Trabajo Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

# Estudio sobre computación cuántica con aplicación a particionado de grafos

Autor: Miguel Antonio Sanz Pérez
Tutor: José María Maestre Torreblanca,
Ascensión Zafra Cabeza

**Dpto. Ingeniería de Sistemas y Automática**
**Escuela Técnica Superior de Ingeniería**
**Universidad de Sevilla**

Sevilla, 2019

Trabajo Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

# Estudio sobre computación cuántica con aplicación a particionado de grafos


Autor:

Miguel Antonio Sanz Pérez

Tutor:

José María Maestre Torreblanca

Profesor titular

Ascensión Zafra Cabeza

Profesor titular

Dpto. Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla



Sevilla, 2019

Trabajo de Fin de Grado: Estudio sobre computación cuántica con aplicación a particionado de grafos

Autor: Miguel Antonio Sanz Pérez

Tutor: José María Maestre Torreblanca,
Ascensión Zafra Cabeza

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Sevilla, 2019

# Agradecimientos

Agradecer a todos mis amigos y compañeros que me han acompañado durante este periodo de mi vida tan bueno y fructífero, pero que sin embargo es la base para un futuro que espero me permita vivir nuevas experiencias y conocer a muchas personas que me aporten tanto como ellos me han aportado a mí.

Mención especial a todos los profesores que me han ayudado ha llegar hasta aquí y haberme inculcado la importancia de la educación, desde primaria hasta la universidad.

Y como no podría ser de otra manera, a mi familia; mi padre, mi madre y mi hermana por apoyarme en todo momento y confiar en mi incluso cuando ni yo confiaba en mis posibilidades. Por aconsejarme y educarme para llegar a lo que soy y lo que el día de mañana seré. Muchas gracias.

*Miguel Antonio Sanz Pérez*

*Sevilla, 2019*

# Resumen

En los últimos años se ha aumentado el interés de conceptos cuánticos como la superposición o el entrelazamiento cuántico, así como el fenómeno de la teleportación cuántica. Esto se debe principalmente a un nuevo concepto que ha surgido de la aplicación de estas propiedades cuánticas al mundo de la computación, cuyo nombre es computación cuántica.

En este trabajo se va a realizar una breve descripción de estos conceptos cuánticos así como de los conceptos que surgen de la aplicación de la mecánica cuántica a la computación, como el qubit, las puertas lógicas cuánticas o el teorema de no clonación. Todo esto será de gran ayuda para poder abordar el verdadero objetivo del trabajo, aplicar estos conceptos revolucionarios a un problema combinatorio que en la actualidad, debido a su dificultad computacional, es imposible resolver para aplicaciones real. Este es el caso del problema de máximo corte, un problema combinatorio cuya solución no puede ser obtenida en tiempo polinomial por un ordenador clásico, pero en cambio, las propiedades cuánticas permiten realizar cálculos simultáneos de múltiples estados mediante la superposición y el entrelazamiento cuántico de los qubits.

Por ello, se ha elegido un algoritmo cuántico que se aproveche de estas propiedades y consiga obtener resultados prometedores. Este es el caso de QAOA (Quantum Approximate Optimization Algorithm), un algoritmo que permite resolver varios problemas combinatorios de tipo NP (no polinomiales) de forma aproximada. Además, este algoritmo tiene múltiples beneficios frente a otros algoritmos cuánticos utilizados para resolver problemas de este tipo, entre los que destaca la posibilidad de utilizar este algoritmo para poder demostrar la tan ansiada por muchos expertos supremacía cuántica, la cual se alcanzará cuando se consiga obtener resultados mejores en un ordenador cuántico real que en uno clásico.

Al final del trabajo, se realizarán simulaciones de este algoritmo en un ordenador clásico con un software que permite simular algoritmos cuánticos, y de donde se obtendrán resultados de casos sencillos, pero que pueden ser extrapolados para casos más complejos.

# Abstract

Last years, interest in quantum concepts such as superposition or quantum entanglement has increased, as well as the quantum teleportation. This is mainly due to a new idea that has appeared from the application of these quantum properties to computing, whose name is quantum computing

In this dissertation we will make a short description of these quantum ideas and the concepts that appear from the application of quantum mechanics to computing, such as qubit, quantum logic gates or the non-cloning theorem. All this will help to achieve the objective of this dissertation; apply these revolutionary concepts to a combinatorial problem that nowadays, due to its computational difficulty, is impossible to solve for complex cases. For example, max-cut problem is a combinatorial problem whose solution cannot be obtained in polinomial time (NP problem), but quantum properties allow to simultaneous calculations of many states due to superposition and quantum entanglement of qubits.

Therefore, a quantum algorithm has been chosen for this problem. QAOA (Quantum Approximate Optimization Algorithm) is an algorithm that allows to solve some combinatorial problems of type NP (not polynomial time) approximately. In addition, this algorithm has multiple benefits when it is compared to other quantum algorithms that have been used to solve problems like the max-cut. One of those benefits is the possibility of using this algorithm to demonstrate the expected quantum supremacy, which will be achieved when a real quantum computer improves the results of a supercomputer.

# Índice

**Agradecimientos** **vii**

**Resumen** **ix**

**Abstract** **xi**

**Índice** **xiii**

**Notación** **xv**

**1 Conceptos previos** **1**
*1.1 Superposición* *1*
1.1.1 Paralelismo cuántico 2
*1.2 Entrelazamiento cuántico* *2*
1.2.1 Paradoja EPR 3
*1.3 La medida cuántica* *4*

**2 Computación cuántica** **7**
*2.1 Qubit* *7*
*2.2 Esfera de Bloch* *8*
*2.3 Teorema de no clonación* *9*
*2.4 Procesadores cuánticos* *10*
2.4.1 Quantum annealing 10
*2.5 Simuladores cuánticos* *11*
2.5.1 Quil 11
2.5.2 OpenQuasm 12
2.5.3 Qiskit 12
2.5.4 Forest 12
2.5.5 Quantum Development Kit y Q# 13
2.5.6 QMASM 13

**3 Puertas cuánticas** **15**
*3.1 Hadamard* *15*
*3.2 Pauli* *16*
3.2.1 Pauli-X gate 16
3.2.2 Pauli-Y gate 16
3.2.3 Pauli-Z gate 17
*3.3 Desplazamiento de fase* *17*
*3.4 SWAP* *18*
*3.5 CNOT* *18*

**4 Problema de máximo corte** **19**
*4.1 Formulación del problema* *20*
*4.2 Descripción geométrica* *21*
*4.3 Métodos clásicos de solución* *23*
4.3.1 Branch and Bound 23
4.3.2 Gráfica plana 25
4.3.3 Goemans-Williamson 25

**5 QAOA** **27**
*5.1 Introducción* *27*
*5.2 Adaptación del máximo corte* *28*
*5.3 Fundamento teórico* *29*
*5.4 Desarrollo matemático* *31*
*5.5 Implementación* *34*
*5.6 Instalación y ejecución* *36*

**6 Resultados** **39**
*6.1 Gráfica de un cuadrado* *39*
*6.2 Gráfica de una línea recta* *41*
*6.3 Gráfica de una red compleja* *42*

**7 Conclusión y futuras ampliaciones** **49**

**Apéndice A: Código Python** **51**

**Índice de Tablas** **53**

**Índice de Figuras** **55**

**Referencias** **57**

# Notación

| | |
|---|---|
| $\oplus$ | Operador XOR |
| e | Número e |
| $\|\Psi\rangle$ | Función de onda de Ψ |
| sen | Función seno |
| cos | Función coseno |
| = | Igual que |
| < | Menor que |
| > | Mayor que |
| $\geq$ | Mayor o igual que |
| $\leq$ | Menor o igual que |
| i | Número complejo |
| º | Grados (ángulo) |
| QPU | Quantum Processor Unit |
| CNOT | Controlled NOT |
| NP | Non-Polinomial |
| $\overline{S}$ | Complementario de S |
| $\in$ | Pertenece a |
| $\notin$ | No pertenece a |
| Max | Máximo |
| s.a. | Sujeto a |
| $X'$ | Transpuesta de X |
| VLSI | Very Large Scale Integration |
| Lim | Límite |
| QAOA | Quantum Approximate Optimization Algorithm |
| QAA | Quantum Adiabatic Algorithm |
| qb | qubit |

# 1 CONCEPTOS PREVIOS

*Anyone who is not shocked by*
*quantum theory has not understood it.*

Niels Bohr

La computación cuántica tiene especial atención tanto por parte del ámbito científico como del tecnológico debido a la aparición de algoritmos que podrían suponer una mayor capacidad de cómputo respecto a los ordenadores convencionales e incluso que los superordenadores desarrollados en estos últimos años. Esto se debe a su base teórica, puesto que estos ordenadores cuánticos utilizan propiedades físicas como lo son la superposición o el entrelazamiento cuántico permitiendo que una sola unidad de cómputo pueda llegar a tomar idealmente cualquier valor.

Para conocer mejor estas propiedades físicas de la mecánica cuántica, se realiza en este capítulo una breve descripción de qué son y cómo influyen en el diseño de los algoritmos cuánticos, así como de algunos fenómenos interesantes de la mecánica cuántica y sus aplicaciones en la computación.

## 1.1 Superposición

La superposición de estados se define como la propiedad que tienen las partículas cuánticas mediante la cual, estas pueden encontrarse en distintos estados a los que se les atribuye una probabilidad de que al ser medida la partícula, sea obtenido un estado u otro [1-3]. Esto se puede ver reflejado en la siguiente ecuación:

$$|\Psi\rangle = c_1 * S_1 + c_2 * S_2 \tag{1.1}$$

Donde $|\Psi\rangle$ es la función de onda de una determinada partícula, $c_1\ y\ c_2$ son los coeficientes cuya raíz cuadrada es la probabilidad de dicho estado y $S_1\ y\ S_2$ son los estados posibles en los que se encontraría la partícula. Una de las paradojas más famosas donde se puede observar la implicación de esta propiedad es la ideada por Erwin Schrödinger en 1935 conocida como la paradoja del Gato de Schrödinger que consiste en introducir un gato dentro de una caja sellada y tapada para que no pueda ser observado el estado del gato. En esta caja existiría un veneno, una partícula radiactiva y un detector de radiación; esto permitirá que cuando se descomponga la partícula, se detectará y se liberará el veneno que matará al gato. Puesto que el tiempo de desintegración no se puede conocer con total exactitud, el gato estará en una superposición entre vivo y muerto, y solo se podrá conocer su estado con total exactitud cuándo se destape la caja y se observe su estado, que sería lo mismo que en el caso de la función de onda, hacerla colapsar a un estado concreto mediante la observación. Por lo tanto, es la propia acción de realizar la observación la que interactúa con la partícula y provoca que esta deje de estar en una superposición de estados, y por lo tanto pase a colapsar a un solo estado. Este procedimiento de observación podría ser por ejemplo, iluminar la partícula para que pueda ser observada, interfiriendo con la misma al excitarla con fotones. Por ello, es tan complicada la creación de un computador cuántico, ya que para evitar que colapse la función de onda, la partícula ha de estar perfectamente aislada y sin que puedan existir posibles interferencias al manipularla.

Por todo esto, la superposición de estados se podría considerar uno de los pilares fundamentales que permiten que la computación cuántica tenga las características que le permite obtener unos resultados en cuanto a tiempo

de ejecución superlativos frente a la computación clásica basada en transistores que solo pueden tomar el estado 0 → apagado/no conduce, o el estado 1 →encendido/en conducción.

### 1.1.1 Paralelismo cuántico

El paralelismo cuántico es uno de los efectos más notorios (sino el que más) de la aplicación de la superposición cuántica a los algoritmos en los ordenadores cuánticos. Este se basa en la posibilidad de realizar múltiples operaciones sobre un mismo estado cuántico, dando lugar a un incremento en la capacidad computacional. Este fenómeno se aplica en el algoritmo de Deutsch, el cual se basa en conocer si una determinada función es constante o balanceada, de tal forma que solo evaluando un solo qubit (unidad básica de computación cuántica), ya se puede conocer qué tipo de función es, mientras que en los algoritmos clásicos, sería necesario realizar la comprobación en el peor de los casos para la mitad de los posibles valores de la función.

Para comprender mejor como funciona este fenómeno, se plantea realizar la suma de dos qubits. El primero de ellos estará en el estado 1, mientras que el segundo estará tomando el valor

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{1.2}$$

Por lo tanto, el valor final de la suma, debería de ser que el primer qubit tome el valor 1, mientras que el segundo tome el valor 1 o 0, y a su vez, el tercer qubit, que se encargaría de mostrar el acarreo, también estaría en la superposición de valores 0 o 1.

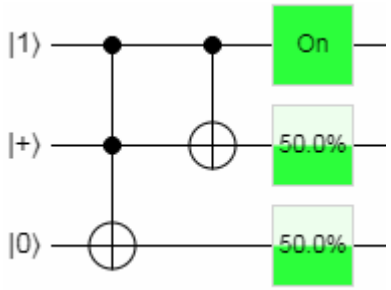


Figura 1-1. Ejemplo de suma de 2 qubits (puertas CCNOT y CNOT)

## 1.2 Entrelazamiento cuántico

El entrelazamiento cuántico es una propiedad de la mecánica cuántica en la que un conjunto de partículas cuánticas están correlacionadas (entrelazadas) y por lo tanto sus estados no pueden describirse de forma separada o individualmente, sino que existe una dependencia entre las partículas que pertenecen a dicho conjunto. Este fenómeno fue expuesto por los científicos Albert Einstein, Boris Podolsky y Nathan Rosen mediante la paradoja EPR, aunque sería Erwin Schrödinger poco tiempo después quien diese nombre a este fenómeno. Como consecuencia de la realización de experimentos, se sabía que debía existir alguna propiedad que permitía que los estados de dos o más partículas estuviesen enlazados hasta el punto de que cuando se realizaba alguna observación sobre una de las partículas del sistema, y por lo tanto se hacía colapsar la función de onda de la misma, la otra partícula también colapsaba, obteniéndose por tanto su estado; se realizaron múltiples interpretaciones y suposiciones por los científicos del momento acerca de cuál podría ser el origen de este extraño fenómeno. Esto se vio fuertemente incitado por la paradoja EPR descrita por los tres científicos expuestos anteriormente (Einstein-Podolsky-Rosen), la cual se explicará en profundidad más adelante.

Por otra parte, en lo que a computación cuántica concierne, esta es una de las propiedades más importantes, ya que es la causante entre otras, de que los algoritmos cuánticos en su marco teórico puedan obtener resultados mucho mejores que los clásicos. Esto se debe a que al entrelazar dos o más qubits, se pueden realizar operaciones sobre esta superposición de estados que afectará al sistema, y por tanto, a cado uno de los qubits que están superpuestos y entrelazados.

Para una mejor comprensión de este fenómeno, es necesario explicar como se puede generar dicho entrelazamiento desde el punto de vista de la computación. Para simplificar, se realizará sobre dos qubits, ya que

es la cantidad mínima necesaria para poder generar entrelazamiento. En primer lugar, se muestra una función de onda en la que no existe entrelazamiento a pesar de que exista superposición de dos qubits

$$|\Psi\rangle = \frac{|01\rangle + |11\rangle}{\sqrt{2}} \tag{1.3}$$

Ya que como se mencionó anteriormente, para que haya entrelazmamiento es necesario que los qubits del sistema no puedan ser separados de forma individual, pero en este caso, esto no es así como se puede ver en la ecuación 1.4

$$|\Psi\rangle = \frac{(|0\rangle + |1\rangle) \oplus |1\rangle}{\sqrt{2}} \tag{1.4}$$

Puesto que el primer qubit se encuentra en la superposición de estado 0 y 1, mientras que el segundo qubit se encuentra en el estado 1. Por lo tanto, existe en este caso la posibilidad de separar la función de onda en sus dos qubits que la forman. A continuación se muestra una función de onda en la que sí que existiría entrelazamiento cuántico

$$|\Psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{1.5}$$

Esta ecuación a diferencia de la 1.2, no puede ser separada de forma individual en 2 qubits. Este estado entrelazado se corresponde a su vez con uno de los estados de Bell, los cuales son perfectos ejemplos de estados entrelazados.

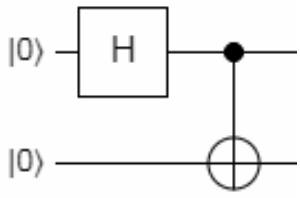


Figura 1-2. Ejemplo de entrelazamiento de 2 qubits (puerta H+CNOT)

### 1.2.1 Paradoja EPR

Esta paradoja recibe su nombre debido a los apellidos de los científicos que la enunciaron en 1935 (Einstein-Podolsky-Rosen). Esta paradoja se basa en la posibilidad de transmitir información de forma instantánea debido al entrelazamiento cuántico. A esta transmisión de información también se le conoce como teleportación cuántica, ya que si se tienen dos partículas entrelazadas y que están separadas, al realizar la observación sobre una de ellas, la función de onda de la misma colapsaría y por lo tanto, la otra partícula del sistema entrelazado también quedaría determinada según el valor que haya tomado la partícula observada. Esto provoca que se realice la transmisión de información de forma instantánea y sin importar la distancia a la que se encuentre, por lo que se suele utilizar el concepto de teleportación. Esto es imposible puesto que violaría la teoría de la relatividad de Albert Einstein que fija la velocidad máxima en la velocidad de la luz, por lo que el fenómeno de entrelazamiento cuántico de las partículas derrumbaría la teoría de la relatividad de Einstein.

La exposición de esta paradoja no busca mostrar que la mecánica cuántica es errónea, sino que simplemente es una teoría incompleta, ya que la teoría no respeta el principio de localidad, y por lo tanto propone la utilización de variables ocultas que permiten que la teoría sea determinista y se respete la localidad. Esto era lo que pensaba Einstein en aquel momento, ya que no podía pensar que fuese posible que la acción de medición sobre una partícula que se encuentra alejada de otra, pueda llegar a afectarla (no localidad de la mecánica cuántica), y a su vez, pensaba que sería necesario que existiesen algunas variables ocultas (variables/propiedades de las partículas que se desconocen y que por lo tanto provoca que la mecánica cuántica sea probabilística y no determinista). Esta paradoja tuvo un fuerte rechazo por numerosos científicos que defendían el carácter

probabilístico de la mecánica cuántica. A pesar de esto, no sería hasta 1964 cuando John Bell plantease las desigualdades de Bell que han permitido mediante numerosos experimentos dar la razón a la mecánica cuántica y dejar al planteamiento de los tres físicos (EPR) como una simple paradoja.

