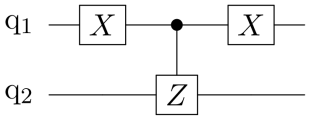
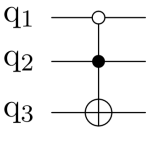
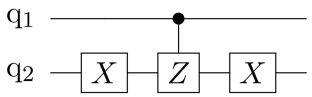
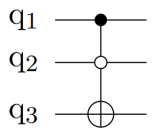
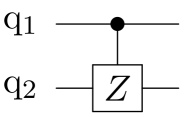
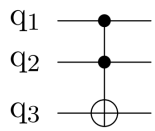
Solución	Operador de fase O	Operador booleano O'
$\|00\rangle$		
$\|01\rangle$		
$\|10\rangle$		
$\|11\rangle$		

Tabla 3: *Oráculos utilizando y sin utilizar el qubit extra del espacio de trabajo del mismo operador para cada una de las soluciones. Caso $N = 4$ y $M = 1$.*

El operador cambio de fase no depende de la solución deseada, por lo que la única dependencia viene dada por el número de qubits. De nuevo, tenemos varias maneras de implementar dicho operador. [6]

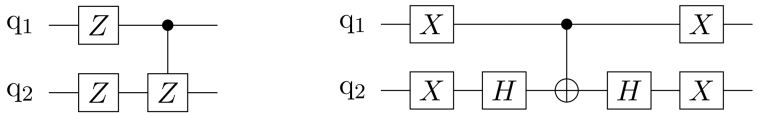


Figura 16: *Diferentes maneras válidas de implementar el operador cambio de fase. Caso $N = 4$.*

Más adelante se centra la atención en el estado $|00\rangle$ como solución y se realizan dos implementaciones: haciendo uso y prescindiendo del qubit extra. Las representaciones de los circuitos implementados son las siguientes:

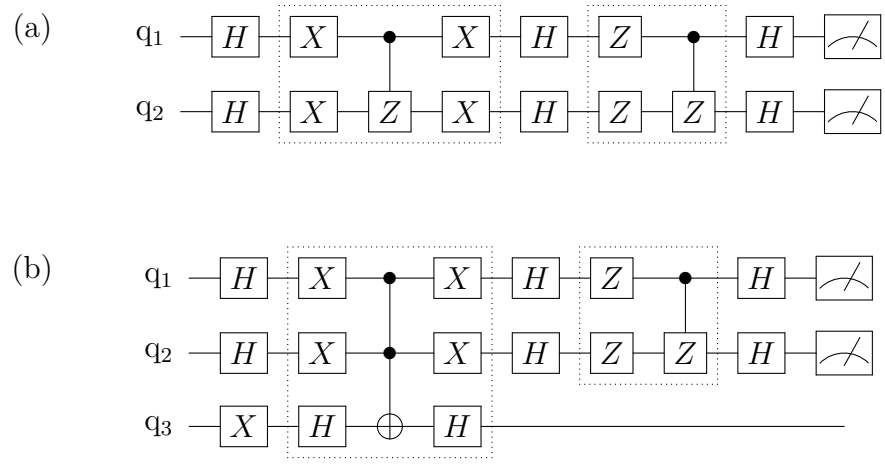


Figura 17: *Circuitos implementando $n = 2$ y $M = 1$ con solución $|00\rangle$. Sin hacer uso del qubit extra (a) y utilizando un qubit en el espacio de trabajo del oráculo (b). Los elementos marcados de cada circuito son el oráculo y el cambio de fase, en ese orden. [6]*

De ahora en adelante al circuito de la figura 17a se le conoce como escenario 1 y al de la figura 17b como escenario 2. Nótese que en la figura anterior los circuitos implementados llaman una única vez al operador de Grover. Esto se debe a que en el caso elegido ($N = 4$ y $M = 1$) tan solo es necesaria una llamada al operador de Grover para obtener la solución deseada con probabilidad máxima según la expresión 37. La demostración de la validez de los oráculos y cambio de fase presentados se encuentra en los anexos 3, 4 y 5.

3.2. Implementación para varias soluciones

El objetivo de esta sección es mostrar que sí es posible tener como solución más de un estado de la base computacional. Comenzamos con 2 qubits en el espacio de búsqueda y 2 soluciones ($N = 4, M = 2$) y se nombra como escenario 3. Para este caso se presenta como ejemplo el circuito de la figura 18. [6]

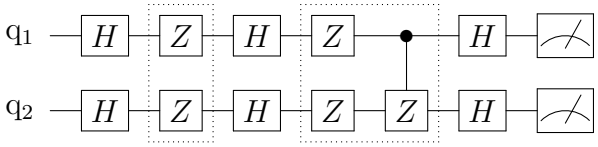


Figura 18: *Circuito implementando $n = 2$ y $M = 2$ con soluciones $|01\rangle$ y $|10\rangle$. Sin hacer uso del qubit extra. Los elementos marcados son el oráculo y el cambio de fase, en ese orden.*

Según la expresión 37 el número óptimo de veces que el circuito debe llamar al oráculo no es entero. Por ejemplo, si $a = 1$ se tiene $k = \frac{1}{2}$ y si $a = 2$ se tiene $k = \frac{3}{2}$. Como resultado, como se comenta más adelante, la probabilidad de éxito es baja. Se presenta entonces otro circuito que permita obtener más de una solución al problema de búsqueda. Se trabaja con 3 qubits en el espacio de búsqueda y 2 soluciones ($N = 8, M = 2$) ya que trabajar con 2 qubits y 3 soluciones es simétrico a trabajar con 2 qubits y una única solución, como ya se ha hecho anteriormente. El ejemplo que se trabaja recibe el nombre de escenario 4 y es el mostrado en la figura 19. [5]

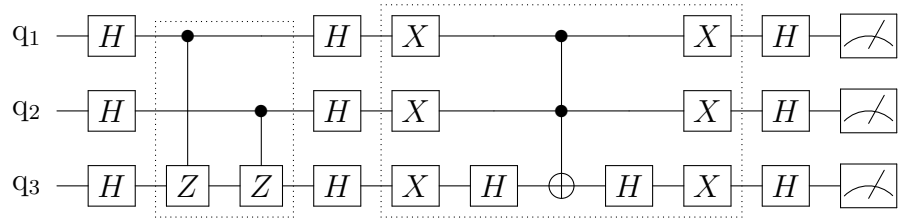


Figura 19: *Circuito implementando $n = 3$ y $M = 2$ con soluciones $|011\rangle$ y $|101\rangle$. Sin hacer uso del qubit extra. Los elementos marcados son el oráculo y el cambio de fase, en ese orden.*

La validez de los oráculos y cambio de fase presentados se muestra en los anexos 3 y 4.

3.3. Errores

Como se ha comentado anteriormente, la gran ventaja de la computación cuántica reside principalmente en el paralelismo cuántico de los qubits. Por otro lado, el estado del arte de esta tecnología emergente está muy lejos de estar optimizado,

de forma que generalmente la implementación de circuitos cuánticos arroja resultados diferentes de los que predice la teoría. Nos referimos a estas diferencias como *errores* que hasta la fecha limitan el potencial de esta tecnología. En esta sección se introducen los errores más comunes asociados a los ordenadores cuánticos. [4]

- Error de medida. Existe una cierta probabilidad de que la medida sea incorrecta y se obtengan resultados diferentes a los reales.

- Error de puerta. El hardware puede implementar una puerta de manera errónea o incluso implementar una diferente, por lo que el resultado sería también distinto. Por ejemplo, se pude dar el caso de que deseemos implementar la puerta P y se implemente $P + \delta P$.

- Decoherencia. Al preparar un estado en superposición del tipo $|0\rangle + e^{i\phi}|1\rangle$ se añade una fase aleatoria e impredecible de tal forma que el estado queda como $|0\rangle + e^{i(\phi+\delta\phi)}|1\rangle$. La dispersión de esta fase aumenta con el tiempo, por lo que acaba quedando totalmente indefinida. Esta pérdida de coherencia puede ser causada por vibraciones, fluctuaciones de temperatura, ondas electromagnéticas y otras interacciones con el medio exterior.