Otra forma de ver la solución a la paradoja que no sea mediante las desigualdades de Bell sería que para que pudiese realmente transmitir información entre los dos observadores, sería necesario que el primero que realiza la medición y obtenga los resultados, comunique al otro observador la base utilizada para realizar la medida, Esta comunicación sería puramente clásica, por lo que será como máximo a la velocidad de la luz. En cambio, el segundo observador podría realizar múltiples copias del estado cuántico de la partícula, y realizar mediciones en todas las bases posibles, obteniendo el resultado buscado, pero esto es imposible debido al teorema de no clonación que se verá más adelante.

A pesar de todo esto, el entrelazamiento cuántico está muy ligado a las nuevas metodologías de encriptación cuántica (o post-cuántica), ya que en el marco teórico, el algoritmo de Shor permitiría desencriptar los mensajes que utilizan clave pública como RSA, ya que permite descomponer en factores un número en un tiempo mucho más reducido que los ordenadores clásicos. Por ello, surge la necesidad de desarrollar una nueva metodología, y puesto que el entrelazamiento cuántico permite el envío de información entre dos observadores sin que exista un medio de propagación por el que discurra la información, ya que dicha información estaría contenida en las partículas que se encuentran entrelazadas.

## 1.3 La medida cuántica

Otro aspecto importante relacionado con la computación cuántica, y que a su vez difiere completamente de la concepción clásica de computación es el proceso de medida [4]. Hasta ahora se han expuesto algunos fenómenos que permiten que la computación cuántica tenga unas características muy prometedoras frente a cierto tipo de algoritmos y aplicaciones, pero estos beneficios también repercuten en otros aspectos como lo es en este caso la medida. Como se ha visto, el estado de un bit en un ordenador clásico se restringe a 0 o 1, pero en un ordenador cuántico, este puede tomar esos dos valores de forma superpuesta y con una probabilidad asociada; y por ello, a la hora de realizar la medida es muy importante tener en cuenta esto. Es por esto, que al proceso de medida de un estado cuántico se le conoce también como el colapso de la función de onda, ya que se pasa de tener una función de onda en la que existen múltiples estados superpuestos a un valor concreto, como el que se tendría en un ordenador clásico.

Como se ha explicado, el proceso de medida permite pasar de una onda de probabilidad en la que sus distintos estados posibles fluctúan continuamente a un valor determinista, y es por ello que surge otro problema añadido al de medir, y es el de interferencia, ya que cuando se realizan operaciones sobre un qubit o un conjunto de ellos, es muy importante que estos estén completamente aislados del exterior o de lo contrario la función de onda colapsará y las propiedades de la mecánica cuántica desaparecen con dicho colapso. Es por ello, que uno de los principales problemas que existen en la actualidad a la hora de desarrollar ordenadores cuánticos es el tratamiento y aislamiento que se hace con las partículas cuánticas y por lo tanto, provoca que en la actualidad, el número de qubits de un ordenador sea muy inferior al número de bits de un ordenador convencional. A su vez, esta escasez de unidades de cálculo provoca que por el momento siga siendo mucho mejor la tecnología basada en el silicio (ordenadores clásicos) frente a la tecnología cuántica, y por lo tanto, se reduzca al ámbito teórico la posibilidad que los algoritmos cuánticos lleguen a superar a los clásicos.

Aunque la interferencia tenga consecuencias similares al de la medida, ya que se produce el colapso de la función de onda, la interferencia es una acción indeseada en nuestro sistema mientras que la medida es necesario para poder obtener resultados del algoritmo que se ha implementado, y es por ello, que se va a exponer en mayor profundidad este concepto. Como ya se ha mencionado anteriormente, en el mundo macroscópico (en el que estamos acostumbrados y las leyes físicas básicas se cumplen) los objetos están determinados: su forma, color, brillo, posición, velocidad, momento,… en cambio en el mundo cuántico esto es radicalmente distinto puesto que estas magnitudes comienzan a carecer de sentido o simplemente están indeterminadas y lo único que se puede saber es la probabilidad de que tomen un valor u otro. Es por ello que mientras que en la computación clásica, el efecto de medir un bit sería el de simplemente hacer aflorar su valor en ese momento, ya que en todo momento el bit tiene determinado su valor sin importar que esté siendo observado o no. En cambio, para un qubit el proceso de medida incide directamente en su naturaleza, provocando que pase de estar en un estado indeterminado (probabilístico) a un valor concreto y determinado. Este es uno de los conceptos más extraños y

complejos de la mecánica cuántica, y es objeto de debate todavía en la actualidad siendo llamado comúnmente como el problema de la medida cuántica, y el cuál tiene distintas interpretaciones desde la comunidad de científicos. A continuación se presentan algunas de las más importantes y conocidas para que se pueda entender un poco mejor la concepción que existe en la actualidad sobre cómo se produce esté fenómeno de la medida:

- **Interpretación de Copenhague:** esta interpretación es la más compartida por la comunidad científica y la predominante en las explicaciones educativas. Esta fue formulada por Niels Bohr y Werner Heisenberg, y se basa en que el proceso de medida toma un valor aleatorio de los posibles en la función de onda, dependiendo de las probabilidades asociadas a cada estado. En este caso, el acto de observación por un observador cualquiera es el que provoca que la función de onda colapse y por lo tanto pase a tomar un valor determinado. Un gran número de experimentos concuerdan con este pensamiento.
- **Many worlds:** en este caso, se introduce la posibilidad de que existan múltiples universos, y que por lo tanto, cada vez que se realiza una observación, dicho universo se desdobla de forma que en cada uno de ellos se observa un valor distinto. Esta interpretación no termina de resolver el problema de medida, ya que simplemente deriva el problema hacia la aparición de un universo o varios universos paralelos en los que se dan cada uno de los posibles estados de la función de onda.
- **Interpretación de Bohm:** se basa en la existencia de variables ocultas; es decir, variables de las que se desconoce su valor, pero que en caso de conocerse, el valor de la medición sería determinista incluso antes de realizarse la medición, ya que este valor sería conocido en todo momento, como sucede en el mundo macroscópico. Esta teoría es por ello determinista y el concepto de colapso carece de sentido puesto que la función de onda no colapsa hacia un valor, ya que dicho valor ya estaba determinado con anterioridad.

Como se puede observar, hasta el momento se ha ido tomando la interpretación de Copenhague para realizar la explicación, y por lo tanto, se continuará haciendo a lo largo de todo el proyecto puesto que es la más extendida y conocida, además de que es fuertemente respaldada por la experimentación.

A pesar de todo esto, existe una nueva propuesta en la actualidad conocida como decoherencia, la cual explica porque si las partículas elementales se basan por ecuaciones probabilísticas, el mundo macroscópico que está formado por el conjunto de muchas partículas es determinista. Para ello, expone que intentar analizar una partícula aislada (por ejemplo un fotón) es un caso concreto y además no se le permite interaccionar con el entorno, ya que está completamente aislado. Es por ello que lo que refleja el mundo macroscópico es consecuencia de que existen múltiples observadores, muchos fotones interaccionando entre si y que provocan que la función de onda colapse continuamente y por lo tanto se observe un carácter determinista en los objetos y todo lo que nos rodea. A pesar de que la decoherencia permite realizar una relación o explicación de porqué son tan distintos el mundo macroscópico de la mecánica cuántica, no es capaz de dar una explicación al fenómeno del colapso para las partículas cuánticas.

Por último, cuando se realiza un circuito lógico cuántico, a diferencia de en un circuito clásico, es necesario utilizar una puerta concreta que muestre que se realiza una operación de medición y que por lo tanto colapsará la función de onda, dando lugar a que los estados en los que se pueda encontrar el qubit pasen a tomar un valor concreto. El símbolo utilizado para dicha puerta se puede observar en la figura 1-3.

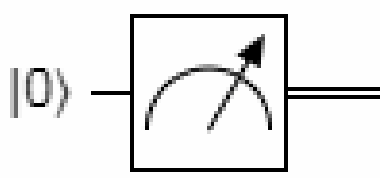


Figura 1-3. Puerta de medición

# 2 COMPUTACIÓN CUÁNTICA

En este segundo capítulo se desarrollará el concepto de computación cuántica abordado ligeramente en los conceptos previos, así como cuáles son sus elementos diferenciadores respecto a la tecnología de computación clásica predominante en la actualidad, pero que si continúa el avance de la cuántica y sus algoritmos, podría llegar a quedar obsoleta en campos tan importantes como la criptografía o la resolución de problemas de optimización que en la actualidad no pueden ser resueltos en tiempo polinomial, mientras que con la aparición de estos nuevos ordenadores, esto podría llegar a cambiar radicalmente.

Además, se realizará un breve recorrido sobre las tecnologías actuales que se están desarrollando para poder construir el ordenador cuántico, ya sea tanto a nivel de hardware con los procesadores cuánticos, como a nivel de software, con los simuladores de ordenadores cuánticos que más se utilizan en la actualidad, así como el que será utilizado para este proyecto.

## 2.1 Qubit

El primer concepto que se va a abordar es el de qubit, el cual es la unidad fundamental de la computación cuántica, así como lo es el bit en la computación clásica [1-3,5]. Esta unidad fundamental puede tomar infinidad de valores, cuya base queda definida en el espacio mediante los dos vectores siguientes que se asemejan a los estados 0-1 de un bit convencional.

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, |1\rangle = \begin{pmatrix}0\\1\end{pmatrix} \tag{2.1}$$

A pesar de la similitud, existe una gran diferencia entre ambos, puesto que mientras que un bit puede tomar solo los valores 0 o 1 (encendido/conduce o apagado/no conduce), un qubit aprovecha el fenómeno cuántico conocido como superposición expuesto anteriormente, y que permite que cada qubit pueda encontrarse en una superposición de los dos estados vistos anteriormente, dando a la siguiente equivalencia:

$$nº\ estados = 2^{qubit}$$

Esto permite que conforme se aumenta el número de qubits de un ordenador cuántico, su potencia de cálculo respecto a un ordenador convencional aumente considerablemente. A pesar de ello, la construcción de este tipo de ordenadores en la actualidad está en pleno desarrollo y la dificultad a la que se enfrentan físicos, matemáticos e ingenieros cada vez que se intenta aumentar el número de qubits también aumenta exponencialmente, ya que como se ha visto en los conceptos básicos, la mecánica cuántica ofrece ventajas respecto a la potencia de cálculo, pero en cambio, también produce numerosos problemas tanto en la fase de construcción, como en la de obtención de resultados una vez se ha ejecutado un programa. Por ello, se sigue investigando y destinando recursos a la materia a pesar de que existan numerosas voces críticas que argumenten que no se podrá llegar a alcanzar la potencia de cálculo que existe con los ordenadores comerciales, o que debido a su complejidad, estos ordenadores nunca lleguen a ser comercializados para el gran público y se limiten a funciones de investigación o cálculos muy concretos donde sus ventajas se hagan notar, pero siempre acompañado y sustentado por un ordenador convencional.

## 2.2 Esfera de Bloch

Un concepto muy importante y utilizado a la hora de representar el estado en el que se encuentra un qubit es la esfera de Bloch, ya que se pueden englobar todos los posibles estados en los que se encuentra un qubit en una esfera, de la misma forma que un bit convencional se puede representar como un interruptor que está encendido o apagado. Para ello, el eje Z sirve para representar los estados $|0\rangle$ y $|1\rangle$, donde el la parte positiva del eje sería el estado 0 mientras que la negativa sería el estado 1.

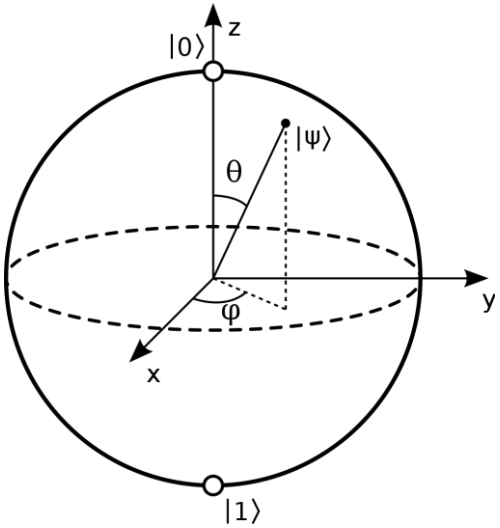


Figura 2-1. Representación de la esfera de Bloch para los estados de un qubit.

En el siguiente capítulo se expondrán cuáles son las puertas lógicas cuánticas más conocidas actualmente, y para poder visualizar cuales son los cambios que se le realiza a un qubit tras aplicarle la puerta lógica, se utiliza la esfera de Bloch, ya que esto permite visualizar muy claramente qué tipo de variación se le está realizando al qubit, y como esta afecta a la probabilidad de medir el estado 0 o 1. Además, es necesario recalcar que esta esfera pertenece al espacio complejo, por lo que existen regiones en las cuales la función de onda tomará valores complejos, aunque estos no afectan a la probabilidad, ya que como se explicó anteriormente, la probabilidad de que se mida un qubit en un determinado estado u otro surge de realizar el módulo al cuadrado del coeficiente que acompaña al estado 0 o 1, por lo que el término complejo 'i', al realizarle el cuadrado y posterior módulo provoca que no varía la probabilidad.

En siguiente lugar, es necesario destacar que la esfera de Bloch permite conocer rápidamente la probabilidad de que un determinado qubit tome un valor u otro, pero no es una fiel representación sobre el estado en su conjunto del qubit, ya que si se parte del estado 0, y se realiza una rotación de 180º entorno al eje Y, el valor esperado sería que el qubit estuviese en el estado 1, pero sin embargo, el valor real es el de $i|1\rangle$, que aunque no afecta a la probabilidad de la función de onda, ya que el qubit se encontraría en el estado 1 en el 100% de los casos, el término complejo que aparece no sería el esperado según la representación geométrica.

Esto último se debe a que la ecuación que rige la esfera de Bloch es la siguiente:

$$|\Psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + (cos\ \theta + i * sen\varphi) * \text{sen}\left(\frac{\theta}{2}\right)|1\rangle \tag{2.2}$$

Donde

$$0 \leq \theta < \pi\ , 0 \leq \varphi < 2\pi \tag{2.3}$$

Por lo que si se realiza un giro de 180º entorno al eje Y, lo que equivaldría a girar 90º $\varphi$ y 180º $\theta$, se obtiene que el término del estado 0 se anula, mientras que del segundo término aparece como resultado $i|1\rangle$.

## 2.3 Teorema de no clonación

Una de las características más representativas de la computación cuántica viene dada por el teorema de no clonación. Este teorema expone que no existen dos partículas que puedan tener el mismo estado cuántico, y que por lo tanto, no se puede realizar una copia de un estado de una partícula a otra, siempre y cuando, sus estados sean arbitrarios, como se verá más adelante. Si esto se ve desde el punto de vista de la computación cuántica, lo que provoca es que el estado en el que se encuentra un qubit no pueda ser copiado en otro qubit, y que por lo tanto, la gran mayoría de algoritmos que se utilizan hoy en día para ordenadores convencionales que utilizan unidades de información en bits, dejen de tener sentido y sean irrealizables en ordenadores cuánticos.

Además, este teorema produce una fuerte restricción a la hora de elaborar cualquier algoritmo para un ordenador cuántico, ya que las comúnmente conocidas variables auxiliares que se utilizan en la programación convencional no pueden ser utilizadas, ya que no se puede realizar una copia sobre el valor de un qubit sin que este se vea alterado.

Como se ha dicho, este teorema se aplica para dos estados arbitrarios, ya que existe alguna excepción a este problema en el que sí se puede llegar a realizar una copia de un estado a otro, pero ambos estados (el que se desea copiar y el estado sobre el que va a ser copiado) deben ser ortogonales entre ellos.

Una posible demostración sencilla de este fenómeno se podría realizar con el siguiente ejemplo: se va a tratar de realizar la operación de clonación partiendo de un estado cuántico $|\Psi\rangle$ que quiere ser copiado en un qubit inicializado a 0

$$U_{clonar}(|\Psi 0\rangle) = |\Psi\Psi\rangle \tag{2.4}$$

Donde $U_{clonar}$ sería la operación (puerta cuántica) que se encargaría de realizar la clonación, y donde $|\Psi\rangle$ tiene la siguiente expresión

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2.5}$$

Si se desarrollan los términos de la ecuación 2.4, se obtiene lo siguiente

$$U_{clonar}(|\Psi 0\rangle) = \alpha|00\rangle + \beta|10\rangle \tag{2.6}$$

$$|\Psi\Psi\rangle = (\alpha|0\rangle + \beta|1\rangle) \oplus (\alpha|0\rangle + \beta|1\rangle) \tag{2.7}$$

Y desarrollando la operación se obtiene

$$|\Psi\Psi\rangle = \alpha^2|00\rangle + \alpha\beta|01\rangle + \alpha\beta|10\rangle + \beta^2|11\rangle \tag{2.8}$$

Por lo que si comparamos las expresiones de las ecuaciones 2.6 y 2.8, se observa que no son iguales excepto para el caso en el que alguno de los dos coeficientes valga 0 (y por lo tanto el otro sea 1), por lo que mediante este ejemplo se puede demostrar que para el caso general de dos qubits arbitrarios, la operación de clonación o copiado de dos estados, no es posible.

Esto tiene fuertes implicaciones y consecuencias además de las ya mencionadas anteriormente, este teorema provoca que no sea posible realizar la comunicación instantánea a través de los estados de dos partículas cuánticas entrelazadas y separadas, ya que como se explicó en el capítulo anterior, si fuese posible realizar múltiples copias de un estado cuántico, se podría llegar a conocer el estado cuántico de la otra partícula que estaría entrelazado con la que estamos midiendo. Con esto, se evita dicho problema conceptual en el que se transmitiría información de un lugar a otro sin importar la distancia que los separase, chocando de frente con la teoría de la relatividad de Einstein en la que se enuncia que nada puede viajar más rápido que la luz, incluida la información, ya que al fin y al cabo, la información no deja de ser un paquete de partículas con una estructura determinada que ha de ser transmitida y recibida para que se pueda interpretar dicha información.

## 2.4 Procesadores cuánticos

Los ordenadores cuánticos siguen en pleno desarrollo en la actualidad y todavía se desconoce el alcance que estos puedan llegar a tener el día de mañana, pero a pesar de ello, existen empresas que están invirtiendo fuertes cantidades en investigar en este campo debido al gran interés que suscita entre la opinión pública, así como las posibles implicaciones que podrían llegar a tener estos ordenadores en un futuro. Fruto de este desarrollo, han aparecido dos tipos de procesadores cuánticos QPU (Quantum Processor Unit) entre los que se encuentran repartidos algunas de las empresas más importantes del área tecnológica.

Para poder conocer el desarrollo actual de estas tecnologías por las distintas empresas, es necesario primer exponer brevemente en que se diferencian estas dos visiones que existen entorno a como llevar a cabo un ordenador cuántico, y como su estructura afecta a los resultados que se obtienen de los mismos. Estos procesadores se dividen entre procesadores cuánticos que se basan en una característica conocida como annealing cuántico, que se explicará a continuación, y el modelo universal de procesador cuántico que se basa en circuitos cuánticos y sus respectivas puertas lógicas cuánticas.

### 2.4.1 Quantum annealing

Antes de realizar la comparación entre ambos procesadores cuánticos, es necesario explicar un concepto muy importante, ya que el primer tipo de ordenadores cuánticos están basados en este tipo de algoritmos. Se está hablando del algoritmo conocido como simulated annealing (recocido simulado) [6,7], el cuál recibe su nombre del proceso utilizado para el enfriamiento de los metales. Este algoritmo meta-heurístico se utiliza en problemas de optimización en los que se intenta buscar el máximo global, y no uno local. Para ello, se evalúa un punto y sus alrededores, y mediante un análisis probabilístico, se decide si se realiza un cambio a un punto de menor energía, o se decide permanecer en el mismo lugar.

Este proceso es muy similar al utilizado en el ámbito de la metalurgia al aplicarse un recocido a un metal, ya que los átomos de los metales se redistribuyen para conseguir llegar al estado de mínima energía, y es fruto de esta redistribución entre el átomo en el que nos centramos, y los átomos de su entorno, de donde nace la idea de este algoritmo. Al igual que sucede con el proceso físico de enfriamiento, si este se realiza lentamente (sin que sea un enfriamiento rápido o forzado), el número de imperfecciones en el cristal disminuye, y esto también se puede ver en el algoritmo como el número de iteraciones que se deciden realizar, ya que cuanto mayor sea el número, mejor será la aproximación al mínimo global del problema.

Una vez visto este concepto, el quantum annealing o temple cuántico (recocido cuántico) se puede ver como una analogía al recocido simulado, pero donde en lugar de ser similar al proceso de enfriamiento y redistribución de átomos, ahora se realiza mediante el efecto túnel que se experimenta en la mecánica cuántica. Como ya se ha visto anteriormente, la mecánica cuántica es una teoría probabilística, por lo que las partículas llevan asociada una función de onda que no es más que una ecuación donde cada término lleva asociada una probabilidad de que un determinado estado sea el que aparezca como resultado cuando se mida dicha partícula y por lo tanto se haga colapsar su función de onda. Es por ello que aparece el efecto túnel, que no es más que la posibilidad de que la probabilidad de una determinada partícula a ser capaz de superar un muro energético sea posible. Esto se puede ver con un simple ejemplo, donde la magnitud cuántica será la posición de una determinada partícula que se encuentra en una caja. Según las leyes del mundo macroscópico, si está caja estuviese perfecta mente sellada, la partícula no podría llegar a salir, pero sin embargo, si se tiene en cuenta los fenómenos cuánticos, existe una cierta probabilidad asociada a la partícula que podría ser que se midiese dicha partícula fuera de la caja y no dentro, siendo capaz de atravesar una de las paredes de la caja y dar ese salto energético que en el mundo macroscópico sería imposible.

Este concepto es por tanto uno de los usados actualmente para realizar procesadores cuánticos, y aunque solamente destaca D-WAVE como la única empresa que está logrando buenos resultados, este tipo de ordenadores suelen ser criticados por científicos y expertos, ya que aunque en tienen una cantidad de qubits muy superior a los procesadores cuánticos universales, los resultados obtenidos en estos ordenadores cuánticos están de lo que se podría llegar a obtener con el otro tipo de ordenadores si estos aumentasen su número de qubits. Esto afecta a que se piense constantemente que no se está realizando un uso de las propiedades de la mecánica cuántica, sino que realmente se está utilizando un algoritmo de optimización en el que está involucrado un fenómeno cuántico como es el efecto túnel.

Sin embargo, los procesadores cuánticos universales están basados en las puertas lógicas cuánticas y los propios

circuitos cuánticos, los cuales se verán en el siguiente capítulo. Este tipo de ordenadores tienen una fuerte inversión por las empresas más importantes del ámbito tecnológico, y aunque en cuanto a capacidad de computación estén actualmente por detrás de los procesadores basados en el temple cuántico, se espera que estos sean el concepto de los futuros ordenadores cuánticos.

| Tipología | Empresa | Nombre QPU | Nº QUBITS |
|---|---|---|---|
| Quantum Annealing | D-WAVE | D-WAVE 200Q | 2048 qb |
| QPU universal | Google | Bristelcone | 72 qb |
| | IBM | IBM Q 50 prototype | 50 qb |
| | Intel | Tangle Lake | 49 qb |
| | Rigetti | 19Q Acorn | 19 qb |

Tabla 2-1. Principales QPUs según nº de qubits

Como se puede observar a partir de esta tabla, existe una gran diferencia entre los procesadores basados en el modelo universal, y los que están basados en el temple cuántico, pero a pesar de esta diferencia, otro dato muy importante es realizar una comparación entre este número de qubits y el número de bits de un procesador de hoy en día de un ordenador convencional. Esto permite ver lo joven que sigue siendo esta tecnología y lo mucho que todavía queda por investigar y mejorar en este ámbito. A pesar de ello, y aunque se hayan podido realizar ciertas pruebas y experimentos muy concretos, se ha podido observar el gran potencial que estos ordenadores pueden llegar a tener un futuro, y es por ello, que se sigue investigando para mejorar tanto el hardware y construcción de estos ordenadores, como del software; es decir, algoritmos que permitan dar solución a problemas que en la actualidad son muy costosos en tiempo de computación para un ordenador convencional.

Por ello, es necesario que se realicen pruebas en ordenadores convencionales pero que de alguna forma simulen o emulen el comportamiento de un ordenador cuántico, para que de esta forma se puedan ir probando algunos algoritmos y nuevas formas de pensar computacionalmente, para que de esta forma, cuando el hardware lo permita, existan numerosos algoritmos que justifiquen la fuerte inversión que se realiza en la mejora de estos procesadores cuánticos. A continuación, se van a exponer algunos de estos simuladores que existen, como también se expondrá aquellos que van a ser utilizados en este proyecto para implementar el algoritmo.

## 2.5 Simuladores cuánticos

Durante los años de desarrollo de la tecnología cuántica se ha ido desarrollando múltiples lenguajes de programación, así como conjunto de instrucciones y kits que permitan acercar el mundo de los ordenadores cuánticos a los programadores que están acostumbrados a los lenguajes que más se utilizan hoy en día. Por ello, se van a mostrar a continuación algunos de estos conjuntos de instrucciones que se han desarrollado, como lenguajes de programación completamente nuevos que se utilizan en ordenadores convencionales, pero que cuando llegue el momento en el que los ordenadores cuánticos sean algo común, estos lenguajes también puedan ser aplicados en ellos.

### 2.5.1 Quil

Quil es un conjunto de instrucciones que permite implementar múltiples algoritmos cuánticos basados en optimización, teleportación cuántica o corrección de error. Este es usado como base de lenguajes más complejos, ya que permite pasar de instrucción cuánticas a instrucciones de bajo nivel en un ordenador convencional. Además, este conjunto de instrucciones permite compartir memoria clásica y cuántica, lo cual es un importante factor a tener en cuenta, ya que existen operaciones que son más óptimas computacionalmente en un ordenador

clásico que en uno cuántico y viceversa. Este conjunto de instrucciones también serán la base del algoritmo utilizado en este proyecto, aunque este conjunto sea algo distinto, ya que el utilizado pertenece a la empresa Rigetti.

### 2.5.2 OpenQuasm

Al igual que Quil, OpenQASM es un conjunto de instrucciones, pero en esta ocasión ha sido desarrollado por la empresa IBM, y es utilizado junto con Qiskit, que como se verá más adelante, permite realizar simulaciones de circuitos cuánticos en Python y otros lenguajes de programación. Además, OpenQASM es también utilizado por el proyecto de IBM conocido como IBM Q Experience, en el cual se pueden realizar simulaciones online de circuitos cuánticos básicos utilizando las puertas cuánticas más conocidas, y además, se pueden realizar pruebas en un ordenador cuántico de 4 qubits, aunque están suelen llevar un elevado retraso de tiempo, ya que este es un único ordenador cuántico para todos aquellos que quieran realizar pruebas a través del servicio online. A continuación se muestra el circuito cuántico y el resultado de la simulación para una puerta Hadamard utilizando este servicio online.

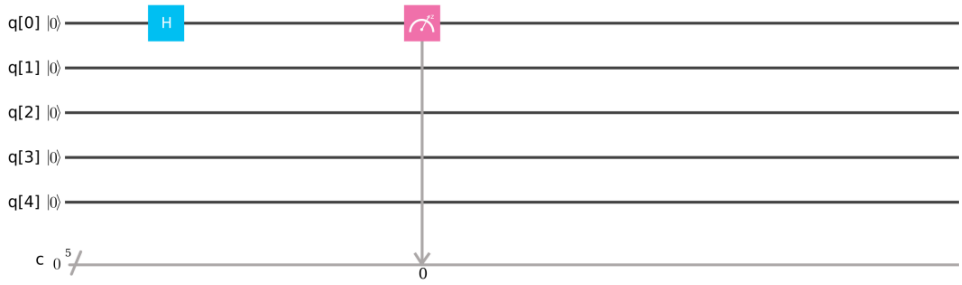


Figura 2-2. Circuito cuántico para puerta Hadamard

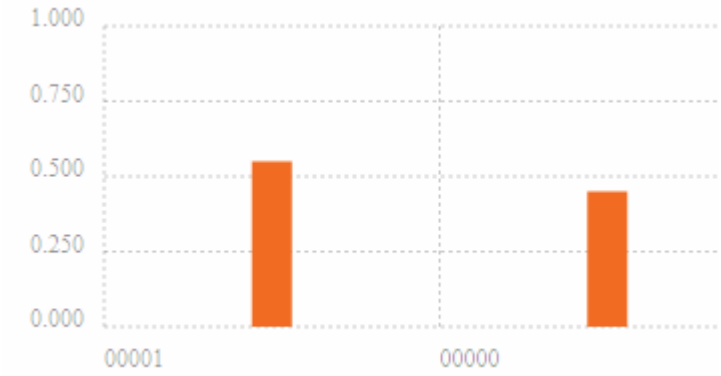


Figura 2-3. Resultado de la simulación de una puerta Hadamard

### 2.5.3 Qiskit

Como se ha mencionado, Qiskit es un proyecto desarrollado por IBM para plataformas como Python Swift o Java, y que es de código abierto. Este permite realizar programas en estos lenguajes en los que se utilicen puertas cuánticas y circuitos cuánticos, además de dar la posibilidad de realizar programación de alto nivel con otras ampliaciones. Este sistema es muy utilizado y existe una gran comunidad en su entorno que realizan implementaciones de algunos de los algoritmos cuánticos más conocidos e importantes hasta el momento, así como algoritmos que permiten realizar operaciones que se realizan actualmente en los ordenadores convencionales.

### 2.5.4 Forest

Forest es el análogo a Qiskit pero de la empresa Rigetti, y este está basado en Python únicamente. Al igual que Qiskit, permite implementar y manipular circuitos cuánticos, y los resultados de las simulaciones se obtienen a

través de simuladores de la propia empresa. Este será utilizado en este proyecto, ya que cuenta con una fuerte presencia en algoritmos cuánticos para problemas de optimización, que es el motivo de este trabajo.

### 2.5.5 Quantum Development Kit y Q#

Como su propio nombre indica, es un kit de desarrollo que permite escribir y simular programas cuánticos mediante Visual Studio y Visual Studio Code. Este ha sido desarrollado por la empresa Microsoft y en él se encuentra uno de los lenguajes de programación cuántico más importantes hasta el momento, Q#, el cual se puede utilizar desde VS y realizar una programación de circuitos cuánticos (bajo nivel), o incluso implementar algunos algoritmos de alto nivel.

### 2.5.6 QMASM

En último lugar, y como se ha hablado anteriormente de ordenadores cuánticos basados en el temple cuántico, QMASM es un lenguaje de bajo nivel utilizado específicamente en este tipo de ordenadores. Su nombre proviene de las siglas Quantum Macro Assembler y permite construir algoritmos a través de bloques de alto nivel que son ensamblados, permitiendo que el código no sea de bajo nivel y por lo tanto no sea necesario tener un alto conocimiento acerca del hardware del ordenador.

# 3 PUERTAS CUÁNTICAS

En siguiente lugar, se procede a exponer algunas de las puertas cuánticas más conocidas y describir su funcionamiento y aplicación en los algoritmos cuánticos. Estas reciben este nombre ya que su funcionamiento es similar al de las puertas lógicas convencionales como NOT o XOR, pero en esta ocasión, explotando todo el potencial que existe al utilizar qubits que permiten estar en múltiples estados en vez de bits, que solo pueden tomas los valores binarios 0 o 1.

## 3.1 Hadamard

Una de las puertas cuánticas más conocidas es la puerta Hadamard [8], la cual actúa sobre un único qubit devolviendo a partir del estado 0 o 1 un valor que tendrá la misma probabilidad de ser 0 o 1; es decir, permite distribuir la probabilidad de que el qubit esté en el estado 0 o 1, sin importar si su estado inicial era 0 o 1. Esto se puede entender como una rotación de un ángulo $\pi$ entorno al eje $\frac{x+z}{\sqrt{2}}$ de la esfera de Bloch expuesta anteriormente. La representación matricial de la puerta es la siguiente:

$$H = \frac{1}{\sqrt{2}} * \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{3.1}$$

Y los resultados de dicha operación son:

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{3.2}$$

$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{3.3}$$

Por último, se muestra en la figura 3-1 cuál es la simbología de la puerta Hadamard así como el resultado que se obtiene tanto para una entrada cuyo estado es 1 como 0, y otras dos en las que se observa como cuando la entrada es el resultado de las ecuaciones 3.2 y 3.3, el resultado de la puerta Hadamard es el de 0 o 1 respectivamente, y acompañado todos esos resultados de una representación del qubit en la esfera de Bloch.

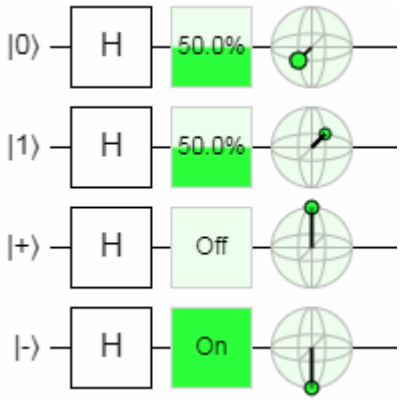


Figura 3-1. Casuística de la puerta Hadamard (H)

## 3.2 Pauli

Otras puertas cuánticas son las aplicaciones de las matrices de Pauli a la computación cuántica. Estas matrices reciben esta denominación en nombre del famoso físico Wolfgang Pauli, y las cuales permiten actuar sobre un solo qubit al igual que la puerta vista anteriormente, realizando 3 transformaciones o rotaciones distintas a través de cada uno de los ejes de la esfera de Bloch.

### 3.2.1 Pauli-X gate

Esta primera puerta cuántica se encarga de realizar un giro de 180º entorno al eje X. Su efecto es el mismo que provoca la puerta NOT convencional, es por ello, que también se le atribuya la nomenclatura de puerta NOT cuántica. La expresión de la matriz de Pauli es la siguiente:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{3.4}$$

Mientras que el resultado de la transformación para una función de onda tal que

$$|\Psi\rangle = a * |0\rangle + b * |1\rangle = \begin{pmatrix} a \\ b \end{pmatrix} \tag{3.5}$$

sería el siguiente:

$$X|\Psi\rangle = NOT|\Psi\rangle = b * |0\rangle + a * |1\rangle = \begin{pmatrix} b \\ a \end{pmatrix} \tag{3.6}$$

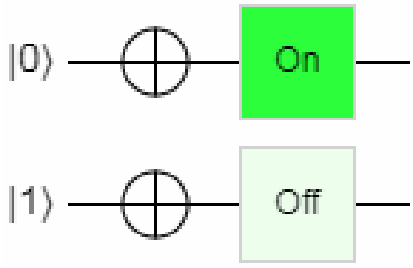


Figura 3-2. Puerta X de Pauli (NOT)

### 3.2.2 Pauli-Y gate

En segundo lugar, esta puerta realiza un giro de 180º entorno al eje Y, lo cual permite transformar el estado de un qubit de la siguiente forma:

$$Y|0\rangle = i|1\rangle \tag{3.7}$$

$$Y|1\rangle = -i|0\rangle \tag{3.8}$$

A diferencia de la puerta anterior y de la próxima, la puerta de rotación alrededor del eje Y no es tan utilizada en los algoritmos que existen en la actualidad, aunque debido al poco número de algoritmos que existen en la actualidad y su reducido campo de estudio, puede ser que en el futuro sea una de las más utilizadas. Su matriz de transformación correspondiente sería:

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{3.9}$$

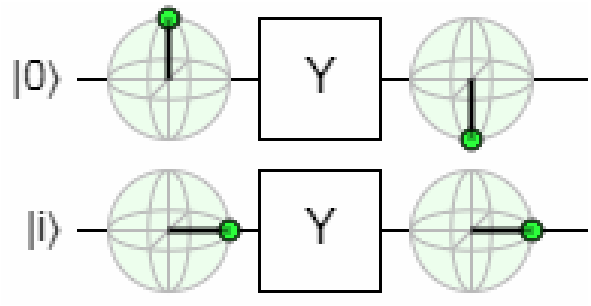


Figura 3-3. Puerta Y de Pauli

### 3.2.3 Pauli-Z gate

Por último, esta puerta cuántica permite realizar un giro de 180º entorno al eje Z, lo que significa que se permite realizar un desplazamiento en la fase del qubit. Es por esto, que esta puerta es un caso concreto de la puerta conocida como phase-flip o desplazamiento de fase que se expondrá a continuación, donde el ángulo girado es $\varphi = 180$. Su matriz por lo tanto, sería de la siguiente forma:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{3.10}$$

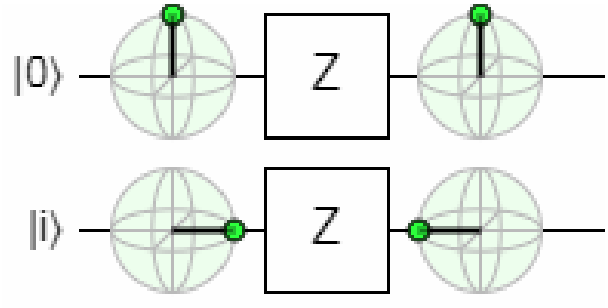


Figura 3-4. Puerta Z de Pauli

## 3.3 Desplazamiento de fase

Esta puerta también conocida como Phase-flip permite realizar un giro entorno al eje Z de φ radianes, lo que provoca que el estado $|0\rangle$ permanezca invariante, mientras en el estado $|1\rangle$ tome valores en función de la siguiente expresión:

$$R_\varphi|1\rangle = e^{i\varphi}|1\rangle \tag{3.11}$$

Y por lo tanto, las probabilidades de medir tanto un valor 0 como 1 permanecen inalteradas, puesto que lo que se está variando es la fase de la función de onda del qubit. La matriz de transformación dependerá por lo tanto también del ángulo de giro entorno al eje z.

$$R_\varphi = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix} \tag{3.12}$$

Hasta ahora, se han expuesto las puertas cuánticas más importantes que actúan sobre un único qubit cambiando su estado. A partir de ahora, se expondrán las más utilizadas en algoritmos cuánticos y que a su vez actúan sobre múltiples qubits, permitiendo realizar operaciones más complejas, y que por lo tanto, su propia estructura se complica tanto desde el punto de vista teórico como a la hora de realizar su implementación en un ordenador cuántico real.