- Relajación o disipación de energía. Los estados $|0\rangle$ y $|1\rangle$ corresponden a autovalores de energías diferentes del Hamiltoniano de un qubit. En concreto, el estado $|0\rangle$ es el estado fundamental y $|1\rangle$ el excitado. Por tanto, si está preparado este último estado hay una cierta probabilidad de que, tras permanecer un tiempo inalterado, se relaje y transite al otro estado.

- *Cross Talking*. Hace referencia a la probabilidad de modificar el estado cuántico de un qubit diferente del que se hace uso.

Se debe tener también en cuenta la configuración el propio ordenador cuántico. Es importante considerar el número de qubits y las puertas que se pueden ejecutar de forma nativa en un backend, ya que cualquier otra puerta debe ser construida a partir de otras y con ello se potencian los errores anteriores. Otro aspecto a tener en cuenta es qué pares de qubits admiten operaciones de puertas de dos qubits entre ellos, que se conoce como topología o conectividad del dispositivo. Debido a las conexiones de los qubits se producen distintos errores. Por último se debe

considerar que el número de ejecuciones realizadas determina la precisión de la distribución de probabilidad de salida. [8]

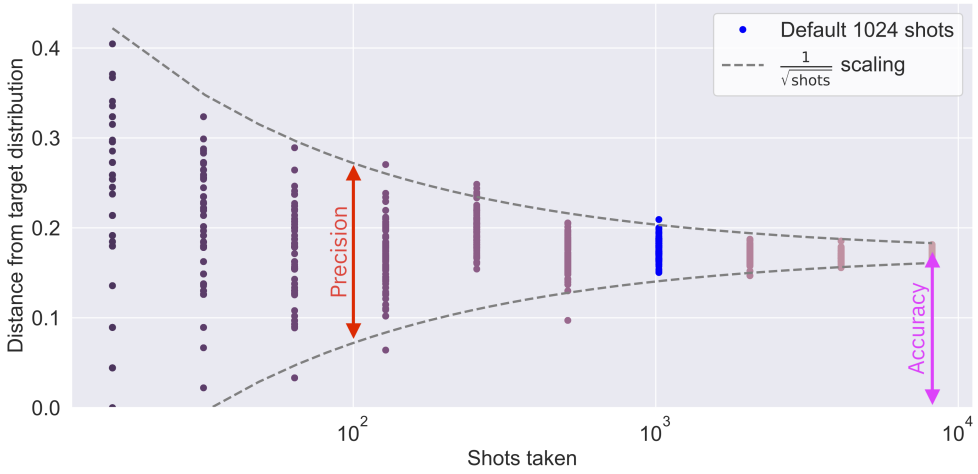


Figura 20: *Distancia respecto del resultado teórico en términos de distancia de Hellinger en función del número de ejecuciones realizadas. Para un estado de Bell ejecutado en el sistema IBM Quantum Boeblingen. Gráfica propiedad de la plataforma IBM Quantum Experience.[8]*

Debido a estos errores los estados cuánticos son propensos a colapsar en un estado no deseado o no esperado teóricamente. El origen reside en que la información cuántica es sensible al ruido de una manera que la información clásica no lo es. En ciertos casos este ruido es aleatorio debido a la propia naturaleza cuántica, por lo que para mejorar los resultados se deben tratar de una manera diferente los errores cuánticos. Aquí entra en juego la corrección cuántica de errores (QEC), teoría desarrollada a partir de las ideas de corrección de errores clásica y cuyo fin es el de tratar los errores de los qubits para hacer que los ordenadores cuánticos sean tolerantes a los fallos [1]. Por lo tanto, el objetivo principal en la actualidad consiste en construir sistemas cuánticos con errores reducidos y tiempos de coherencia largos. El tiempo de coherencia es el tiempo que un qubit conserva sus propiedades cuánticas, por lo que es un parámetro común para evaluar la calidad de un qubit. Es por este ruido que también se suelen proteger los qubits del mundo exterior en cámaras de vacío sobreenfriadas, por ejemplo.

4. Resultados

La implementación en ordenadores cuánticos de los circuitos anteriores se lleva a cabo gracias a la plataforma *IBM Quantum Experience*. Esta plataforma permite el uso de dos interfaces: *Circuit Composer* y *Circuit Notebooks*. La primera de ellas es un interfaz gráfico sencillo con opciones básicas. La segunda opción permite opciones más avanzadas al programar nuestro propio código mediante el marco de programación Qiskit, que está escrito en Python y nos permite interactuar con ordenadores cuánticos. Esta es la opción que se utiliza en este trabajo para obtener los resultados.

En esta plataforma se permite el acceso al público a ciertos backends. Un backend puede referirse a una interfaz para un sistema cuántico o un simulador cuántico clásico. La primera opción hace referencia a los propios ordenadores cuánticos mientras que la segunda consiste en realizar simulaciones de manera clásica donde no se tienen en cuenta los errores asociados a la computación cuántica. [10]

4.1. Simulación

Una vez implementados los circuitos anteriores se utiliza el backend de simulación al cual proporciona acceso la plataforma, que es el llamado *qasm_simulator*. Se trata de un simulador de alto rendimiento para la creación de circuitos de hasta 32 qubits. Admite además la implementación de varios modelos de ruido para simular el rendimiento de los circuitos bajo la acción del ruido del dispositivo. A continuación se realiza la simulación ideal y la simulación considerando un modelo de ruido sencillo con el fin de comparar los resultados. Más concretamente, se genera un modelo simplificado basado en los errores de decoherencia, relajación térmica y medida de los dispositivos reales. [12]. Los resultados obtenidos son los que se presentan a continuación.

Comenzamos analizando el escenario 1. En la tabla 4 se puede ver como la simulación ideal proporciona una probabilidad del 100 % de obtener el resultado deseado. Se trata entonces de un algoritmo determinista debido a que el número de repeticiones es óptimo según la expresión 37. Para la simulación con ruido los resultados son diferentes. Si bien se tiene una gran probabilidad de que el re-

sultado sea correcto existe una pequeña probabilidad de que no lo sea. Esto se debe a los errores comentados anteriormente. Se diferencia además entre los backends *ibmq_ourense* y *ibmq_essex* para observar que existen diferencias entre ellos debido a la configuración de cada ordenador cuántico en cuanto al esquema conexiones de los qubits, además del ruido aleatorio. Según el modelo implementado en este caso se obtendrían mejores resultados para el segundo backend aunque la probabilidad de éxito ronda el 95 % en ambos casos.

	Simulación ideal	Simulación con modelo de ruido básico	
		ibmq_ourense	ibmq_essex
$\|00\rangle$	1	0.947	0.962
$\|01\rangle$	0	0.017	0.020
$\|10\rangle$	0	0.031	0.015
$\|11\rangle$	0	0.005	0.003

Tabla 4: *Probabilidad de medida de cada estado en el escenario 1 para las simulaciones. Se considera $n = 2$, $M = 1$ (solución $|00\rangle$) y 1024 repeticiones.*

	Simulación ideal	Simulación con modelo de ruido básico	
		ibmq_ourense	ibmq_essex
$\|00\rangle$	1	0.859	0.836
$\|01\rangle$	0	0.059	0.061
$\|10\rangle$	0	0.058	0.058
$\|11\rangle$	0	0.024	0.045

Tabla 5: *Probabilidad de medida de cada estado en el escenario 2 para las simulaciones. Se considera $n = 2$, $M = 1$ (solución $|00\rangle$) y 1024 repeticiones.*

La tabla 5 muestra los resultados obtenidos para el escenario 2. Para la simulación ideal se obtiene el mismo resultado que en el caso anterior por el mismo motivo. Nótese que el número óptimo de llamadas al operador de Grover no depende de la presencia de qubits en el espacio de trabajo del oráculo. Las probabilidades de éxito en este caso son menores al tener un circuito más complejo. En ambos casos toman valores del 85 % aproximadamente, por lo que sí se observa diferencia

con respecto del escenario 1. Para este modelo se obtendrían mejores resultados con el primer backend.