## 3.4 SWAP

Esta puerta cuántica múltiple permite, como su propio nombre indica, intercambiar entre si dos qubits. Es muy utilizada en los algoritmos cuánticos, ya que según el teorema de no clonación visto anteriormente no se puede realizar una copia del estado de un qubit en otro distinto, por lo que intercambiar el estado entre dos de ellos, permite dar una solución a este problema. La matriz tiene la siguiente expresión:

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.13}$$

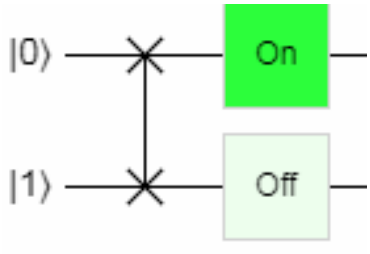


Figura 3-5. Puerta SWAP

## 3.5 CNOT

Otra puerta cuántica múltiple es la Controlled-NOT, la cual utiliza uno o varios qubits como control, mientras que realiza la operación de negación (NOT) vista anteriormente sobre el resto. Este tipo de operaciones se puede realizar con cualquier puerta que se diseñe; es decir, se puede utilizar uno o varios qubits que se encarguen de controlar que se realice o no una determinada operación, pero la más utilizada en la actualidad es la CNOT, y por ello, es la que se ha decidido explicar. Continuando con su descripción, y suponiendo que se utiliza un único qubit de control, se realizará la operación de negación solo cuando el estado del qubit sea 1, dando lugar a la siguiente expresión matricial:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.14}$$

Con esto se consigue que en caso de que el primer qubit tome el valor 0, el segundo qubit mantendrá su estado, mientras que si tiene un valor 1, se le aplicará al segundo qubit la operación de negación.

Para poder observar más claramente este efecto, se ha decidido realizar una simulación donde el primer qubit cuyo estado inicial es 0, pasa a través de una puerta Hadamard, con lo que se obtiene un 50% de que dicho qubit esté en el estado 0 y otro 50% de que esté en el estado 1. Tras esto, se realiza la operación de negación al segundo qubit, siendo controlada esta operación por el primer qubit, y por lo tanto, solo se realizará en el caso de que el primero tome valor 1. Por lo tanto, el resultado debe ser que ambos qubits tengan la misma probabilidad.

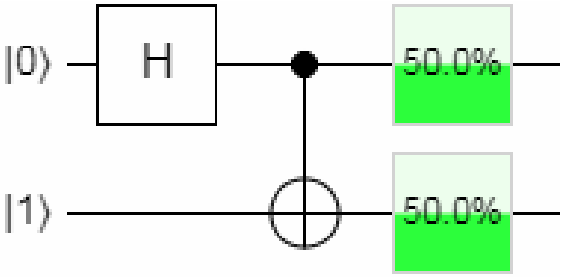


Figura 3-6. Circuito con una puerta Hadamard y una CNOT

# 4 PROBLEMA DE MÁXIMO CORTE

En los anteriores capítulos se ha realizado una breve introducción de los aspectos más importantes de la mecánica cuántica, así como de la computación cuántica para que se pueda comprender cuál es la base de los algoritmos cuánticos, y más concretamente, el algoritmo que se utilizará en este proyecto, aunque antes de exponer dicho algoritmo, es necesario dedicarle un capítulo a explicar cuál será el problema que se busca resolver mediante un computador cuántico, y la ventaja que esto tiene frente a un ordenador convencional

El problema es conocido con el nombre de problema de máximo corte, el cual es un problema combinatorio NP-completo [9-11]; es decir, un problema que no puede ser resuelto en tiempo polinomial mediante un ordenador clásico, y que por lo tanto, a mayor complejidad, el tiempo de computación necesario aumenta notablemente, lo que impide que cuando se tiene un problema de una complejidad considerable, sea imposible en la práctica calcular cuál sería el máximo corte de dicho problema con un ordenador clásico. Es aquí donde aparece la ventaja de este tipo de ordenadores, ya que aprovechándose de los fenómenos físicos de la mecánica cuántica que se han expuesto anteriormente, es posible reducir el tiempo de ejecución de un algoritmo que resuelva este problema para casos complejos.

El máximo corte de una determinada gráfica o red se basa en separar los nodos o vértices en dos conjuntos $S$ $y$ $\overline{S}$, de tal forma que se maximice el número vértices adyacentes que pertenecen a conjuntos distintos. Esta necesidad de realizar un análisis para cada nodo de la red hace que el problema aumente rápidamente en complejidad y tiempo de ejecución al mismo tiempo que se aumenta el número de nodos de la red, por lo que para un ordenador clásico en el que es necesario realizar todas las posibles combinaciones de que cada nodo se agrupe en un conjunto u otro, se convierte en una tarea muy complicada de resolver para estos ordenadores.

Por ello, existen múltiples algoritmos que buscan resolver este problema desde distintos puntos de vista, ya sea desde pruebas aleatorias con distintas combinaciones, los cuales darán resultados aleatorios, siendo imposible conocer si el resultado obtenido es el máximo del problema, o por el contrario, es un valor que esté alejado del máximo real del problema a resolver.

Este problema tiene dos versiones, la primera de ellas es la versión simple (o relajada) del problema, donde cada unión de los nodos (o la arista que une cada vértice) tiene el mismo valor, y por lo tanto, solo es necesario optimizar que el mayor número de estas uniones o aristas, estén uniendo nodos de conjuntos distintos, mientras que por otra parte, la otra versión del problema introduce distintos valores para cada unión (distintos pesos), por lo que a la hora de realizar la búsqueda del máximo hay que tener en cuenta estos pesos, dando lugar a un problema más complejo y por lo tanto, a unos tiempos de resolución mayores. Más adelante se realizará el desarrollo matemático del caso general (incluyendo los pesos), pero el algoritmo cuántico realizado en este proyecto está diseñado para resolver el caso más simple, donde todas las uniones de los nodos tienen los mismos valores (pesos).

Otro interesante aspecto acerca sobre este problema (la versión que incluye pesos para cada segmento) es que está recogido como uno de los 21 problemas NP-completos de Karp entre los que se encuentran otros problemas combinatorios famosos como el problema de la mochila. Este problema trata sobre la forma óptima de llenar una determinada mochila que solo puede cargar una cantidad limitada de peso, y unos objetos que tienen asociado un peso y un valor, buscando que el resultado final maximice el valor total de los objetos que se cargan en la mochila. Richard Karp agrupó varios problemas combinatorios que podían ser demostrados que eran NP-completos mediante la reducción de estos problemas al problema de satisfacibilidad booleana, el cual evalúa si una determinada ecuación booleana puede dar como solución el valor TRUE, y entonces se considera que la ecuación es satisfactoria, o por el contrario, si para cualquier valor que se le dé a las variables de la ecuación, el resultado siempre es FALSE, entonces la ecuación no será satisfactoria. Este problema fue el primer en ser probado como NP-completo (la clase más compleja de problemas NP).

## 4.1 Formulación del problema

Una vez expuesta una introducción sobre el problema que se va a tratar, se procede a explicar de forma más detallada cual es el fundamento matemático del problema de máximo corte, y posteriormente, se realizará un análisis gráfico acerca para mostrar el significado geométrico del problema sobre una red de puntos y sus uniones.

En primer lugar, se parte de una gráfica G constituida por un conjunto de vértices V y segmentos o uniones E, o lo que es lo mismo G= (V,E). A continuación se le asigna la variable x a cada vértice de la gráfica, tomando los siguientes valores:

$$x_i = \begin{cases} +1 & si\ el\ vértice\ pertenece\ a\ S \\ -1 & si\ el\ vértice\ pertenece\ a\ \overline{S} \end{cases} \tag{4.1}$$

Donde $S\ y\ \overline{S}$ son los dos conjuntos en los que se quiere separar los vértices de la gráfica, y siendo por tanto S el complementario de $\overline{S}$. Además, para hacer más completo el desarrollo matemático del problema, se tendrá en cuenta la posibilidad de utilizar unos pesos asignados a cada segmento (E), dando lugar a los siguientes valores:

$$w_{ij} \begin{cases} > 0 & si\ (i,j) \in E \\ 0 & si\ (i,j) \notin E \end{cases} \tag{4.2}$$

Es decir, si $w_{ij}$ es un segmento de la gráfica que une dos vértices, entonces tomará un valor positivo mayor que cero (para el caso del problema simple, $w_{ij} = 1$), mientras que si los vértices i y j no están unidos por un segmento, el valor de la variable $w_{ij}$ será 0, ya que estos dos vértices no están unidos por un segmento.

Con todo esto, la expresión del problema de optimización sería la siguiente

$$\begin{aligned} Max \quad & \frac{1}{2}\sum_{i<j}^{n} w_{ij} * \left(1 - x_i x_j\right) \\ s.a. \quad & x_i \in \{+1, -1\} \ \text{i = 1...n} \end{aligned} \tag{4.3}$$

Donde el producto $x_i x_j$ puede ser -1 si ambos vértices pertenecen a un conjunto distinto o +1 si ambos vértices pertenecen al mismo conjunto, por lo que el resultado de la multiplicación de los pesos por el segundo término será 2 en el caso de que dos vértices estén unidos por un segmento perteneciente a E y que estos vértices estén en conjuntos distintos, mientras que si esto no se cumple, la operación dará como resultado 0. Es por ello, que una vez que se realiza el sumatorio de todas las posibles combinaciones, el resultado obtenido será el doble del real, por lo que es necesario dividir entre 2. Además, para eliminar la restricción de que el sumatorio se restrinja a valores de i inferiores a j, se puede dividir de nuevo por 2 el resultado final, y se seguiría obteniendo el resultado esperado. Por lo que añadiendo esto último y variando ligeramente la restricción de $x_i$, se obtiene la siguiente expresión

$$Max \quad \frac{1}{4}\sum_{i,j}^{n} w_{ij} * (1 - x_i x_j) \qquad (4.4)$$
$$s.a. \quad x_i^2 = 1 \text{ i = 1...n}$$

Si esta última expresión se traduce a notación vectorial, se obtiene la siguiente expresión

$$Max \quad \frac{1}{4} X' * L * X \qquad (4.5)$$
$$s.a. \quad x_i^2 = 1 \text{ i = 1...n}$$

Donde X= ($x_i$) y L es la laplaciana que se define de la siguiente forma

$$L = diag(W * e) - W \qquad (4.6)$$

W es la matriz de pesos W= ($w_{ij}$), e es el vector de unos y 'diag' permite obtener una matriz donde todos sus elementos son nulos excepto la diagonal, que toma los valores del vector que se diagonaliza.

Por lo tanto, se extraen dos ecuaciones distintas, dependiendo de si se utiliza la notación vectorial o no (expresiones 4.5 y 4.4 respectivamente). En cambio, como se verá en el siguiente capítulo, cuando se traslada este problema hacia un algoritmo cuántico, es necesario realizar ciertas modificaciones a la expresión 4.4 para poder ajustarla a la notación cuántica, pero sin modificar su sentido matemático.

## 4.2 Descripción geométrica

Una vez descrita la formulación del problema del máximo corte, es necesario cual es el sentido geométrico de este problema, y puesto que se trata de un problema de gráficas, vértices y segmentos, esto permite tener un conocimiento más pleno y sencillo sobre el objetivo de este problema de optimización.

Como su propio nombre indica, el máximo corte se consigue una vez separados los vértices en los dos conjuntos como la línea que separa a los dos conjuntos de vértices y que corta a los segmentos, y que en este problema, se busca maximizar el número de segmentos que son cortados por esta línea

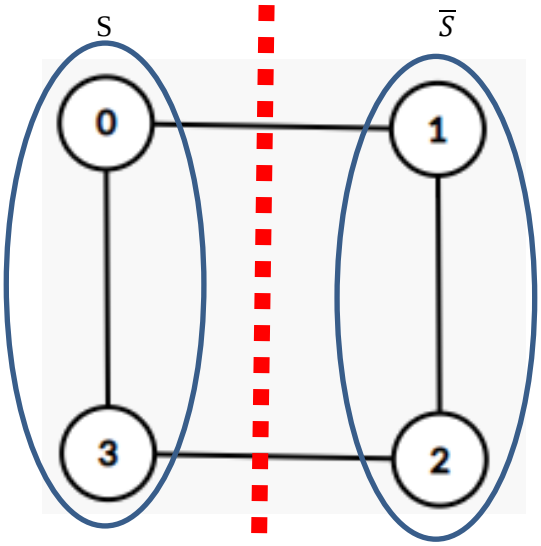


Figura 4-1. Ejemplo de máximo corte aplicado a un cuadrado

En la figura 4-1 se observa como los vértices 0 y 3 pertenecen al conjunto S, mientras que los vértices 1 y 2 al conjunto de $\overline{S}$, y al trazar la línea de corte (línea roja) se observa que corta a dos segmentos (el que unen 0 y 1; y la unión entre 2 y 3). Este resultado se puede mejorar, ya que para un caso tan simple, se puede observar rápidamente que la distribución adecuada de los vértices en los conjuntos es la de la figura 4-2.

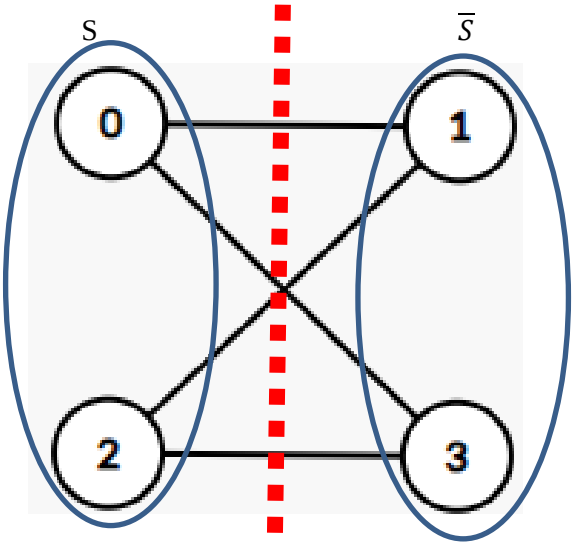


Figura 4-2. Ejemplo de máximo corte aplicado a un cuadrado (resultado óptimo)

Como se puede observar, si se redistribuyen de forma correcta los vértices, se puede obtener hasta un máximo de 4 cortes de segmentos, lo cual es el resultado buscado para este caso. A pesar de lo sencillo que pueda parecer, este no es más que un ejemplo en el que aparecen solamente 4 vértices y 4 segmentos que los unen entre ellos, pero conforme se va complicando la estructura de la gráfica, se vuelve más complicado resolver, y para casos de cientos de miles de nodos interconectados entre ellos, este problema se vuelve imposible de resolver incluso para un ordenador clásico, ya que la cantidad de combinaciones posibles se incrementa rápidamente conforme se aumenta el número de vértices y segmentos de la gráfica, y es por ello que existe interés acerca de la posibilidad de que los ordenadores cuánticos sean capaces de resolver este problema y muchos otros similar con un coste de tiempo de ejecución mucho menor.

A continuación, se muestra otro ejemplo algo más complejo en el que la solución deja de ser trivial y es necesario emplear un ordenador y un algoritmo para poder resolver este problema rápidamente. En esta figura (4-3) se observa como al aparecer 8 vértices y 12 segmentos, el problema para encontrar el valor óptimo deja de ser trivial, y tras realizar la simulación, el resultado de máximo corte de la figura es 7

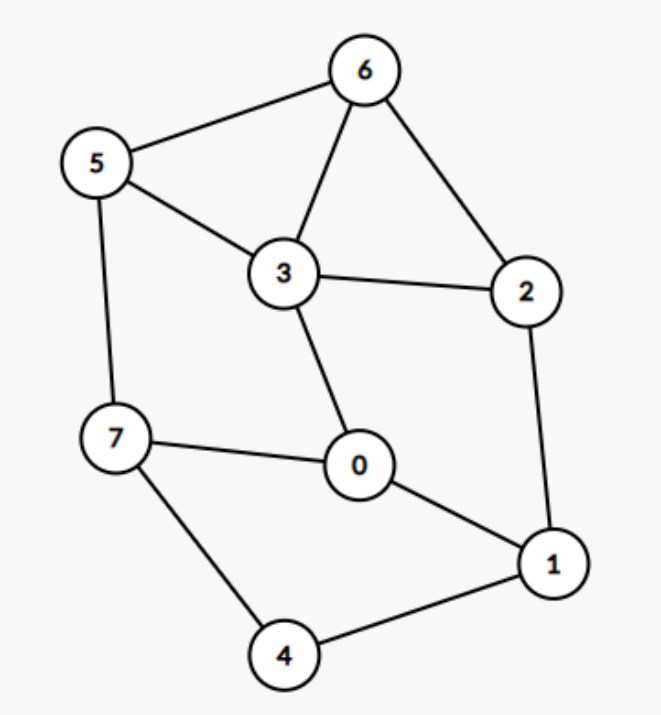


Figura 4-3. Ejemplo de máximo corte complejo

## 4.3 Métodos clásicos de solución

Como se ha podido observar, el problema del máximo corte tiene un gran interés desde el punto de vista geométrico, puesto que tiene aplicaciones en diversos temas como en el diseño de circuitos mediante la técnica conocida como "Very Large Scale Integration", o por sus siglas VLSI. Esta técnica se utiliza para crear circuitos integrados en el que se implementan millones de transistores tipo MOS. Por ello, se han realizado múltiples intentos por desarrollar un algoritmo que permita obtener unos resultados aproximados al valor máximo y en relativamente poco tiempo de ejecución.

Por ello, se van a exponer a continuación algunas de las técnicas más empleadas para resolver este problema, aunque todas ellas están orientadas a ser utilizadas en la computación clásica, mientras que el algoritmo desarrollado para resolver este problema en un ordenador cuántico y que es objeto de estudio de este trabajo será descrito en el siguiente capítulo. Las técnicas que se van a exponer parten desde conceptos completamente distintos para poder conseguir una solución al problema, ya sea una solución aproximada o una simplificación del problema.

### 4.3.1 Branch and Bound

El algoritmo conocido como ramificación y poda, o por su terminología en inglés Branch and Bound es una técnica empleada para resolver distintos problemas de optimización y mediante la cual se permite obtener resultados aproximados de la solución del problema. Este método se basa en distribuir el problema en distintas ramas de soluciones y realizar evaluaciones de cada una de estas ramas, y en caso de que la rama evaluada deje de ser óptima, se procede a "podar" dicha rama (eliminarla) y por lo tanto dejar de seguir realizando cálculos sobre una posible solución que ya se sabe que no puede ser la óptima.

Este algoritmo es muy utilizado en el problema de la mochila que se explicó anteriormente, ya que permite descartar soluciones cuando estás exceden el peso máximo de la mochila o incluso cuando la solución evaluada no puede superar a la óptima. En la figura 4-4 se puede ver un claro ejemplo de cómo funciona este algoritmo y las distintas ramificaciones en las que se dividen las posibles soluciones de tal forma que conforme se va descendiendo por una de las ramas, se puede ir tomando decisiones acerca de si esa rama es una posible solución, y por lo tanto se continúa con el cálculo, o cuando dicha rama deja de ser óptima por cualquiera de los motivos, momento en el que se desecha (se poda) dicha rama.

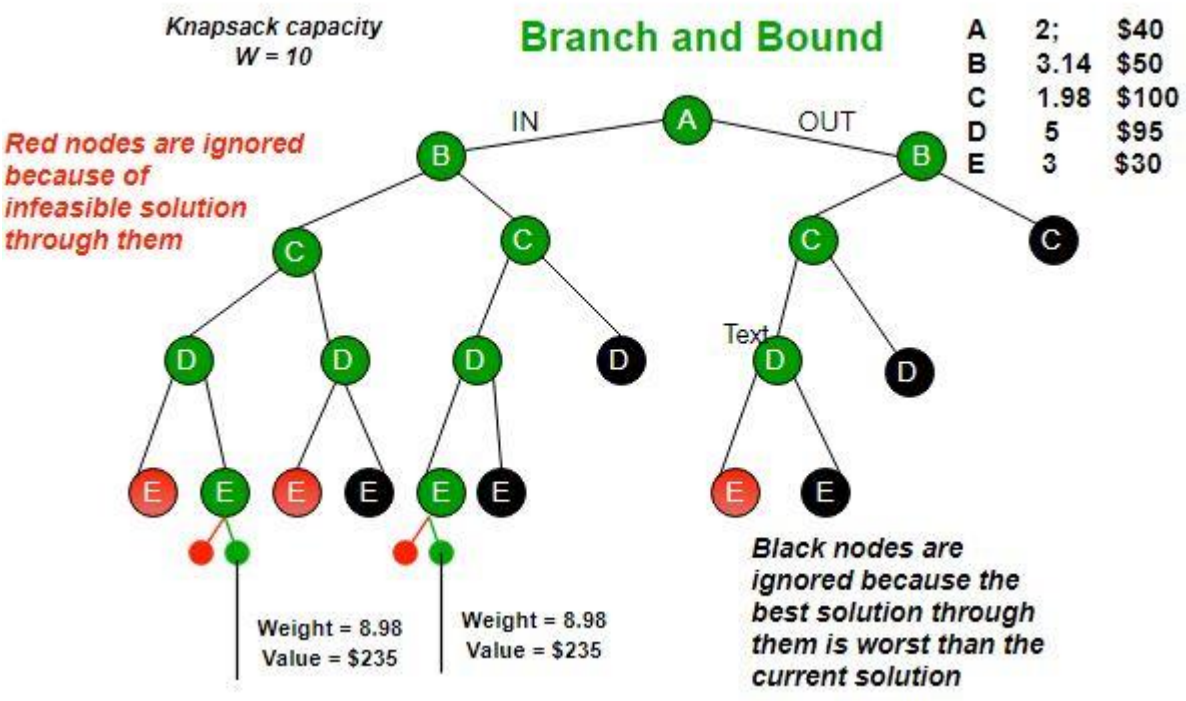


Figura 4-4. Branch and Bound aplicado al problema de la mochila

Una vez introducido el algoritmo, se puede realizar una aplicación al problema del máximo corte. Para comenzar, se parte de la expresión en notación vectorial 4.5 y se procede con el primer paso del algoritmo. Este se basa en ir descomponiendo cada par de vértices en dos problemas, que estén unidos (‘join’) o que estén separados (‘split’), por lo que el segmento que los une no será cortado en el primer caso, mientras que en el segundo sí. Estos dos subsistemas se expresan de la siguiente forma:

$$\begin{aligned} S_{join} &= \{x \in \{-1,1\}^n \;\; x_i - x_j = 0\} \\ S_{split} &= \{x \in \{-1,1\}^n \;\; x_i + x_j = 0\} \end{aligned} \tag{4.7}$$

Con esto se consigue que los pares de vértices que se encuentren en el mismo conjunto (S o $\overline{S}$), serán incluidos en el subproblema $S_{join}$, ya que al tener ambos vértices el mismo valor, si se restan entre ellos, el resultado será 0. Mientras que por otra parte, si pertenecen a distintos conjuntos, se incluirán en $S_{split}$, ya que al tener ambos vértices el mismo valor (1), pero distintos signo, la suma de ambos da como resultado 0.

Aplicando la expresión 4.7 a la expresión 4.5, se consigue obtener la expresión unificada del problema de máximo corte y el algoritmo de Branch and Bound.

$$\begin{aligned} Max \quad & \frac{1}{4} X' * L * X \\ s.a. \quad & x_i - x_j = 0 \;\; para \; (i,j) \in S_{join} \\ & x_i + x_j = 0 \;\; para \; (i,j) \in S_{split} \end{aligned} \tag{4.8}$$

### 4.3.2 Gráfica plana

Por otra parte, existe una simplificación del problema del máximo corte que permite ser resuelta en tiempo polinomial. Esta simplificación se produce cuando la gráfica G=(V,E) del problema es una gráfica plana [12]; es decir, las aristas o segmentos que unen los vértices pueden ser distribuidas de tal forma que no se corten entre ellas. Esto se puede ver más fácilmente en la comparativa de las imágenes de la figura 4-5.

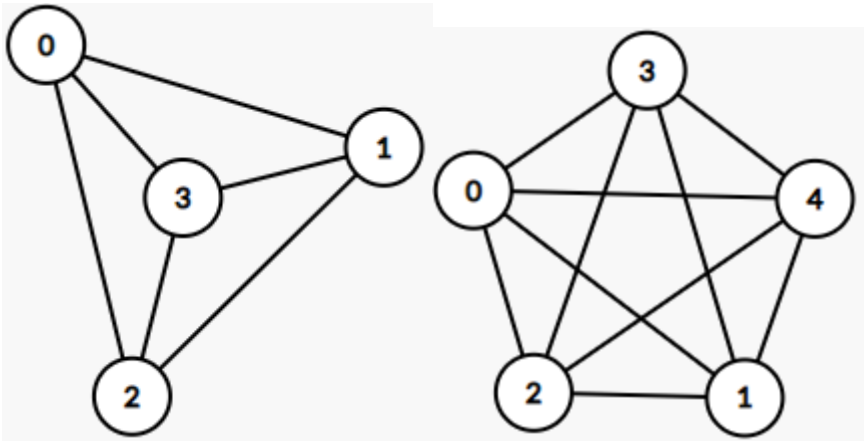


Figura 4-5. Comparativa de una figura plana (izquierda) con una no plana (derecha)

El concepto de gráfica o grafo plano se debe a que este tipo de imágenes pueden ser representadas en un solo plano sin que ninguna de sus aristas se corten, mientras que por el contrario, es necesario recurrir a la tercera dimensión para poder representar estas gráficas.

La aplicación al problema del máximo corte es muy importante, ya que para este tipo de gráficas, este problema es dual al problema del cartero chino, el cual se basa en intentar recorrer una ruta de la forma más rápida pasando por todos los caminos (aristas) de la ruta (gráfica). De esta forma, la solución óptima del problema será aquella que consiga recorrer todas las aristas de la gráfica pasando una única vez por cada una de ellas (siempre y cuando la gráfica lo permita).

El problema del cartero tiene solución y esta puede ser hallada en tiempo polinomial, por lo que la dualidad con el problema de máximo corte para el caso de que la gráfica sea plana, permite que este problema también pueda ser resuelto en tiempo polinomial, por lo que a través de un caso concreto o simplificación del problema del máximo corte, el cuál como se ha mencionado anteriormente es un problema NP (non polinomial time), se puede obtener una solución en tiempo polinomial.

### 4.3.3 Goemans-Williamson

Por último, el algoritmo de Goemans-Williamson es un algoritmo desarrollado específicamente para resolver el máximo corte, a diferencia del algoritmo Branch and Bound que puede ser utilizado para múltiples problemas combinatorios con resultados satisfactorios. Este algoritmo se basa en un proceso de redondeo para conseguir obtener una solución aproximada.

Este algoritmo está basado en la técnica conocida como programación semidefinida (semidefinite programming, SDP) y está demostrado que consigue obtener unos resultados que se aproximan a 0.878 el valor óptimo, aunque es posible superar este límite dependiendo del problema (propiedades del grafo, vértices y segmentos) que se analice.

Para resolver el problema, en primer lugar es necesario realizar una simplificación de la ecuación 4.5 y que permite obtener un límite superior del problema de máximo corte utilizando la técnica SDP mencionada anteriormente y consiguiendo este resultado en tiempo polinomial.

A partir de este modelo simplificado y su límite superior, se procede a realizar la factorización de Cholesky de la matriz X=x*x’ y se realiza posteriormente el procedimiento de redondeo en el que se van clasificando cada uno de los vértices del problema en cada uno de los conjuntos (S o $\overline{S}$) dependiendo de que cumplan unas condiciones determinadas por el algoritmo. Este procedimiento se repite para todos los vértices del problema,

ya que estos se van clasificando uno por uno en el conjunto dependiendo de lo descrito anteriormente.

# 5 QAOA

En los anteriores capítulos se ha ido realizando un recorrido desde los conceptos de la mecánica cuántica que son utilizados en la computación, como la propia computación cuántica y sus propiedades, así como de las puertas lógicas en las que se sustenta. A continuación se ha descrito el problema que se va a tratar de resolver, el problema de máximo corte, un problema que es del tipo NP-hard; es decir, la clase más compleja de los problemas NP, y a su vez, estos son los problemas que no pueden ser resueltos por un ordenador en tiempo polinomial, a diferencia de los problemas de tipo P, que sí lo son.

Por ello, es que se recurre a un nuevo tipo de estrategia para resolver estos problemas, ya que hasta la actualidad se han desarrollado múltiples estrategias para poder conseguir un algoritmo que consiga un resultado cercano al óptimo del problema, pero que a su vez se pueda hallar en tiempo polinomial. Es aquí donde aparece un nuevo concepto, la computación cuántica con sus ventajas e inconvenientes que ya han sido descritos anteriormente.

El mundo de la computación cuántica ha comenzado a tomar el interés de la comunidad científica debido a la aparición de algoritmos como el de Peter Shor, ya que permite descomponer un número en sus factores primos de forma mucho más eficiente que los algoritmos clásicos. Este problema tiene una gran transcendencia, ya que en la actualidad muchas criptografías se basan en utilizar los factores de un número muy grande y su descomposición en factores para poder realizar la comunicación entre dos puntos, y puesto que para un ordenador es imposible descifrar esta clave debido a que no es posible mediante algoritmos clásicos obtener una solución al problema en tiempo polinomial, esta clave se puede considerar seguro. En cambio, con el algoritmo de Shor, esto deja de ser real y la clave puede romperse y por lo tanto ser descifrado el mensaje por un intruso en la comunicación en tiempo polinomial. A esto se le conoce como supremacía cuántica.

## 5.1 Introducción

La supremacía cuántica es un concepto que se está extendiendo poco a poco entre los científicos, ya que con esto se busca explicar el potencial que tiene la computación cuántica de superar a la clásica. En la actualidad, esta frontera está lejos de superarse, ya que los superordenadores siguen teniendo una capacidad de cálculo superior a la del mejor ordenador cuántico, que como ya se vio, es el ordenador desarrollado por Google que alcanza los 72 qubits, lo que permite hasta $2^{72}$ estados, que aunque es una gran capacidad, muchos de ellos son necesarios para resolver problemas como la corrección de errores (muy necesario en los algoritmos cuánticos), así como en fenómenos como la decoherencia.

A pesar de ello, se ha seguido buscando posibles algoritmos y aplicaciones en las que los ordenadores cuánticos puedan tener un fuerte peso, y es por ello que surge el algoritmo conocido como Quantum Approximate Optimization Algorithm o pos sus siglas QAOA. Este ha sido desarrollado por Edward Farhi and Jeffrey Goldstone [13-15] como una posible estrategia para poder resolver problemas de tipo NP-hard de forma aproximada mucho más eficientemente que un ordenador clásico junto con el correspondiente algoritmo. QAOA ha despertado el interés de gran parte de los expertos en este campo, ya que se ha podido demostrar que el algoritmo no puede ser simulado eficientemente en un ordenador clásico, ya que si esto fuese posible, entonces se podría afirmar que P=NP, o lo que es lo mismo, que los problemas que en la actualidad son considerados NP, y que por lo tanto no pueden ser resueltos en tiempo polinomial, dejarían de serlo y podrían ser resueltos en tiempo polinomial. Esto le da a QAOA la posibilidad de ser utilizado para poder llegar a demostrar la tan ansiada por muchos científicos supremacía cuántica.

Además, otra ventaja del algoritmo frente a otras estrategias desarrolladas para resolver este tipo de problemas en ordenadores cuánticos es que no necesita invertir grandes recursos computacionales en corrección de errores, mientras que otras estrategias para solucionar estos problemas como quantum adiabatic son más críticas en este concepto. Esto permite que QAOA haya tenido un reconocimiento mayor en estos momentos en los que los ordenadores cuánticos están desarrollándose y el problema de la corrección de errores es uno de los aspectos más importantes a resolver por ellos.

Por otra parte, también se ha conseguido demostrar que QAOA y QAA (quantum adiabatic algorithm) son

equivalentes cuando un parámetro del algoritmo QAOA se hace infinito, aunque este concepto se desarrollará más adelante.

Otro aspecto a resaltar es que el artículo de Edward Farhi y Jeffrey Goldstone se publicó en 2014 [13], pero a pesar de ser un artículo bastante reciente, se han ido realizando nuevos estudios acerca del mismo que han conseguido que sea uno de los algoritmos cuánticos más conocidos para resolver problemas combinacionales. Gracias a esto, Zhang Jiang, G. Rieffel y Zhihnui Wang aplicaron el algoritmo al problema de realizar una búsqueda en una secuencia de datos que no ha sido ordenada. Este problema ya ha sido tratado eficientemente con un algoritmo cuántico, y recibe el nombre de su inventor, Grover. Este algoritmo es junto al algoritmo de Peter Shor, uno de los más conocidos y por el que se ha alzado el interés acerca de la computación cuántica, ya que en el caso concreto del algoritmo de Grover, es posible realizar la búsqueda en un tiempo $\sqrt{n}$ donde n es el número de datos, mientras que en un algoritmo clásico, es necesario un tiempo n; es decir, recorrer todos los datos hasta que se encuentre el buscado, mientras que el algoritmo de Grover solo necesita la raíz cuadrada de la cantidad de datos de la secuencia.

Es aquí donde la aplicación del algoritmo QAOA realizada por los expertos mencionados anteriormente ha conseguido obtener un resultado cuyo tiempo de computación es solamente $\sqrt{2}$ veces superior al algoritmo de Grover, lo que sigue siendo mejor que los algoritmos implementados en computadores clásicos.

## 5.2 Adaptación del máximo corte

Una vez realizada la introducción al algoritmo y cuáles son sus precedentes y sus objetivos, se procede a explicar el razonamiento matemático utilizado en el algoritmo, pero antes es necesario realizar un breve repaso de algunos de los conceptos explicados anteriormente y que van a ser utilizados en el desarrollo.

En primer lugar, las matrices de Pauli vistas en el capítulo 3 son utilizadas para poder reconvertir la ecuación de optimización del problema del máximo corte al ámbito de la computación cuántica. A continuación se muestran las 3 matrices asignada cada una de ellas a un eje del espacio tridimensional.

$$\sigma^x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$
$$\sigma^y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad (5.1)$$
$$\sigma^z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Aunque como se va a mostrar, solo se utiliza la matriz del eje Z, ya que como se vio en el capítulo 2, la esfera de Bloch es la representación geométrica de los posibles estados de un qubit, pero como en este caso solo nos interesa medir si el qubit se encuentra en el estado 0 o 1, solo se utiliza el eje Z, que es el que contiene estos dos estados, donde la parte positiva del eje se corresponde con el 0 y la negativa con el 1, como se puede ver en la figura 2-1.