En la tabla 6 se presentan los resultados de la simulación ideal para el escenario 3. Tal como cabía esperar, en este caso se obtienen errores considerables. Para la simulación ideal se tiene aproximadamente un 25 % de probabilidad de medir cada uno de los resultado deseados. La probabilidad de éxito es del 50 % aproximadamente, por lo que alrededor de un 50 % de las veces el algoritmo fallaría en el caso ideal. Esto se debe a que el número óptimo de llamadas al operador de Grover no es entero en ningún caso, por lo que no se puede implementar. Como consecuencia la probabilidad de éxito disminuye debido al comportamiento oscilatorio. En el caso de querer resolver precisamente este problema se pueden añadir elementos al espacio de búsqueda que no constituyan una solución, permaneciendo así el número de soluciones inalterado. Por ejemplo, si se añaden 4 elementos se tiene $N = 8$ y $M = 2$. Para estos valores sí se obtiene un número de llamas entero, por lo que el problema podría resolver de forma determinista. Se opta por descartar este caso y trabajar con el escenario 4 para mostrar un ejemplo con múltiples soluciones.

	Simulación ideal
$\|00\rangle$	0.250
$\|01\rangle$	0.236
$\|10\rangle$	0.250
$\|11\rangle$	0.264

Tabla 6: *Probabilidad de medida de cada estado en el escenario 3 para las simulaciones. Se considera $n = 2$, $M = 2$ (soluciones $|01\rangle$ y $|10\rangle$) y 1024 repeticiones.*

Por último se muestran los resultados para el escenario 4 en la tabla 7. Para la simulación ideal se obtienen los resultados esperados al tener un 50 % de probabilidad de obtener cada elemento solución. En las simulaciones con ruido se obtienen probabilidades de éxito menores. En este caso para cualquiera de los dos backends presentados los resultados serían muy similares. Se ronda el 80 % de probabilidad de éxito.

	Simulación ideal	Simulación con modelo de ruido básico	
		ibmq_ourense	ibmq_essex
$\|000\rangle$	0	0.021	0.014
$\|001\rangle$	0	0.039	0.065
$\|010\rangle$	0	0.036	0.027
$\|011\rangle$	0.501	0.411	0.397
$\|100\rangle$	0	0.022	0.040
$\|101\rangle$	0.499	0.412	0.398
$\|110\rangle$	0	0.014	0.017
$\|111\rangle$	0	0.045	0.041

Tabla 7: *Probabilidad de medida de cada estado en el escenario 4 para las simulaciones. Se considera $n = 3$, $M = 2$ (soluciones $|011\rangle$ y $|101\rangle$) y 1024 repeticiones.*

4.2. Implementación real

En este apartado se ejecutan los circuitos en los propios ordenadores cuánticos. En primer lugar se realiza la implementación real y posteriormente se mitiga el ruido existente. Para ello se utiliza la librería *ignis*, que permite reducir el error creando un filtro a partir de los erres de los propios ordenadores cuánticos. Se ejecutan circuitos formados por cada puerta del circuito original y se mide el error obtenido [11]. Los backends utilizados son a los que da acceso la plataforma. Todos ellos se componen de 5 qubits excepto el llamado *ibmq_16_melbourne*, que tiene 15 qubits.

Los resultados obtenidos para el escenario 1 se presentan en la tabla 8. En el bloque de implementación real se observa que existen errores considerables en algunos casos, aunque por lo general los resultados son relativamente buenos al ser la probabilidad de éxito superior al 90 %. Al mitigar los errores se obtienen mayores probabilidades de éxito pese a que el resultado no es el esperado teóricamente. Cabe destacar que la acción de mitigar los errores con el modelo utilizado es limitada debido a su simplicidad y que en algunos casos es más efectivo que en otros.

	Backend	$\|00\rangle$	$\|01\rangle$	$\|10\rangle$	$\|11\rangle$
Valor teórico	cualquiera	1	0	0	0
Implementación real	ibmq_16_melbourne	0.918	0.035	0.023	0.024
	ibmq_london	0.931	0.035	0.023	0.011
	ibmq_burlington	0.649	0.182	0.107	0.062
	ibmq_essex	0.807	0.060	0.102	0.032
	ibmq_ourense	0.940	0.023	0.027	0.009
	ibmq_5_yorktown	0.958	0.018	0.017	0.008
	ibmq_vigo	0.960	0.005	0.030	0.005
	ibmq_rome	0.961	0.016	0.017	0.007
	ibmq_athens	0.962	0.019	0.011	0.009
	ibmqx2	0.962	0.014	0.013	0.012
Implementación real mitigando error	ibmq_16_melbourne	0.925	0.028	0.021	0.027
	ibmq_london	0.939	0.035	0.015	0.012
	ibmq_burlington	0.700	0.215	0.037	0.049
	ibmq_essex	0.829	0.043	0.079	0.039
	ibmq_ourense	0.969	0.011	0.010	0.010
	ibmq_5_yorktown	0.966	0.011	0.020	0.004
	ibmq_vigo	0.980	0.000	0.020	0.000
	ibmq_rome	0.976	0.008	0.008	0.007
	ibmq_athens	0.970	0.015	0.005	0.010
	ibmqx2	0.975	0.007	0.009	0.010

Tabla 8: *Probabilidad de medida de cada estado en el escenario 1 en los dispositivos reales en función de cada uno de ellos. Se considera $n = 2$, $M = 1$ (solución $|00\rangle$) y 1024 repeticiones.*

Al probar todos los backends se está haciendo un estudio sobre la variabilidad de los resultados. En la figura 21a se muestra la probabilidad de éxito en función del backend utilizado de forma más resumida, además de incluir la media. La variabilidad de los dispositivos se aprecia claramente. Destaca, como se ha dicho anteriormente, que para algunos de los backend los resultados no son tan buenos como en otros. Los backends también muestran resultados variables en el tiempo debido a los errores comentados. Esto significa que para diferentes ejecuciones se

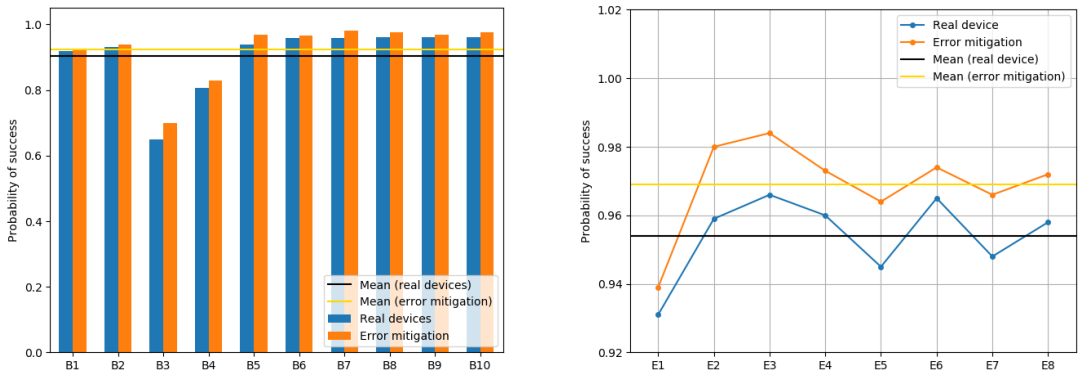
obtienen resultados ligeramente diferentes si transcurre un determinado tiempo. Como ejemplo se muestra la figura 21b.

	Backend	$\|00\rangle$	$\|01\rangle$	$\|10\rangle$	$\|11\rangle$
Valor teórico	cualquiera	1	0	0	0
Implementación real	ibmq_16_melbourne	0.599	0.154	0.204	0.043
	ibmq_london	0.622	0.087	0.146	0.145
	ibmq_burlington	0.419	0.192	0.169	0.220
	ibmq_essex	0.670	0.130	0.159	0.041
	ibmq_ourense	0.824	0.078	0.066	0.032
	ibmq_5_yorktown	0.734	0.039	0.172	0.055
	ibmq_vigo	0.861	0.062	0.053	0.024
	ibmq_rome	0.647	0.078	0.128	0.147
	ibmq_athens	0.790	0.086	0.060	0.064
	ibmqx2	0.755	0.041	0.147	0.057
Implementación real mitigando error	ibmq_16_melbourne	0.533	0.155	0.214	0.097
	ibmq_london	0.618	0.082	0.140	0.160
	ibmq_burlington	0.457	0.182	0.195	0.166
	ibmq_essex	0.694	0.111	0.158	0.037
	ibmq_ourense	0.846	0.072	0.051	0.031
	ibmq_5_yorktown	0.856	0.031	0.072	0.041
	ibmq_vigo	0.870	0.056	0.048	0.026
	ibmq_rome	0.655	0.058	0.0.128	0.159
	ibmq_athens	0.789	0.085	0.056	0.071
	ibmqx2	0.907	0.032	0.011	0.050

Tabla 9: *Probabilidad de medida de cada estado en el escenario 2 en los dispositivos reales en función de cada uno de ellos. Se considera $n = 2$, $M = 1$ (solución $|00\rangle$) y 1024 repeticiones. Haciendo uso de un qubit en el espacio de trabajo del oráculo.*

En la tabla 9 se presentan los resultados experimentales para el escenario 2. Recordemos que es el mismo que el escenario 1 pero haciendo uso de un qubit en el espacio de trabajo del oráculo. En el bloque de implementación real se observan mayores errores que en el caso anterior, al igual que en el de mitigación de ruido.

Esto significa que las probabilidades de éxito son menores. Se muestra también la limitación del modelo de corrección de errores empleado. En la figura 22 se muestran los datos de forma resumida y se confirma la disminución de probabilidad de éxito del algoritmo. También se aprecia la variabilidad de los backends, que en este caso es más evidente. Los errores son más notables en el escenario 2 por el aumento del número de puertas utilizadas, que contribuye a un mayor ruido. Se puede concluir que el comportamiento general de los datos es el mismo pese a que la probabilidad de éxito en todos los casos es menor que en el caso anterior.



(a) (b)

Figura 21: *Probabilidad de éxito en función del backend utilizado (a) y probabilidad de éxito para el backend ibmq_london en diferentes ejecuciones (b) para el escenario 1.*

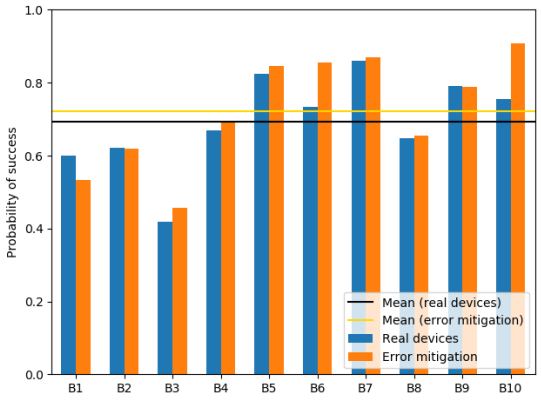


Figura 22: *Probabilidad de éxito en función del backend utilizado para el escenario 2.*

Por último se presentan los datos experimentales para el escenario 4 en la tabla

10. De nuevo, se presentan las probabilidades para cada estado de la base computacional. Al tener dos soluciones existen dos estados para los cuales se obtienen considerablemente mayor probabilidad de medida. Esta probabilidad de éxito, sin embargo, es mucho menor que en los casos anteriores, llegando incluso a un 16 % para un elemento solución en la implementación real. Mitigando los errores los resultados mejoran pero es cierto que los errores en algunos de los resultados proporcionados por el algoritmo cuántico no son tolerables, lo cual supone un problema. Como en los casos anteriores, en la figura 23 se muestran estos datos de manera resumida junto a las medias correspondientes.

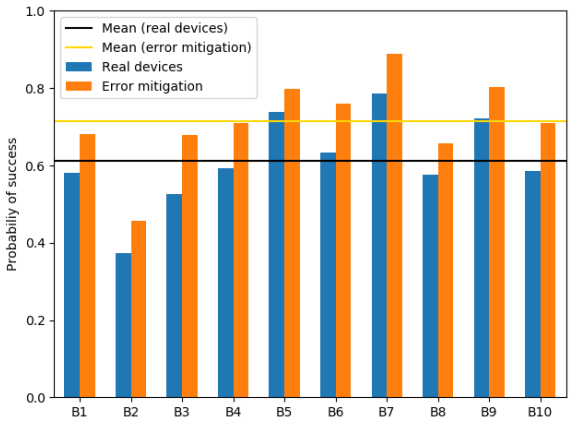


Figura 23: *Probabilidad de éxito en función del backend para el escenario 3.*

	Backend	$\|000\rangle$	$\|001\rangle$	$\|010\rangle$	$\|011\rangle$	$\|100\rangle$	$\|101\rangle$	$\|110\rangle$	$\|111\rangle$
Valor teórico	cualquiera	0	0	0	0.501	0	0.499	0	0
Implementación real	ibmq_16_melbourne	0.069	0.125	0.060	0.289	0.049	0.292	0.035	0.081
	ibmq_london	0.077	0.099	0.116	0.161	0.144	0.212	0.091	0.101
	ibmq_burlington	0.060	0.124	0.068	0.286	0.090	0.239	0.050	0.083
	ibmq_essex	0.033	0.112	0.062	0.272	0.056	0.320	0.042	0.102
	ibmq_ourense	0.023	0.057	0.051	0.395	0.035	0.339	0.028	0.072
	ibmq_5_yorktown	0.027	0.062	0.066	0.257	0.058	0.376	0.083	0.070
	ibmq_vigo	0.023	0.062	0.040	0.374	0.029	0.413	0.033	0.025
	ibmq_rome	0.041	0.086	0.056	0.253	0.087	0.323	0.102	0.053
	ibmq_athens	0.027	0.083	0.038	0.391	0.048	0.331	0.053	0.029
	ibmqx2	0.030	0.072	0.054	0.237	0.066	0.358	0.097	0.085
Implemetación real mitigando errores	ibmq_16_melbourne	0.059	0.081	0.040	0.346	0.030	0.335	0.034	0.075
	ibmq_london	0.042	0.070	0.092	0.204	0.108	0.252	0.099	0.134
	ibmq_burlington	0.030	0.033	0.071	0.393	0.076	0.286	0.040	0.070
	inmq_essex	0.019	0.096	0.013	0.337	0.023	0.373	0.028	0.112
	ibmq_ourense	0.022	0.039	0.032	0.419	0.021	0.378	0.026	0.064
	ibmq_5_yorktown	0.022	0.059	0.064	0.363	0.027	0.397	0.032	0.037
	ibmq_vigo	0.019	0.035	0.000	0.441	0.013	0.449	0.029	0.014
	ibmq_rome	0.031	0.064	0.034	0.282	0.068	0.375	0.106	0.046
	ibmq_athens	0.019	0.052	0.012	0.436	0.028	0.366	0.055	0.031
	ibmqx2	0.024	0.076	0.047	0.314	0.041	0.395	0.050	0.052

Tabla 10: *Probabilidad de medida de cada estado en el escenario 4 en los dispositivos reales en función de cada uno de ellos. Se considera $n = 3$, $M = 2$ (soluciones $|011\rangle$ y $|101\rangle$) y 1024 repeticiones. Ausencia de qubits en el espacio de trabajo del oráculo.*

Por último se comparan las medias aritméticas y las dispersiones de cada escenario en la tabla 11. Para el estudio de la dispersión se ha calculado la desviación estándar de la media. Se observa claramente que para el primer escenario se tiene una mayor probabilidad media de éxito debido a ser el circuito más simple. Al aumentar el número de puertas aumentan los errores y con ello disminuye dicha media. Se confirma también que la mitigación de errores es efectiva, aunque en algunos casos lo es más que en otros debido al modelo simplificado basado en errores aleatorios. Finalmente la mayor probabilidad de éxito obtenida es el 91 % aproximadamente para el escenario 2 frente al 71 % del escenario 3. Como resultado se puede extraer que el algoritmo cuántico de búsqueda no es conveniente en algunas situaciones pero sí es efectivo en otras. Por otra parte, la dispersión es menor en el primer escenario, por lo que la precisión de los datos es mayor. Para los escenarios 2 y 4 la desviación aumenta, siendo mayor para el segundo. Destaca que en el caso de mitigación de errores la dispersión de los datos se reduce excepto para el segundo caso.