En siguiente lugar, es necesario adaptar la ecuación del problema de máximo corte a una expresión cuántica, por lo que a parir de la ecuación 4.3 se obtiene que

$$Max\ C = \sum_{(j,k)\in \mathrm{E}}^{n} C(j,k) \quad (5.2)$$

Donde j y k son dos vértices que están unidos por un segmento perteneciente a E, el cual es el conjunto de todos los segmentos de la gráfica G, como ya se vio en el capítulo anterior.

$$C(j,k) = \frac{1}{2}(1 - \sigma_j^z \sigma_k^z) \tag{5.3}$$

Pero en esta ocasión, $\sigma_j^z$ y $\sigma_k^z$ no son las matrices de Pauli, si no que son la medida del vértice j o k en el eje Z de la esfera de Bloch; es decir, es la proyección del estado en el que se encuentra el qubit del vértice j o k, y donde se guarda la información de a qué conjunto pertenece el vértice, ya sea S o $\overline{S}$.

Por lo tanto, con esta nueva expresión para el problema de máximo corte, lo que se está buscando es que el resultado de la multiplicación de $\sigma_j^z \sigma_k^z$ sea siempre -1, o lo que es lo mismo, que el estado del qubit del vértice j y del vértice k cuando es medido en la base Z, tomen distintos valores, y que por lo tanto, ambos pertenezcan a conjuntos distintos, ya sea S o su complementario.

## 5.3 Fundamento teórico

Tras realizar el tratamiento necesario para adaptar la ecuación del problema de máximo corte a una expresión cuántica, ya se puede comenzar a desarrollar cual es el fundamento teórico del problema, para posteriormente poder realizar su desarrollo matemático.

QAOA es un algoritmo aproximado en el que existe un entero p que debe tomar valores mayores o igual que 1, y el cual sirve para conseguir un resultado de mayor calidad, o lo que es lo mismo, más aproximado al valor óptimo. Este valor p se podría traducir como el número de pasos que ha de repetir el algoritmo para ir refinando su resultado final. Esto es similar a lo que sucede con el machining learning, donde cuanto mayor son el número de cálculos de prueba, más se va refinando el algoritmo hasta conseguir el resultado esperado.

Sin embargo, como es de esperar, aumentar este entero provoca un considerable aumento del coste computacional, y por lo tanto, el tiempo que es necesario invertir para resolver el problema aumenta. Es por ello, que no siempre es preferible utilizar el valor más alto, sino que para problemas sencillos, con valores muy bajos de este parámetro, se pueden conseguir resultados que maximicen el problema de máximo corte, mientras que conforme se va complicando (más vértices y enlaces entre ellos), mayor va siendo el número necesario de pasos para poder distinguir una solución óptima, ya que como se verá en el capítulo de resultados, si se utilizan valores demasiado bajos para un problema relativamente complejo, los resultados que arroja el algoritmo son inútiles, ya que otorga probabilidades similares a muchos estados, por lo que es imposible conocer cuál es el resultado óptimo, ya que se obtienen un conjunto de posibles resultados entre los que no se puede diferenciar cual es el óptimo.

Además este entero p puede ser calculado mediante técnicas clásicas de forma que sea el óptimo, siempre y cuando este sea independiente del tamaño de la gráfica que se quiere resolver, mientras que si no es independiente, también se pueden utilizar otro tipo de estrategias para resolverlo.

Otra forma de entender el entero p, se puede ver como su aplicación geométrica, ya que este parámetro restringe el número de segmentos vecinos que se tienen en cuenta para el cálculo del máximo corte del segmento en el que estamos centrados; es decir, si se está calculando el máximo corte para la unión entre los vértices 0 y 1, se tendrán en cuenta todos los segmentos adyacentes al mismo que se encuentren separados una distancia ‘p’ de dicha unión, donde la distancia se contaría como el número de uniones que se han recorrido.

Para poder entender mejor este concepto, en la figura 5-1 se realiza un ejemplo donde p=2 y el segmento que se está estudiando es la unión de los vértices 0 y 1. Como se puede observar, la circunferencia trazada de radio 2 incluye todos los segmentos excepto a los que se unen con los vértices 10 y 11, ya que estos se encuentran a una distancia mayor que dos.

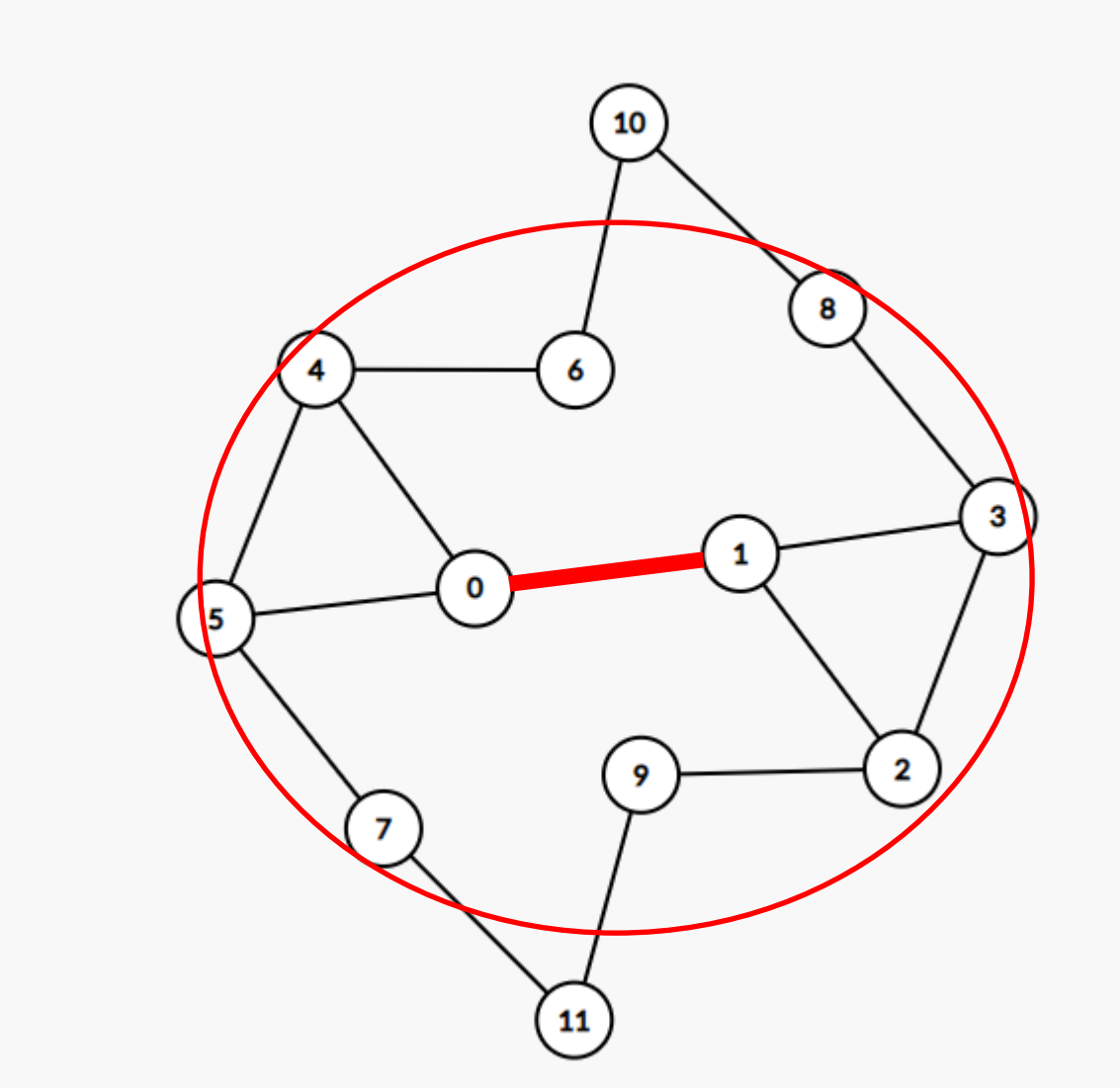


Figura 5-1. Representación del estudio del máximo corte de un eje para p=2

En siguiente lugar, es necesario el cálculo de dos ángulos, β y γ, los cuales a su vez se utilizan para dos operadores unitarios que se le aplican al estado cuántico. En primer lugar, γ varía entre 0 y 2π, mientras que β varía entre 0 y π.

Para p>1, no se calculan dos ángulos, si no que se repite el cálculo de los ángulos anteriores tantas veces como número de pasos hayan; es decir, existirán siempre 2p ángulos que se dividirán entre β y γ.

$$(\beta_p , \gamma_p) \dots (\beta_1 , \gamma_1)$$

Y por último, los operadores unitarios son dos, el primer de ello U(B, β) depende del ángulo β y del operador B que se mostrará más adelante cuál es su expresión para calcularlo. Y por otra parte está el operador unitario U(C, γ), que a su vez depende del ángulo γ y el operador C que ya es conocida su expresión, ya que aparece en la ecuación 5.2

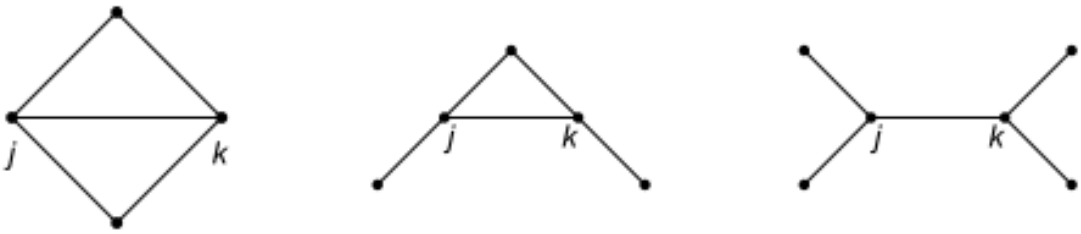


Figura 5-2. Gráficas posibles para p=1 y 3 segmentos

Para finalizar, se muestra en la figura 5-2 un caso concreto de posible gráfica, y se corresponde con 3 gráficas que tienen las mismas características, p=1 entorno a los vértices j y k, y de cada uno de estos vértices se derivan tres uniones. Este caso concreto es estudiado en el artículo original de Edward Farhi y Jeffrey Goldstone y se demuestra que se obtiene un valor aproximado de 0.6924 el valor óptimo, lo cual es mejor que un posible algoritmo que escogiese un corte aleatoriamente, ya que este obtendría un valor de aproximación de $\frac{2}{3}$.

Aunque estos resultados no son especialmente buenos, sí que permiten pensar si para casos más complejos también es posible obtener resultados superiores que un algoritmo que realice pruebas aleatoriamente, como algunos de los que existen en la actualidad que se basan en la búsqueda aleatoria de posibles resultados óptimos.

Además, tras este primer artículo se han realizado estudios por expertos y han demostrado que existen algoritmos clásicos que son capaces de superar los resultados ofrecidos por QAOA para el problema de máximo corte, pero que en cambio, QAOA está pensado como una clase de algoritmo que pueda ser empleado para múltiples problemas combinacionales, mientras que dichos algoritmos clásicos solo están pensados para resolver de forma óptima el problema del máximo corte.

## 5.4 Desarrollo matemático

La base teórica del algoritmo ya ha sido expuesta, e incluso se han introducido algunas conclusiones acerca de la efectividad del algoritmo frente a casos concretos, pero una vez visto esto, es necesario adentrarse en las ecuaciones que rigen este algoritmo y que como todo algoritmo cuántico, tiene una fuerte dependencia de las propiedades cuánticas, lo que le permite obtener estos resultados esperanzadores de cara a un futuro en el que se hayan podido desarrollar los computadores cuánticos y este tipo de algoritmos puedan ser implementados en ellos.

En primer lugar es necesario mostrar cuales son las expresiones para calcular los operadores B y C que se han expuesto anteriormente. C ya fue mostrado en las ecuaciones 5.2 y 5.3, y que puede ser expresado en una sola ecuación como

$$C = \frac{1}{2} \sum_{(j,k)\in \mathrm{E}} (1 - \sigma_j^z \sigma_k^z) \tag{5.4}$$

Donde (j,k) es la unión o segmente entre los vértices j y k, E es el conjunto de todos los segmentos de G, y por último, $\sigma_j^z$ y $\sigma_k^z$ son los valores de los qubits de los vértices j y k cuando es medido su estado en la base Z. Y como ya se ha mencionado anteriormente, lo que se busca es que el producto de dichos qubits sea -1 (uno toma el valor 1 y el otro -1), o lo que es lo mismo, que ambos vértices se encuentren en conjuntos distintos.

Además, si se observa los posibles resultados de C, este será 1 si el vértice j y el vértice k pertenecen a conjuntos distintos, y 0 si se encuentran en el mismo. Esto se similar a la operación que realiza la puerta XOR, por lo que se puede sustituir la fórmula anterior por la siguiente:

$$C = \begin{cases} 1 & si\ j \neq k \\ 0 & si\ j = k \end{cases} \tag{5.5}$$

$$C = \sum_{(j,k)\in \mathrm{E}} \sigma_j^z \oplus \sigma_k^z \tag{5.6}$$

Por otra parte, el operador B tiene la expresión siguiente

$$B = \sum_{j=1}^{n} \sigma_j^x \tag{5.7}$$

Donde n es el número de vértices (V) de la gráfica G y $\sigma_j^x$ es el valor del qubit perteneciente al vértice j en la base X.

Como se puede observar el operador B depende de los valores de cada vértice de la gráfica a resolver, mientras que C depende de los segmentos o uniones de los pares de vértices. Por ello, a B se le conoce como el operador de vértices mientras que C como el operador de los segmentos.

Una vez vistos los operadores B y C, se continúa con los operadores unitarios que también han sido descritos, y los cuales dependen de B, C y los 2p ángulos del algoritmo.

$$U(B,\beta) = e^{-i\beta \mathrm{B}} = \prod_{j=1}^{n} e^{-i\beta\sigma_j^x} \tag{5.8}$$

$$U(C,\gamma) = e^{-i\gamma \mathrm{C}} = \prod_{(j,k)\in \mathrm{E}} e^{-i\gamma(\sigma_j^z \oplus \sigma_k^z)} \tag{5.9}$$

A su vez, estos operadores unitarios son utilizados para obtener el estado cuántico dependiente de los ángulos, en el cual se busca multiplicar el estado inicial en el que se encuentra la superposición de todas las posibles soluciones al problema por estos operadores unitarios. Pero como se ha mencionado, no solo existen 2 operadores, si no que existirán 1 operador unitario para cada ángulo, por lo que serán también 2p operadores unitarios, dando lugar a la siguiente expresión

$$|\gamma,\beta\rangle = U(B,\beta_p)U(C,\gamma_p)\ldots U(B,\beta_1)U(C,\gamma_1)|s\rangle \tag{5.10}$$

Donde $|s\rangle$ es la superposición uniforme de los distintos posibles estados en los que se pueda encontrar los vértices de la gráfica. Esto se puede ver gráficamente en la figura 5-3. Por otra parte, $|\gamma,\beta\rangle$ es el resultado de la multiplicación de los operadores unitarios por el estado inicial, y se le conoce como el estado cuántico dependiente de los angulos, ya que es el paso intermedio antes de obtener la función a optimizar.

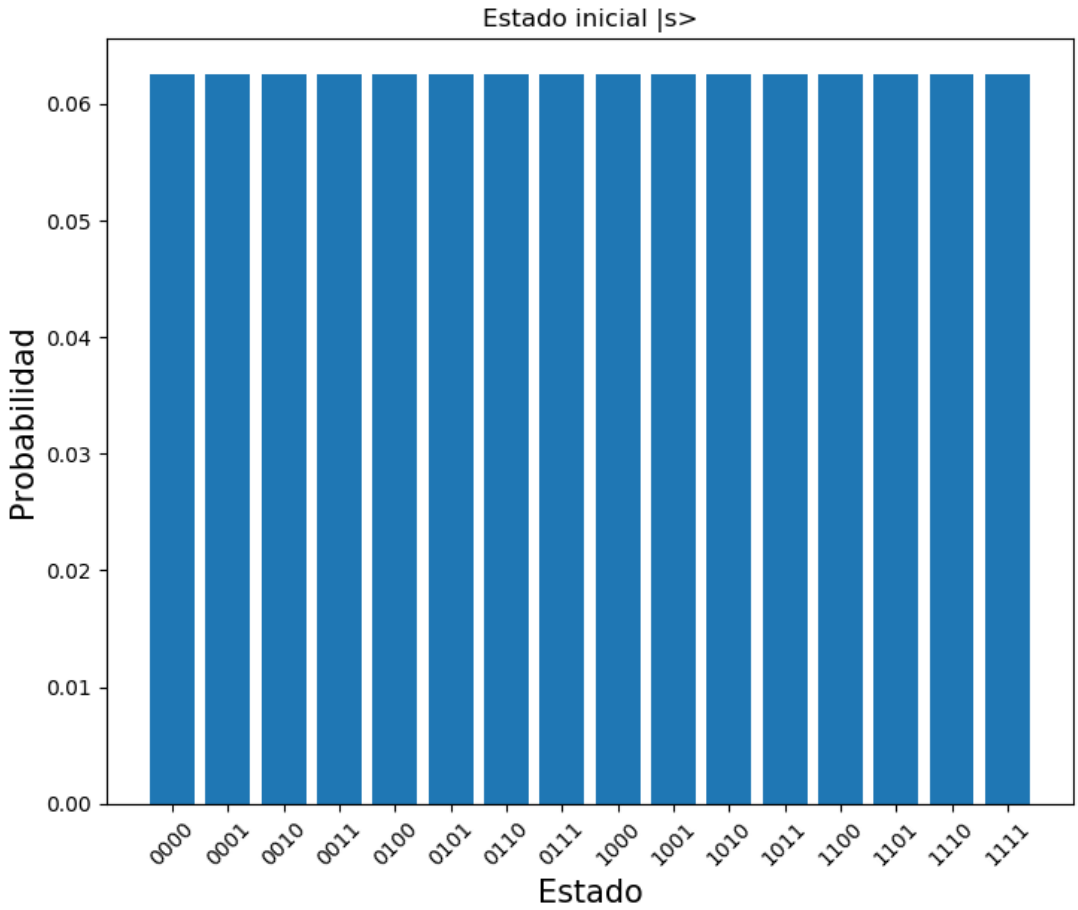


Figura 5-3. Estado inicial de superposición uniforme

Como se puede observar, el estado inicial parte de una superposición uniforme en la base computacional, donde esta base se corresponde a una gráfica de 4 vértices, y por lo tanto, 4 qubits que toman valores entre 0 y 1.