Escenario		1	2	4
Media probabilidad de éxito	Implementación real	0.897	0.692	0.611
	Mitigación de ruido	0.911	0.722	0.714
Desviación estándar probabilidad de éxito	Implementación real	0.096	0.123	0.113
	Mitigación de ruido	0.086	0.147	0.109

Tabla 11: *Medias y las dispersiones de la probabilidad de éxito para las implementaciones de los escenarios 1, 2 y 4.*

4.3. Posibles aplicaciones

Pese a que el algoritmo de Grover no proporciona una mejora tan significativa como en otros algoritmo cuánticos como el de Deutsch-Jozsa (mejora exponencial) debemos tener en cuenta que es aplicable a un rango mucho mayor de problemas. Esto se debe a que la búsqueda es uno de los problemas más importantes y se suele utilizar como una subrutina de muchos otros algoritmos [2]. El algoritmo de Grover es realmente útil si el problema no se puede resolver clásicamente por el tamaño de la base de datos y si el tiempo de ejecución es excesivamente grande.

Entonces el algoritmo estudiado podría tener la capacidad de computar dentro de un período de tiempo razonable.

- Mejora de algoritmos aleatorios. En un algoritmo aleatorio clásico se utiliza una semilla (secuencia de números pseudoaleatorios) para determinar una trayectoria a través del espacio de búsqueda ya que para distintas ejecuciones se obtienen distintas rutas. Clásicamente se ejecuta el algoritmo de búsqueda y tras un cierto número de pasos se evalúa si resultado es solución. La búsqueda cuántica puede acelerar este proceso haciendo uso de una superposición de semillas con el fin de obtener una superposición de estados finales. De esta manera se podría extraer la solución deseada con una mejora cuadrática en el número de ensayos. [2]

- Resolución del problema de satisfacción booleana. Este problema consiste en determinar si existe una interpretación que satisfaga una función booleana dada. Si se da el caso se conoce como función satisfactoria mientras que si no existe tal asignación la función es falsa para todas las variables de entrada posibles y la fórmula no es satisfactoria. La situación planteada se puede tomar como un problema de búsqueda donde la solución es la asignación que satisface la función booleana. [13]

- Preparación de estados de superposición seleccionados. Se puede obtener una superposición de índices que correspondan únicamente a un cierto tipo de números, como por ejemplo los primos. Simplemente se debe implementar un oráculo que considere como soluciones estos números. Aplicando el algoritmo completo se obtendría una superposición de índices asociados a los números primos dentro del rango considerado. Esto sería de gran importancia en física cuántica experimental al poder crear los estados de superposición deseados. [2]

5. Conclusiones

En este trabajo he abordado el análisis del algoritmo cuántico de Grover y su implementación en los ordenadores cuánticos de IBM. Este algoritmo resuelve el problema de encontrar una entrada en una base de datos desestructurada. Se ha demostrado que presenta una mejora cuadrática respecto del caso clásico en cuanto a llamadas al oráculo. Considerando además que es óptimo presenta grandes ven-

tajas teóricamente. Cabe destacar que para cada escenario se necesita un oráculo y un cambio de fase diferentes, que dificulta la implementación del propio algoritmo cuando se tiene un gran número de qubits en el espacio de búsqueda.

A la hora de realizar la implementación real aparecen diversas discrepancias con respecto a la simulación ideal. Los errores no siempre son relativamente bajos y varían de un backend a otro, además de que dependen de la presencia o ausencia de qubits en el espacio de trabajo del oráculo. De ello se puede concluir que es necesario aplicar modelos de corrección de errores para que estos sean tolerables. Un modelo satisfactorio es el anteriormente comentado, que evalúa el ruido para cada puerta del circuito y aplica un filtro al resultado de la implementación real. En los casos de 2 y 3 qubits en el espacio de trabajo principal no hay grandes problemas pero para que esta tecnología sea realmente útil se debe poder aplicar a un mayor número de qubits, que implica un mayor número de puertas y un mayor error. Pese a tener modelos de corrección de errores más complejos y efectivos, el origen de algunos de ellos reside en la naturaleza cuántica de los qubits. Por estos motivos se necesitan avances antes de que esta tecnología tenga aplicaciones reales pese a que teóricamente presenta un gran abanico de posibilidades. Las computadoras cuánticas capaces de realizar operaciones con pocos qubits representan el estado del arte en computación cuántica.

6. Anexos

6.1. Demostración equivalencia de puertas de la figura 4b

Comenzamos con un estado inicialmente preparado como superposición de las componentes de la base computacional, que se expresa como $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \eta|11\rangle$. Aplicando la puerta de la parte derecha queda:

$$|\psi_1^1\rangle = \alpha|01\rangle + \beta|00\rangle + \gamma|10\rangle + \eta|11\rangle \tag{55}$$

Se aplica ahora la sucesión de puertas de la parte izquierda.

$$|\psi_2^1\rangle = \alpha|10\rangle + \beta|11\rangle + \gamma|00\rangle + \eta|01\rangle$$

$$|\psi_2^2\rangle = \alpha|11\rangle + \beta|10\rangle + \gamma|00\rangle + \eta|01\rangle$$

$$|\psi_2^3\rangle = \alpha|01\rangle + \beta|00\rangle + \gamma|10\rangle + \eta|11\rangle \tag{56}$$

Los estados finales $|\psi_1^1\rangle$ y $|\psi_2^3\rangle$ (dados por las expresiones 55 y 56) son iguales, por lo que las puertas son equivalentes.

6.2. Demostración expresión 36

Es conveniente trabajar en forma matricial, por lo que en primer lugar se halla la matriz correspondiente al operador de Grover.

1. Operador oráculo. Tal como se ha definido anteriormente, la acción del oráculo es:

$$O|x_l\rangle = |x_l\rangle \qquad O|x_w\rangle = -|x_w\rangle$$

Se calculan los elementos de matriz proyectando sobre los estados:

$$\langle x_w|O|x_w\rangle = -\langle x_w|x_w\rangle = -1$$

$$\langle x_w|O|x_l\rangle = \langle x_w|x_l\rangle = 0$$

$$\langle x_l|O|x_w\rangle = -\langle x_l|x_w\rangle = 0$$

$$\langle x_l|O|x_l\rangle = \langle x_l|x_l\rangle = 1$$

Por tanto, la matriz resultante es:

$$O = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \tag{57}$$

2. Operador inversión. La expresión general es $J = 2|\psi_1\rangle\langle\psi_1| - I$. Los elementos de la matriz son los siguientes:

$$2\langle x_w|\psi_1\rangle\langle\psi_1|x_w\rangle - \langle x_w|x_l\rangle = 2sin^2\theta - 1 \tag{58}$$

$$2\langle x_w|\psi_1\rangle\langle\psi_1|x_l\rangle - \langle x_w|x_l\rangle = 2sin\theta cos\theta \tag{59}$$

$$2\langle x_l|\psi_1\rangle\langle\psi_1|x_w\rangle - \langle x_l|x_w\rangle = 2cos\theta sin\theta \tag{60}$$

$$2\langle x_l|\psi_1\rangle\langle\psi_1|x_l\rangle - \langle x_l|x_l\rangle = 2cos^2\theta - 1 \tag{61}$$

Por tanto, la matriz resultante es:

$$M = \begin{bmatrix} 2sin^2\theta - 1 & 2sin\theta cos\theta \\ 2sin\theta cos\theta & 2cos^2\theta - 1 \end{bmatrix} \tag{62}$$

Como resultado, el operador de Grover se puede expresar de la siguiente manera:

$$\begin{aligned} G &= \begin{bmatrix} 2sin^2\theta - 1 & 2sin\theta cos\theta \\ 2sin\theta cos\theta & 2cos^2\theta - 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 - 2sin^2\theta & 2sin\theta cos\theta \\ -2sin\theta cos\theta & 2cos^2\theta - 1 \end{bmatrix} \\ &= \begin{bmatrix} cos(2\theta) & sin(2\theta) \\ -sin(2\theta) & cos(2\theta) \end{bmatrix} \end{aligned} \tag{63}$$

Teniendo en cuenta que la matriz de rotación general en sentido horario un ángulo w viene dada por la expresión:

$$R = \begin{bmatrix} cos(\omega) & -sin(\omega) \\ sin(\omega) & cos(\omega) \end{bmatrix} \tag{64}$$

se deduce que la matriz dada por la expresión 63 rota el estado sobre el que actúa un ángulo 2θ en sentido antihorario.