La expresión matemática para obtener el estado inicial en el que todos los valores de la base computacional son uniformes es la siguiente:

$$|s\rangle = \frac{1}{\sqrt{2^n}} \sum_z |z\rangle \tag{5.11}$$

Donde z son todos los valores de la base computacional, y por lo tanto el sumatorio de todos los valores se podría desarrollar de la siguiente forma

$$|000\rangle + |0001\rangle + \cdots + |1111\rangle$$

Mientras que el término $\frac{1}{\sqrt{2^n}}$ es el coeficiente de la función de onda, y que como ya se vió en capítulos anteriores, el cuadrado del coeficiente es la la probabilidad de que un determinado estado de la superposición sea medido, y por lo tanto para el caso de n=4, que es en el que nos encontramos se obtiene que

$$\frac{1}{\sqrt{2^4}}^2 = \frac{1}{2^4} = 0.0625$$

Y este valor es el que se puede apreciar en la figura 5-3 para todos los posibles estados que se encuentran en superposición. Y como no podría ser de otra forma, si se realiza la suma de todas las probabilidades, el resultado es 1 si se realiza en tanto por 1 o del 100%.

Una vez se obtiene el estado cuántico dependiente de los ángulos a partir del estado inicial y los operadores

unitarios (ecuación 5.10), se define la función que se quiere maximizar y que resulta de aplicar dicho estado cuántico a la variable que se quería maximizar en un principio C, dando lugar a la siguiente expresión

$$F_p(\gamma, \beta) = \langle \gamma, \beta | C | \gamma, \beta \rangle \tag{5.12}$$

Desarrollando dicha expresión se obtiene

$$F_p(\gamma, \beta) = \sum_{\langle j,k \rangle} \left\langle s \middle| U'\left(C, \gamma_p\right) \dots U'\left(B, \beta_1\right) C_{\langle j,k \rangle} U\left(B, \beta_p\right) \dots U\left(C, \gamma_1\right) \middle| s \right\rangle \tag{5.13}$$

Donde $F_p(\gamma, \beta)$ es la función que contiene los valores de todos los posibles estados y sus probabilidades en función de los ángulos calculados previamente y de la p (pasos) escogidos previamente.

A partir de esta ecuación también se desprende el valor del operador para un solo segmento $\langle j, k \rangle$

$$U'\left(C, \gamma_p\right) \dots U'(B, \beta_1) C_{\langle j,k \rangle} U\left(B, \beta_p\right) \dots U(C, \gamma_1)$$

Mientras que el valor máximo se obtiene escogiendo el par de ángulos γ y β para el que se maximiza $F_p$, y para este valor se utiliza la variable $M_p$ cuya expresión sería la siguiente

$$M_p = \max_{\gamma,\beta} F_p(\gamma, \beta) \tag{5.14}$$

Y teniendo en cuenta que conforme p aumenta, el resultado de la optimización mejora, se puede realizar la siguiente restricción

$$M_p \geq M_{p-1} \tag{5.15}$$

Por lo que realizando este mismo pensamiento, a mayor sea p, más cercano será el resultado obtenido al óptimo. De hecho, en el artículo original se realiza la demostración para la cual se cumple que

$$\lim_{p \to \infty} M_p = \max_{z} C(z) \tag{5.16}$$

Por lo que cuando se realiza el límite para p tendiendo a infinito, se obtiene que $M_p$ toma el valor máximo posible para el problema del máximo corte que se está intentando resolver. Aunque esto lleva un coste computacional que lo vuelve imposible en la práctica, por ello, el algoritmo se restringe a realizar los cálculos para conseguir un resultado cercano al valor óptimo (o máximo).

## 5.5 Implementación

Una vez expuesta toda la explicación acerca del algoritmo y su significado matemático, se va a proceder a explicar cuál es el procedimiento a seguir para poder realizar un algoritmo basado en QAOA y que permita resolver el problema de máximo corte.

Como ya se ha mencionado anteriormente, QAOA es un algoritmo que está concebido para poder resolver distintos tipos de problemas combinacionales, aunque el mayor desarrollo que se ha realizado hasta el momento, es para poder resolver el problema de máximo corte, y es por ello que se ha escogido este problema para utilizar el algoritmo.

En primer lugar, es necesario asignar un valor al entero p, y este es uno de los puntos clave del algoritmo, ya que de ello depende que se obtengan resultados más próximos al óptimo a cambio de un mayor coste computacional, o por el contrario, sacrificar el resultado a cambio de tener obtenerlo en un menor tiempo.

En este punto es necesario detenerse a explicar que como ya se mencionó, un ordenador clásico no puede llegar

a realizar el algoritmo QAOA, ya que en dicho caso se demostraría que P=NP, lo cual permitía que este algoritmo fuese un buen candidato para mostrar la supremacía cuántica. Sin embargo, sí que se puede llegar a utilizar un ordenador clásico que simule dicho algoritmo en una máquina cuántica virtual, como es el caso de este trabajo, pero en este caso, conforme el valor del entero p aumenta, la complejidad y el tiempo de ejecución se incrementa exponencialmente, lo cual lo hace irrealizable en la práctica para valores grandes de p. Por otra parte, en un ordenador cuántico, el tiempo de ejecución aumenta proporcionalmente a como aumenta el valor de p, por lo que esto permite obtener una gran ventaja del ordenador cuántico frente al clásico, aunque como es conocido, actualmente los ordenadores cuánticos están en desarrollo y su poder computacional está lejos de los superordenadores.

Una vez obtenido p, ya se puede proceder al cálculo de los ángulos γ y β, de tal forma que existirán 2p ángulos. Este cálculo es clave, ya que son estos ángulos de los que depende la función de onda que a su vez es la que permite resolver el problema de optimización.

Por ello, la tarea de selección de dichos ángulos no es sencilla y existen distintas estrategias para conseguir obtener un conjunto de ángulos que sean óptimos para la gráfica G que se esté estudiando. En el primer caso, cuando el número de vértices n, y el valor de p (pasos del algoritmo) son independientes, se pueden elegir los 2p ángulos de la malla

$$[0,2\pi]^p x[0,\pi]^p$$

Permitiendo encontrar de esta forma el máximo de $F_p$ variando los ángulos γ y β a través de la malla. Otra posibilidad es utilizar un algoritmo clásico como podría ser una función de coste o incluse la técnica de descenso por gradiente (gradient descent) muy utilizada en la actualidad para el machining learning o aprendizaje automático, aunque todavía se desconoce una respuesta que permita obtener dichos ángulos y en consecuencia, los operadores de forma óptima.

En nuestro caso, se utiliza una función que se ejecuta en el ordenador clásico para poder obtener los ángulos óptimos según el tipo de gráfica a la que se enfrente el algoritmo.

Por último, se procede a utilizar la función de onda en la que se encuentran superpuestos todos los posibles estados de la base de computación. Para ello, primero es necesario obtener el estado inicial, un estado uniforme en el que todos los posibles estados de la base de computación son equiprobables, como se puede observar en la ecuación 5.11. Tras esto, se aplican los operadores unitarios a dicha función de onda y se aplica a C (ecuaciones 5.10, 5.12 y 5.13).

Y una vez que ya se tiene la función de onda con las amplitudes para cada estado, solamente es necesario tomar dichas amplitudes y elevarlas al cuadrado, con lo que se obtiene finalmente la probabilidad asignada a cada uno de los posibles estados de la base de computación.

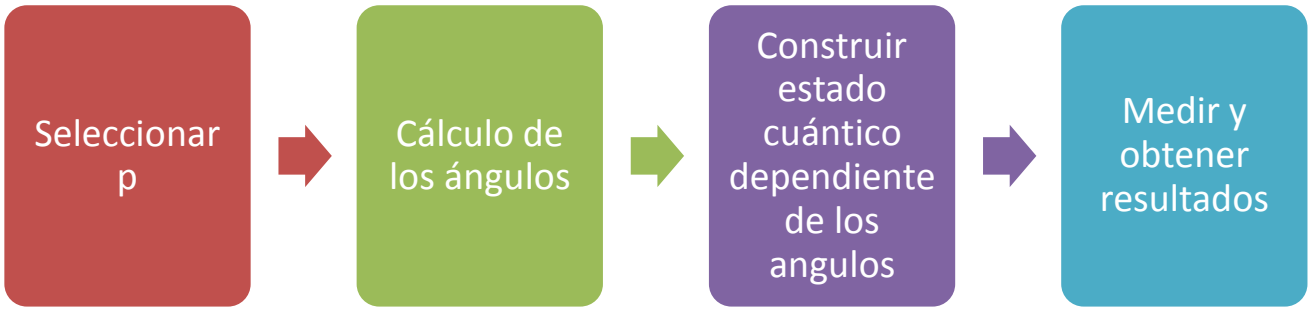


Figura 5-4. Pasos del algoritmo QAOA aplicado al problema de máximo corte

En la figura 5-4 se muestra un resumen gráfico acerca de todos los pasos que se toman a la hora de implementar el algoritmo en un computador. A continuación, se realiza una breve descripción acerca de cada uno de esos puntos y sobre qué tipo de computador se utiliza, ya sea cuántico o clásico.

1. **Seleccionar p**: Este paso se realiza en el computador clásico y se busca utilizar un valor entero que permita obtener un resultado satisfactorio.
2. **Cálculo de los ángulos:** Se calculan 2p ángulos de las múltiples formas que se han visto, aunque en este caso concreto se implementa una función de coste a la que se le introducen como parámetros de entrada el número p y la gráfica G. Esto también se realiza en un ordenador clásico, ya que la función de coste es un algoritmo clásico.
3. **Construir estado cuántico dependiente de los ángulos:** Este paso es el más importante, ya que en este caso si es necesario realizar cálculos en el ordenador cuántico, o en este caso, en el simulador de un ordenador cuántico. Primero se obtiene la función de onda inicial y uniforme para todos los posibles estados, para posteriormente aplicarle los operadores unitarios.
4. **Medir y obtener resultados:** finalmente se realiza la medición de cada uno de los posibles estados de la base computacional y se suman todos ellos de la siguiente forma:

$$C = \sum_{\langle j,k \rangle} C_{\langle j,k \rangle}(z) \tag{5.17}$$

Una vez medidas las amplitudes, ya solamente es necesario elevarlas al cuadrado y se obtiene la probabilidad para cada uno de los posibles estados.

## 5.6 Instalación y ejecución

Para finalizar, es necesario realizar un breve inciso para mostrar algunos detalles a la hora de utilizar el algoritmo QAOA y la máquina virtual cuántica (quantum virtual machine). El código está realizado en Python, por lo que es indispensable tenerlo instalado, así como algunos paquetes adicionales que permiten utilizar librerías cuánticas como Quil. Este ya se vio en el capítulo 2, en la parte final donde se realizaba un breve recorrido acerca del software más importante en la actualidad para poder realizar simulaciones de algoritmos cuánticos en ordenadores clásicos.

Tanto Quil como la máquina virtual cuántica que ha de ser instalada viene detallada en el documento [16] donde aparece una guía sobre cómo realizar la instalación de todo el software, incluido Python a través de Anaconda.

Por otra parte, antes de ejecutar el código, es necesario poner en funcionamiento la máquina virtual, y para ello se va a explicar a continuación cuales son los comandos necesarios. Es importante destacar que estos son para Windows, ya que es el sistema operativo en el que se ha realizado el trabajo, aunque estos serán similares en otros sistemas operativos.

Los comandos son *quilc –S* y *qvm –S*, y estos tienen que ser introducidos en dos pestañas distintas de Command Prompt de Windows (Símbolo de Sistema). El primero de ellos permite utilizar la base cuántica de los algoritmos; es decir, es el traductor entre un algoritmo cuántico, y la interpretación que ha de hacer el ordenador clásico para poder realizar los cálculos de forma clásica, mientras que el segundo inicia la máquina virtual cuántica, y que como se podrá observar, cuando se ejecuta el algoritmo, se puede observar cómo van apareciendo las distintas operaciones que se van realizando en dicha pestaña.

Para que se pueda comprender con mayor facilidad, se han añadido las imágenes 5-5 y 5-6 para poder visualizar dicho procedimiento.

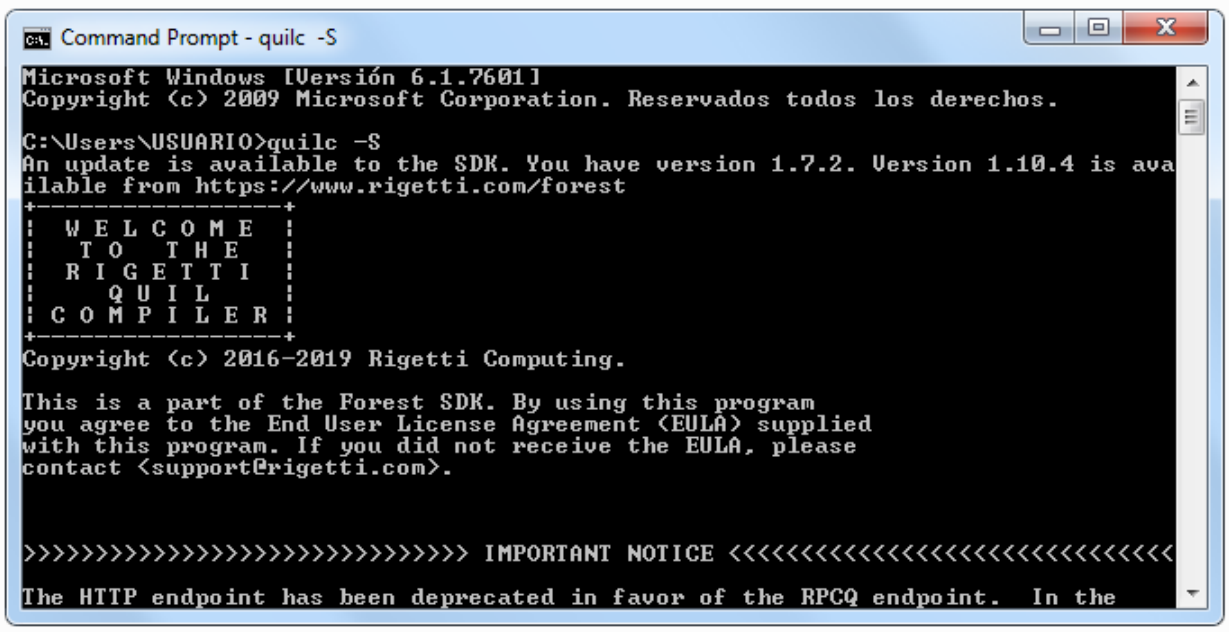


Figura 5-5. Comando quilc –S en el command prompt de Windows

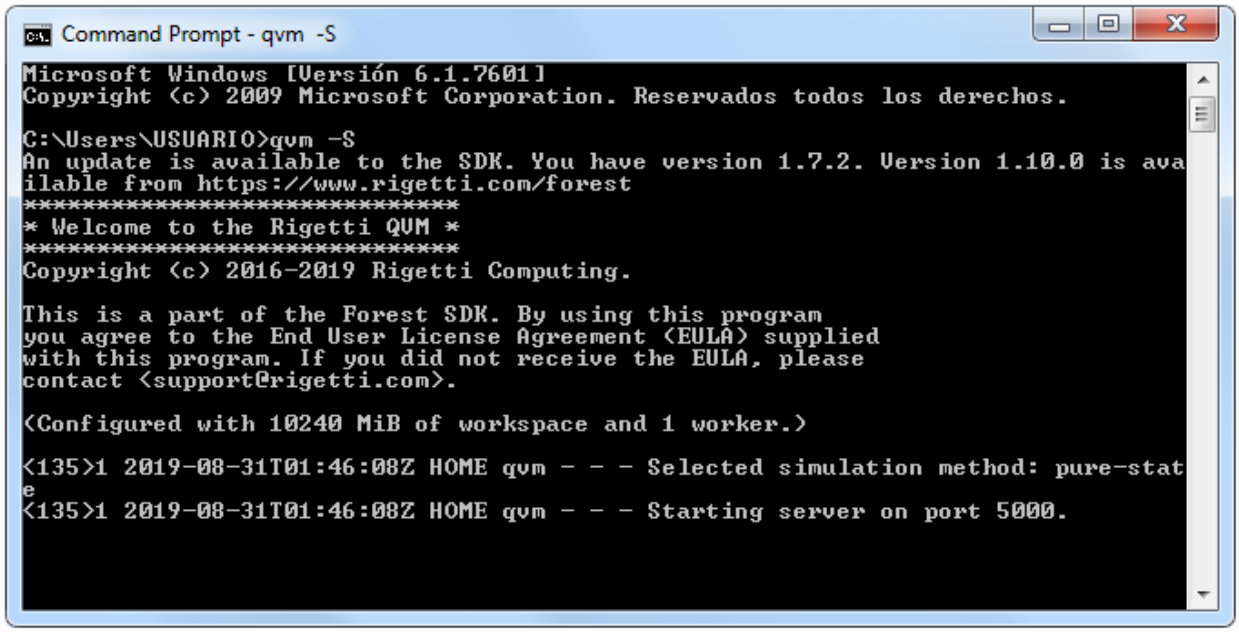


Figura 5-6. Comando qvm –S en el command prompt de Windows

Como se puede observar en la figura 5-6, la máquina virtual cuántica queda alojada en el puerto 5000, mientras que quilc queda alojado en el 6000 (no aparece en la imagen), por lo que en caso de algún problema a la hora de realizar la ejecución del algoritmo, es necesario observar si estos comandos se han ejecutado de forma correcta y los puertos que estén utilizando para crear el servidor no estén ocupados.

# 6 RESULTADOS

Hasta ahora, se ha ido realizando una descripción paso por paso desde conceptos básicos de la mecánica cuántica y la computación cuántica, hasta tratar el problema del máximo corte y finalmente el algoritmo cuántico escogido para resolverlo, QAOA. Tras esto, en este capítulo se van a mostrar los resultados obtenidos de la simulación del algoritmo en un ordenador clásico pero en el que hay instalado un simulador de ordenador cuántico que permitirá obtener resultados aproximados a los que se obtendría en un ordenador cuántico, aunque el tiempo de ejecución sea superior al necesario en un ordenador cuántico real y con una cierta cantidad de qubits (los necesarios para resolver el problema).

Para poder comprobar su funcionamiento, se van realizar varias simulaciones con distintas gráficas y configuraciones para que se pueda observar el funcionamiento del algoritmo y como este responde ante distintos parámetros, ya sea en la propia estructura de la gráfica, o con el ya mencionado entero 'p'.

Las simulaciones van a ir desde los casos más sencillos, hasta gráficas más complejas y se irá realizando varias simulaciones para cada una de ellas en las que se irá variando el parámetro 'p' para que se pueda comprobar cómo funciona y que resultados arroja en función del tipo de gráfica y de su propio valor.