A continuación se calcula G^k, es decir, la matriz de rotación tras haberla aplicado k veces. Para ello se diagonaliza G ya que de esta manera es mucho más sencillo trabajar con potencias de matrices. Una matriz cuadrada es diagonalizable si es semejante a una matriz diagonal, es decir, si puede escribirse como diagonal mediante un cambio de base. Se hace uso del teorema de descomposición espectral y G se descompone de la siguiente forma:

$$G = PDP^{-1} \tag{65}$$

donde D es la forma diagonal asociada a G formada por los valores propios y P es una matriz invertible cuyos vectores columna son vectores propios de G. Al ser P una matriz invertible existe P^{-1} tal que $PP^{-1} = P^{-1}P = I$. A partir de la expresión 65 se calcula G^k como:

$$G^k = (PDP^{-1})^k = PD^kP^{-1} \tag{66}$$

Nótese que en esta expresión la matriz que aparece elevada a una potencia es la diagonal, por la que tan solo se deben elevar los elementos para calcularla. Esta

es la gran ventaja de trabajar con matrices. Comenzamos obteniendo los valores propios de la siguiente manera:

$$|G - \lambda I| = 0 \tag{67}$$

Por tanto:

$$|G - \lambda I| = \begin{vmatrix} cos(2\theta) - \lambda & sin(2\theta) \\ -sin(2\theta) & cos(2\theta) - \lambda \end{vmatrix} = \Big(cos(2\theta) - \lambda\Big)^2 + sin^2(2\theta) = 0 \tag{68}$$

Los autovalores son:

$$\lambda_1 = cos(2\theta) - isin(2\theta) = e^{-i2\theta} \tag{69}$$

$$\lambda_2 = cos(2\theta) + isin(2\theta) = e^{i2\theta} \tag{70}$$

Como consecuencia, la matriz D es:

$$D = \begin{bmatrix} e^{-i2k\theta} & 0 \\ 0 & e^{i2k\theta} \end{bmatrix} \tag{71}$$

Por otro lado, para hallar la matriz P es necesario obtener los autovectores asociados a los autovalores. Para el primer autovalor:

$$(G - \lambda_1 I)X_1 = \begin{bmatrix} isin(2\theta) & sin(2\theta) \\ -sin(2\theta) & isin(2\theta) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} (ia + b)sin(2\theta) \\ (-a + ib)sin(2\theta) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

La condición que se obtiene es $a = ib$, por lo que queda:

$$X_1 = \begin{bmatrix} i \\ 1 \end{bmatrix} \tag{72}$$

Para el segundo autovalor:

$$(G - \lambda_2 I)X_2 = \begin{bmatrix} -isin(2\theta) & sin(2\theta) \\ -sin(2\theta) & -isin(2\theta) \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} (-ic + d)sin(2\theta) \\ (-c - id)sin(2\theta) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

La condición que se obtiene es $c = -id$, por lo que queda:

$$X_2 = \begin{bmatrix} -i \\ 1 \end{bmatrix} \tag{73}$$

Con las expresiones 72 y 73 queda:

$$P = \begin{bmatrix} i & -i \\ 1 & 1 \end{bmatrix} \qquad P^{-1} = \frac{1}{2}\begin{bmatrix} -i & 1 \\ i & 1 \end{bmatrix} \tag{74}$$

Como consecuencia de lo anterior y a partir de la expresión 66:

$$\begin{aligned} \mathrm{G^k} = PD^kP^{-1} &= \begin{bmatrix} i & -i \\ 1 & 1 \end{bmatrix}\begin{bmatrix} e^{-i2k\theta} & 0 \\ 0 & e^{i2k\theta} \end{bmatrix}\frac{1}{2}\begin{bmatrix} -i & 1 \\ i & 1 \end{bmatrix} \\ &= \begin{bmatrix} cos(2k\theta) & sin(2k\theta) \\ -sin(2k\theta) & cos(2k\theta) \end{bmatrix} \end{aligned} \tag{75}$$

Aplicando G^k sobre $|\psi_1\rangle = sin\theta|x_w\rangle + cos\theta|x_l\rangle$ queda:

$$G^k|\psi_1\rangle = \begin{bmatrix} cos(2k\theta) & sin(2k\theta) \\ -sin(2k\theta) & cos(2k\theta) \end{bmatrix}\begin{bmatrix} sin\theta \\ cos\theta \end{bmatrix} = \begin{bmatrix} cos(2k\theta)sin\theta + sin(2k\theta)cos\theta \\ -sin(2k\theta)sin\theta + cos(2k\theta)cos\theta \end{bmatrix} \tag{76}$$

A continuación se desarrolla el primer término de la expresión matricial anterior:

$$\begin{gathered} cos(2k\theta)sin\theta + sin(2k\theta)cos\theta = \\ \left(\frac{e^{i2k\theta} + e^{-i2k\theta}}{2}\right)\left(\frac{e^{i\theta} - e^{-i\theta}}{2i}\right) + \left(\frac{e^{i2k\theta} - e^{-i2k\theta}}{2i}\right)\left(\frac{e^{i\theta} + e^{-i\theta}}{2}\right) = \\ \frac{1}{2i}\left(e^{i(2k+1)\theta} - e^{-i(2k+1)\theta}\right) = sin\left((2k+1)\theta\right) \end{gathered} \tag{77}$$

Análogamente, para el segundo término:

$$\begin{gathered} -sin(2k\theta)sin\theta - cos(2k\theta)cos\theta = \\ -\left(\frac{e^{i2k\theta} - e^{-i2k\theta}}{2i}\right)\left(\frac{e^{i\theta} - e^{-i\theta}}{2i}\right) - \left(\frac{e^{i2k\theta} + e^{-i2k\theta}}{2}\right)\left(\frac{e^{i\theta} + e^{-i\theta}}{2}\right) = \\ \frac{1}{2}\left(e^{i(2k+1)\theta} + e^{-i(2k+1)\theta}\right) = cos\left((2k+1)\theta\right) \end{gathered} \tag{78}$$

Finalmente queda como resultado:

$$G^k|\psi_1\rangle = \begin{bmatrix} sin((2k+1)\theta) \\ cos((2k+1)\theta) \end{bmatrix} \tag{79}$$

Se puede expresar también como:

$$G^k|\psi_1\rangle = sin\Big((2k+1)\theta\Big)|x_w\rangle + cos\Big((2k+1)\theta\Big)|x_l\rangle \tag{80}$$

6.3. Demostración de la validez de los operadores cambio de fase

En la figura 16 se muestran los cambios de fase para $n = 2$, que actúan sobre 2 qubits. Inicialmente el estado general está preparado como superposición normalizada de las componentes de la base computacional y se expresa como $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \eta|11\rangle$. Para el primer cambio de fase (figura 16a) se obtiene:

$$\begin{aligned} |\psi_3^1\rangle &= \alpha|00\rangle - \beta|01\rangle - \gamma|10\rangle + \eta|11\rangle \\ |\psi_3^2\rangle &= \alpha|00\rangle - \beta|01\rangle - \gamma|10\rangle - \eta|11\rangle \end{aligned} \tag{81}$$

Esta función de onda coincide con la teórica al aplicar el cambio de fase al haber cambiado el signo de todos los estados de la base computacional excepto para $|00\rangle$.

Para el segundo operador (figura 16b) se parte de nuevo del estado general:

$$\begin{aligned} |\psi_4^1\rangle &= \alpha|11\rangle + \beta|10\rangle + \gamma|01\rangle + \eta|00\rangle = (\eta|0\rangle + \beta|1\rangle)|0\rangle + (\gamma|0\rangle + \alpha|1\rangle)|1\rangle \\ |\psi_4^2\rangle &= (\eta|0\rangle + \beta|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + (\gamma|0\rangle + \alpha|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ |\psi_4^3\rangle &= (\eta|0\rangle + \beta|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + (\gamma|0\rangle - \alpha|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ |\psi_4^4\rangle &= (\eta|0\rangle + \beta|1\rangle)|0\rangle + (\gamma|0\rangle - \alpha|1\rangle)|1\rangle \\ |\psi_4^5\rangle &= -\alpha|00\rangle + \gamma|10\rangle + \beta|01\rangle + \eta|11\rangle \end{aligned} \tag{82}$$

La expresión 82 difiere en una fase $e^{i\pi} = -1$ respecto de 81, que es la función obtenida con el otro cambio de fase. Los estados cuánticos que difieren en una fase son físicamente indistinguibles. Esto se debe a que la densidad de probabilidad de los dos estados coincide. Para el primero tenemos $|\psi_3^2|^2$ y para el segundo $|\psi_4^5|^2 = |e^{i\pi}\psi_3^2|^2 = |\psi_3^2|^2$. Por tanto, el segundo cambio de fase utilizado también es válido.

El cambio de fase presentado para $n = 3$ se muestra en la figura 19. Comenzamos con un estado inicialmente preparado como superposición de los componentes de la

base computacional, que se expresa como $|\psi\rangle = a|000\rangle + b|001\rangle + c|010\rangle + d|011\rangle + e|100\rangle + f|101\rangle + g|110\rangle + h|111\rangle$. Se considera normalizado. Al aplicar cada puerta se obtiene:

$$\begin{aligned}|\psi_5^1\rangle &= a|111\rangle + b|110\rangle + c|101\rangle + d|100\rangle + e|011\rangle + f|010\rangle + g|001\rangle + h|000\rangle \\ &= (h|00\rangle + f|01\rangle + d|10\rangle + b|11\rangle)|0\rangle + (g|00\rangle + e|01\rangle + c|10\rangle + a|11\rangle)|0\rangle\end{aligned} \tag{83}$$

$$\begin{aligned}|\psi_5^2\rangle &= (h|00\rangle + f|01\rangle + d|10\rangle + b|11\rangle)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ &+ (g|00\rangle + e|01\rangle + c|10\rangle + a|11\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\end{aligned}$$

$$\begin{aligned}|\psi_5^3\rangle &= (h|00\rangle + f|01\rangle + d|10\rangle + b|11\rangle)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ &+ (g|00\rangle + e|01\rangle + c|10\rangle - a|11\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\end{aligned}$$

$$|\psi_5^4\rangle = (h|00\rangle + f|01\rangle + d|10\rangle + b|11\rangle)|0\rangle + (g|00\rangle + e|01\rangle + c|10\rangle - a|11\rangle)|1\rangle$$

$$|\psi_5^5\rangle = -a|000\rangle + b|001\rangle + c|010\rangle + d|011\rangle + e|100\rangle + f|101\rangle + g|110\rangle + h|111\rangle \tag{84}$$

Tan solo cambia el signo del estado fundamental, por lo que se trata de un cambio de fase válido.

6.4. Demostración de la validez de los oráculos de fase

En primer lugar se trabaja con $M = 1$ y $n = 2$, por lo operadores que actúan únicamente sobre 2 qubits. Según la teoría, el efecto debe ser cambiar la fase de la solución al problema tal que $O|x\rangle = (-1)^{f(x)}|x\rangle$. Consideramos el estado inicial como una superposición de las componentes de la base computacional: $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \eta|11\rangle$. Asumimos que está función de onda se encuentra normalizada. Para la solución $|00\rangle$:

$$|\psi_6^1\rangle = \alpha|11\rangle + \gamma|10\rangle + \beta|01\rangle + \eta|00\rangle$$

$$|\psi_6^2\rangle = -\alpha|11\rangle + \gamma|10\rangle + \beta|01\rangle + \eta|00\rangle$$

$$|\psi_6^3\rangle = -\alpha|00\rangle + \gamma|01\rangle + \beta|10\rangle + \eta|11\rangle \tag{85}$$

Si la solución al problema es $|01\rangle$ los estados obtenidos al aplicar las puertas son:

$$\begin{aligned} |\psi_7^1\rangle &= \alpha|10\rangle + \gamma|11\rangle + \beta|00\rangle + \eta|01\rangle \\ |\psi_7^2\rangle &= \alpha|10\rangle - \gamma|11\rangle + \beta|00\rangle + \eta|01\rangle \\ |\psi_7^3\rangle &= \alpha|00\rangle - \gamma|01\rangle + \beta|10\rangle + \eta|11\rangle \end{aligned} \tag{86}$$

Para el estado $|10\rangle$:

$$\begin{aligned} |\psi_8^1\rangle &= \alpha|01\rangle + \beta|00\rangle + \gamma|11\rangle + \eta|10\rangle \\ |\psi_8^2\rangle &= \alpha|01\rangle + \beta|00\rangle - \gamma|11\rangle + \eta|10\rangle \\ |\psi_8^3\rangle &= \alpha|00\rangle + \beta|01\rangle - \gamma|10\rangle + \eta|11\rangle \end{aligned} \tag{87}$$

En el caso en que la solución sea $|11\rangle$ el oráculo es simplemente una puerta CZ:

$$|\psi_9^1\rangle = \alpha|00\rangle + \gamma|10\rangle + \beta|01\rangle - \eta|11\rangle \tag{88}$$

En las expresiones 85, 86, 87 y 88 son concordantes con lo esperado según la teoría, que es cambiar el signo de la solución al problema.

Pasamos a continuación al caso $n = 2$ y $M = 2$, cuyo oráculo se muestra en la figura 18. Las soluciones al problema de búsqueda son $|01\rangle$ y $|10\rangle$. Partiendo del estado general:

$$|\psi_{10}^1\rangle = \alpha|00\rangle - \gamma|10\rangle - \beta|01\rangle + \eta|11\rangle \tag{89}$$

El oráculo es entonces válido.

Por último, se demuestra la validez del oráculo presentado para el caso $n = 3$ y $M = 2$, que se presenta en la figura 19. Como en el apartado anterior, se considera como estado inicial $|\psi\rangle = a|000\rangle + b|001\rangle + c|010\rangle + d|011\rangle + e|100\rangle + f|101\rangle + g|110\rangle + h|111\rangle$. Entonces:

$$\begin{aligned} |\psi_{11}^1\rangle &= a|000\rangle + b|001\rangle + c|010\rangle + d|011\rangle + e|100\rangle - f|101\rangle + g|110\rangle - h|111\rangle \\ |\psi_{11}^2\rangle &= a|000\rangle + b|001\rangle + c|010\rangle - d|011\rangle + e|100\rangle - f|101\rangle + g|110\rangle + h|111\rangle \end{aligned} \tag{90}$$

Queda demostrado entonces que sí se trata de un oráculo válido para las soluciones $|011\rangle$ y $|101\rangle$.

6.5. Demostración de la validez de los oráculos booleanos

En esta sección se demuestra la validez de los operadores O' de la tabla 3, por lo que queda demostrado también que el oráculo completo O es válido. Se trata del caso $n = 2$ y $M = 1$ con un qubit en el espacio de trabajo del oráculo. Para O' se había inicializado $|q\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, por lo que el estado inicial es $|\psi\rangle = \Big(\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \eta|11\rangle\Big) \otimes |q\rangle$. Los resultados son directos. Para $|00\rangle$ se obtiene:

$$|\psi_{12}^1\rangle = \Big(-\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \eta|11\rangle\Big) \otimes |q\rangle \tag{91}$$

Si la solución es $|01\rangle$:

$$|\psi_{13}^1\rangle = \Big(\alpha|00\rangle - \beta|01\rangle + \gamma|10\rangle + \eta|11\rangle\Big) \otimes |q\rangle \tag{92}$$

Para $|10\rangle$ como solución:

$$|\psi_{14}^1\rangle = \Big(\alpha|00\rangle + \beta|01\rangle - \gamma|10\rangle + \eta|11\rangle\Big) \otimes |q\rangle \tag{93}$$

Finalmente, si el objetivo del problema de búsqueda es el estado $|11\rangle$:

$$|\psi_{15}^1\rangle = \Big(\alpha|00\rangle - \beta|01\rangle + \gamma|10\rangle + \eta|11\rangle\Big) \otimes |q\rangle \tag{94}$$

El comportamiento es el esperado en teoría, por lo que se comprueba efectivamente que son oráculos válidos.

6.6. Programa

Importing standard Qiskit libraries and loading IBM Q account###

%matplotlib inline

from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister, execute, Aer, IBMQ, BasicAer

from qiskit.ignis.mitigation.measurement import complete_meas_cal, CompleteMeasFitter

from qiskit.providers.aer.noise import NoiseModel

from qiskit.tools.monitor import job_monitor

```
from qiskit.visualization import plot_histogram
provider = IBMQ.load_account()

### Building the circuit ###
barriers = True
scenario = ' 1 ' # Choose scenario
def fun_barrier(barriers):
        if barriers == True:
                circuit.barrier()

if scenario == '1':
        q = QuantumRegister(2)
        c = ClassicalRegister(2)
        circuit = QuantumCircuit(q, c)
        circuit.h(q)
        barrier = fun_barrier(barriers)
        circuit.x(q)
        circuit.cz(q[0],q[1])
       circuit.x(q)
        barrier = fun_barrier(barriers)
        circuit.h(q)
        barrier = fun_barrier(barriers)
        circuit.z(q)
        circuit.cz(q[0],q[1])
        barrier = fun_barrier(barriers)
        circuit.h(q)
        barrier = fun_barrier(barriers)
```

```
if scenario == '2':
        q = QuantumRegister(3)
        c = ClassicalRegister(3)
        circuit = QuantumCircuit(q, c)
        circuit.x(q[2])
        circuit.h(q[0:2])
        barrier = fun_barrier(barriers)
        circuit.x(q[0:2])
        circuit.h(q[2])
        circuit.ccx(q[0],q[1],q[2])
        circuit.h(q[2])
        circuit.x(q[0:2])
        barrier = fun_barrier(barriers)
        circuit.h(q[0])
        circuit.h(q[1])
        barrier = fun_barrier(barriers)
        circuit.z(q[:2])
        circuit.cz(q[0],q[1])
        barrier = fun_barrier(barriers)
        circuit.h(q[0:2])
        barrier = fun_barrier(barriers)

if scenario == '3':
        q = QuantumRegister(2)
        c = ClassicalRegister(2)
        circuit = QuantumCircuit(q, c)
        circuit.h(q)
        barrier = fun_barrier(barriers)
```

```
        circuit.z(q)
        barrier = fun_barrier(barriers)
        circuit.h(q)
        barrier = fun_barrier(barriers)
        circuit.z(q)
        circuit.cz(q[0],q[1])
        barrier = fun_barrier(barriers)
        circuit.h(q)
        barrier = fun_barrier(barriers)

    if scenario == '4':
        q = QuantumRegister(3)
        c = ClassicalRegister(3)
        circuit = QuantumCircuit(q, c)
        circuit.h(q)
        barrier = fun_barrier(barriers)
        circuit.cz(q[0],q[2])
        circuit.cz(q[1],q[2])
        barrier = fun_barrier(barriers)
        circuit.h(q)
        barrier = fun_barrier(barriers)
        circuit.x(q)
        circuit.h(q[2])
        circuit.ccx(q[0],q[1],q[2])
        circuit.h(q[2])
        circuit.x(q)
        barrier = fun_barrier(barriers)
        circuit.h(q)
```

```
            barrier = fun_barrier(barriers)

    circuit.measure(q,c)
    circuit.draw(output='mpl')

    ### Simulation without noise ###
    backend_ideal = BasicAer.get_backend('qasm_simulator')
    results_ideal = execute(circuit, backend=backend_ideal, shots=1024).result()
    counts_ideal = results_ideal.get_counts()
    plot_histogram(counts_ideal)

    ### Simulation with basic noise model ###
    backend = provider.get_backend('ibmq_london') #choose backend
    noise_model = NoiseModel.from_backend(backend)
    coupling_map = backend.configuration().coupling_map
    basis_gates = noise_model.basis_gates
    result_1 = execute(circuit, Aer.get_backend('qasm_simulator'), coupling_map=
coupling_map, basis_gates=basis_gates, noise_model=noise_model).result()
    counts_1 = result_1.get_counts(0)
    plot_histogram(counts_1)

    ### Real devices ###
    job_2 = execute(circuit, backend=backend, shots=1024)
    job_monitor(job_2, interval = 2)
    result_2 = job_2.result()
    counts_2 = result_2.get_counts(circuit)
    plot_histogram(counts_2)

    ### Error mitigation ###
```

cal_circuits, state_labels = complete_meas_cal(qr = circuit.qregs[0], circlabel = 'measurementmitigationcal')

job_3 = execute(cal_circuits, backend= backend, shots = 1024, optimization_level = 0)

job_monitor(job_3)

results_3 = job_3.result()

meas_fitter = CompleteMeasFitter(results_3, state_labels)

meas_fitter.plot_calibration()

meas_filter = meas_fitter.filter

mitigated_result = meas_filter.apply(result_2)

results_4 = result_2.get_counts(circuit)

counts_4 = mitigated_result.get_counts(circuit)

plot_histogram([results_4, counts_4], legend=['device, noisy', 'device, mitigated'])

Agradecimientos

En primer lugar me gustaría agradecer a Joaquín Fernández Rossier, tutor de este trabajo de fin de grado, el interés mostrado sobre el tema desde un primer momento. Asimismo quiero expresar mi gratitud por el tiempo y dedicación, además de guiarme en este trabajo. Gracias también a Jorge Calera Rubio por el tiempo dedicado y por la ayuda recibida al resolver varias dudas. Por último, me gustaría agradecer a IBM por crear la plataforma IBM Q.

Referencias

[1] Chow, J. (2014). Dealing with errors in quantum computing, *IBM Research Blog.*

[2] Colin P. (2011). Explorations in Quantum Computing, 241–262. Springer-Verlag, London.

[3] Hui, J. (2019). QC – How to build a Quantum Computer with Superconducting Circuit?

[4] Fernández-Rossier, J. (2018). *Guión de la segunda sesión de prácticas*, Mecánica Cuántica I, Grado en Física UA, curso 2018-2019.

[5] Figgatt C., Maslov D., Landsman K.A., Linke N. M., Debnath S. and Monroe C. (2017). Complete 3-Qubit Grover Search on a Programmable Quantum Computer. arXiv:1703.10535

[6] Nielsen, M.A. and Chuang, L.A. (2000). Quantum Computation and Quantum Information. Cambridge University Press, Cambridge.

[7] Wright, J. (2015). Lecture 4: Grover's Algorithm. *Carnegie Mellon University.*

[8] Backend configuration. *IBM Quantum Experience.* https://quantum-computing.ibm.com/docs/cloud/backends/configuration

[9] Grover's Algorithm. *Qiskit Textbook.* https://qiskit.org/textbook/ch-algorithms/grover.html

[10] IBM Quantum backends. *IBM Quantum Experience.* https://quantum-computing.ibm.com/docs/cloud/backends/

[11] Measurement Error Mitigation. *Qiskit Textbook.* https://qiskit.org/textbook/ch-quantum-hardware/measurement-error-mitigation.html

[12] Noise Models. *Qiskit Documentation.*

https://qiskit.org/documentation/apidoc/aer_noise.html

[13] Solving Satisfiability Problems using Grover's Algorithm. *Qiskit Textbook.* https://qiskit.org/textbook/ch-applications/satisfiability-grover.html

[14] What is quantum computing? https://www.ibm.com/quantum-computing/learn/what-is-quantum-computing/