## 6.1 Gráfica de un cuadrado

El primer caso que se va a estudiar es uno muy sencillo y que ya se ha observado anteriormente y se ha resuelto sin necesidad de un algoritmo, ya que se trata de una gráfica de 4 vértices y 4 segmentos unidos de forma que simulan a un cuadrado.

Para poder trazar dicha gráfica, es necesario que la matriz de la gráfica tenga el siguiente aspecto, donde cada fila de la matriz es el segmento que une dos nodos:

$$grafica = [(0,1),(1,2),(2,3),(3,0)]$$

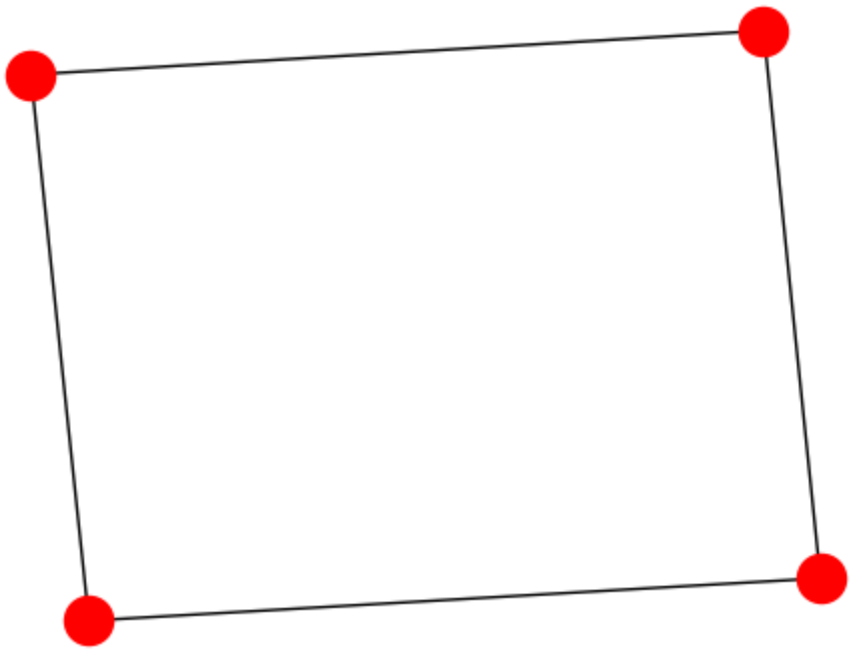

Figura 6-1. Gráfica de un cuadrado

La figura 6-1 es el resultado de trazar con la librería matplotlib de Python la red o grafo que viene determinada por la matriz anterior.

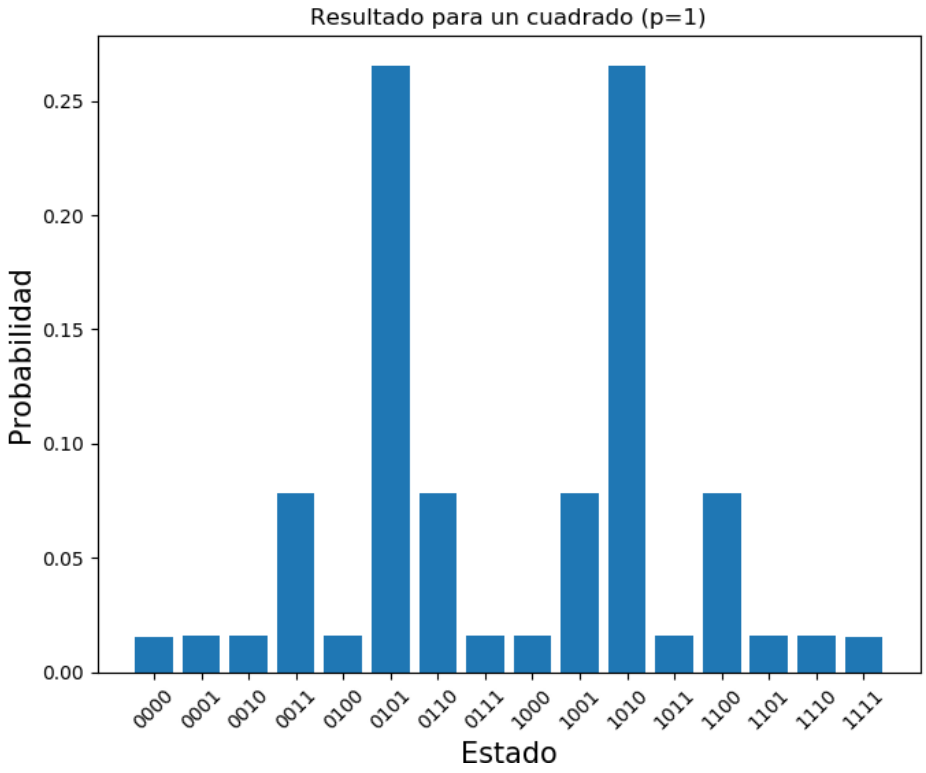


Figura 6-2. Resultados de la gráfica del cuadrado para p=1

Como se puede observar, al ser un caso tan simple, se obtienen dos claros resultados para p=1 que son los óptimos, ya que ambos son complementarios entre sí, y la diferencia reside en que el primer nodo pertenezca al conjunto S o a su complementario.

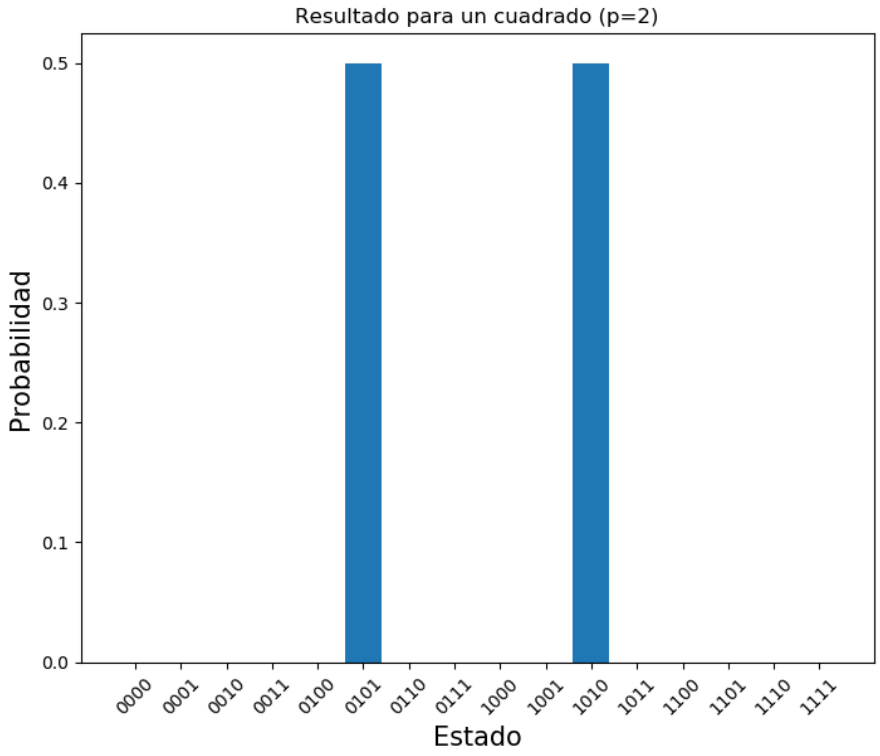


Figura 6-3. Resultados de la gráfica del cuadrado para p=2

Sin embargo, al aumentar p a 2, ya si se obtiene un resultado claro, y es que la solución y su complementario

aparecen como las dos únicas posibles soluciones, los estados 0101 y 1010, los cuales dan como solución la gráfica vista en el capítulo 4 (figura 4-2).

Este cambio radical entre p=1 y p=2 se debe en gran parte a una propiedad que ya se ha mencionado, y es que el valor de p no es otra cosa si no que la circunferencia de radio p en la que quedan contenidos todos los segmentos que estén distanciados una distancia máxima p del segmento (j,k) que se está evaluando, y en el caso del cuadrado, todos los segmentos se encuentran a una distancia $\leq 2$.

En cambio, para poder ver más claramente este efecto, se va a proceder a mostrar un caso más simple a continuación, pero que por el contrario, para valor de p=2 no se obtiene un resultado tan exacto como sucede con el cuadrado.

## 6.2 Gráfica de una línea recta

En este caso se va a evaluar un caso incluso más sencillo que el anterior, una línea recta formada por 4 nodos (vértices) y 3 segmentos que las unen, dando lugar a dos nodos exteriores y 2 nodos interiores, como se puede observar en la figura 6-4. El matriz de la red es la siguiente:

$$grafica = [(0,1),(1,2),(2,3)]$$

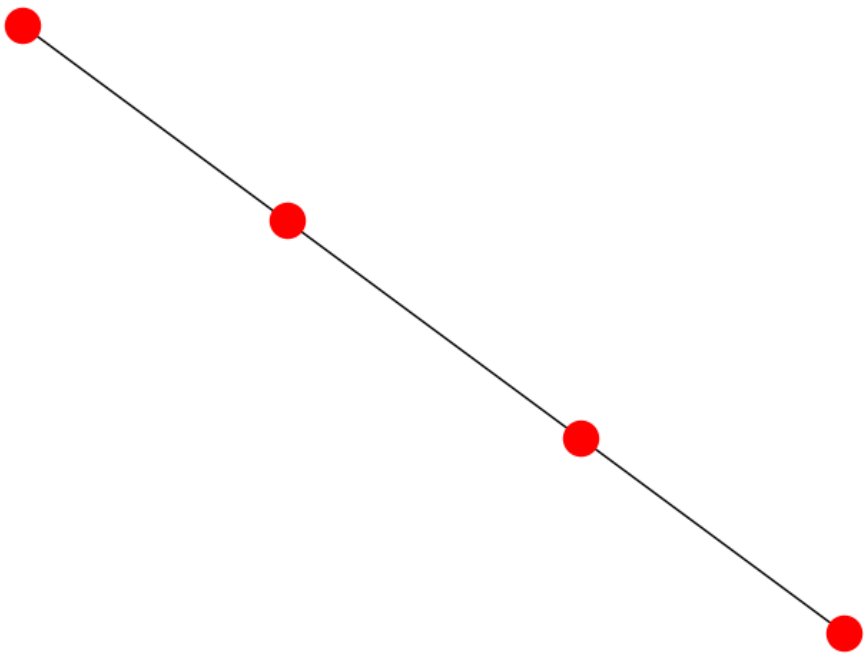

Figura 6-4. Gráfica de una línea recta

Esta gráfica o red de nodos es claramente una simplificación del caso anterior (cuadrado), ya que se elimina el segmento que une los nodos 3 y 0, dando por ello una mayor simplicidad a la estructura de la red. Sin embargo, esto provoca que el nodo 0 y el nodo 3 estén distanciados por 3 segmentos, en vez de estar separados por un solo segmento como sucedía en el cuadrado. Es por ello, que para la misma p que en el cuadrado (p=2) se obtendrán resultados peores, aunque al ser un caso sencillo, seguirá observándose como el caso óptimo destaca sobre el resto.

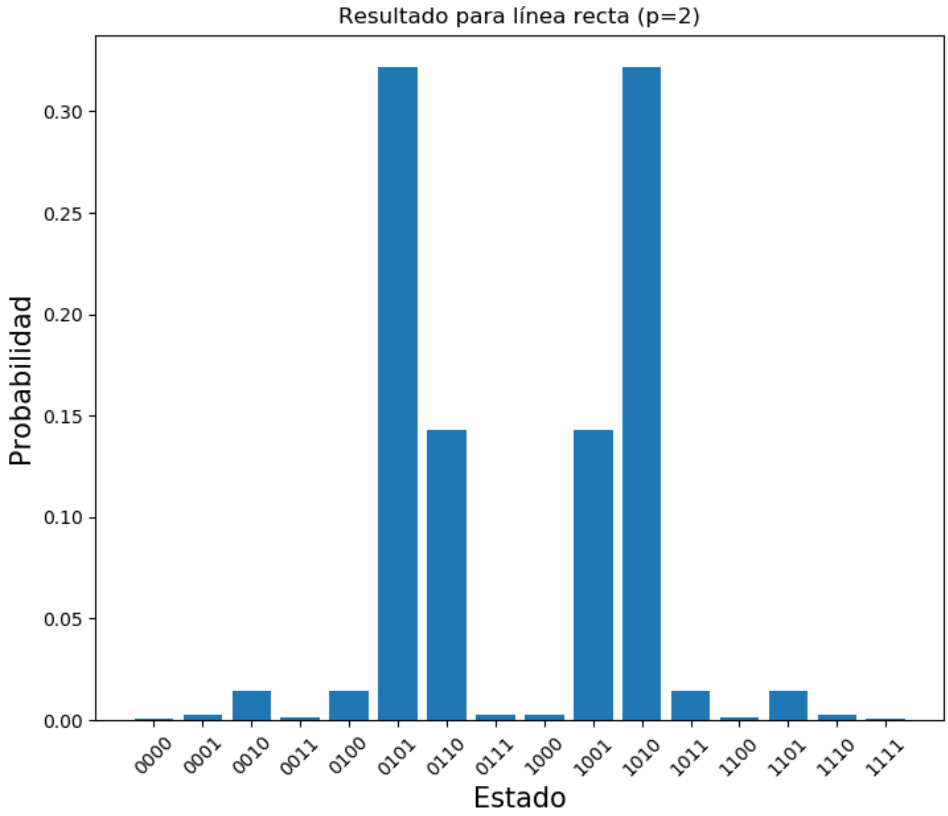


Figura 6-5. Resultados de la gráfica de la recta para p=2

Se observa que para p=2, existen 2 resultados equiprobables que son complementarios (0101 y 1010) y que son los óptimos, pero existen otros 2 resultados que también son complementarios y equiprobables que no se corresponden con el valor óptimo (0110 y 1001), ya que estos dos resultados obtienen un valor de máximo corte de 2, mientras que el óptimo es 3. A pesar de ello, este valor sigue siendo 2/3 el valor máximo, por lo que a pesar de que se introduzca una solución que no es óptima, si queda claro que esta solución es cercana al valor óptimo para esta gráfica.

Otro aspecto a destacar es en este caso la no dependencia de p y n, ya que mientras que n ha permanecido constante (mismo número de nodos), e incluso ha disminuido el número de segmentos (3 frente a 4), p tiene que tomar un valor superior en este caso si se quieren obtener resultados tan buenos como para la gráfica del cuadrado.

## 6.3 Gráfica de una red compleja

Una vez estudiado la importancia de escoger una p adecuada a la hora de realizar la simulación del algoritmo, se procede a simular un caso más complejo que los vistos hasta este momento y que requiere de valores superiores de p para obtener un resultado más fiable, aunque esto conlleva un aumento considerable del tiempo de ejecución del algoritmo.

El valor de la matriz de la gráfica a estudiar es el siguiente

$$grafica = [(0,4),(0,5),(1,4),(1,5),(2,3),(2,4),(4,5),(3,5),$$
$$(3,6),(6,1),(1,2),(5,6),(4,8),(5,7),(2,8),(3,8),(7,8)]$$

Y que da como resultado la red de la figura 6-6

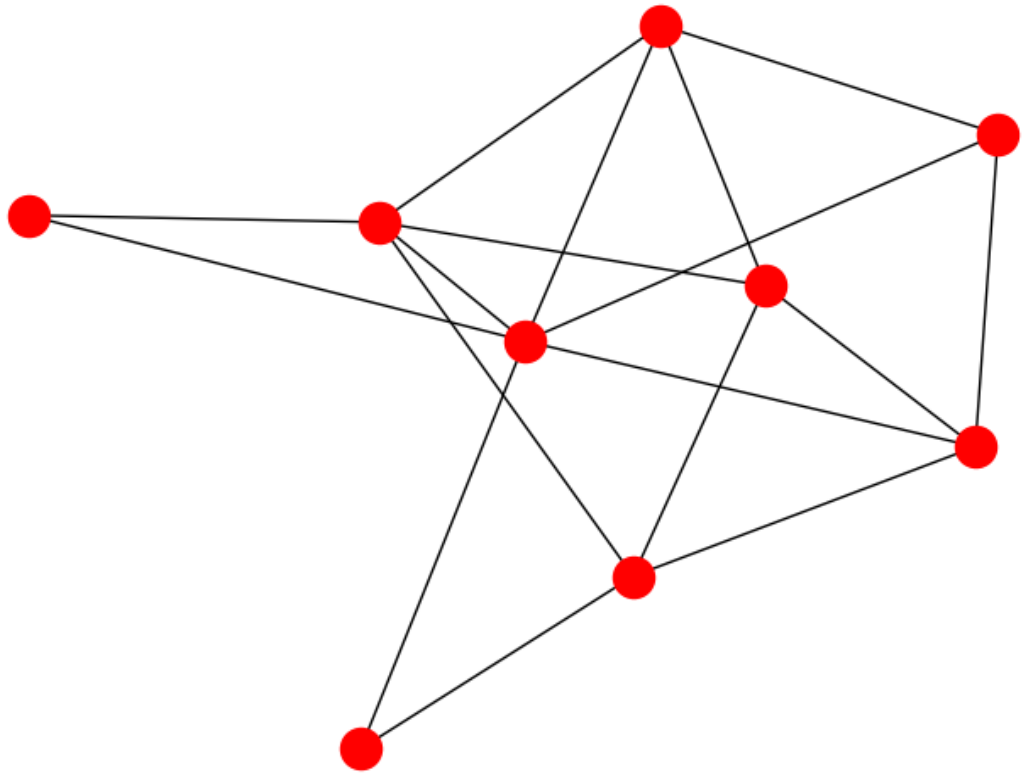

Figura 6-6. Gráfica de una red compleja

Esta red está constituida por un total de 9 nodos y 18 segmentos que los unen entre sí, y cuya solución no es trivial, ya que a pesar de no tener una estructura muy compleja, ya que en caso de utilizar estructuras más realistas a las posibles aplicaciones, estas necesitarían tiempos de ejecución muy superiores, irrealizables en la práctica en un ordenador clásico. De hecho, esta red ya comienza tener tiempo de ejecución elevados (aproximadamente 15-20 minutos en mi caso), y que en comparación con los casos anteriores (10-15 segundos) permite conocer la envergadura y el coste computacional que puede llevar a cabo realizar casos mucho más complejos que los estudiados hasta el momento.

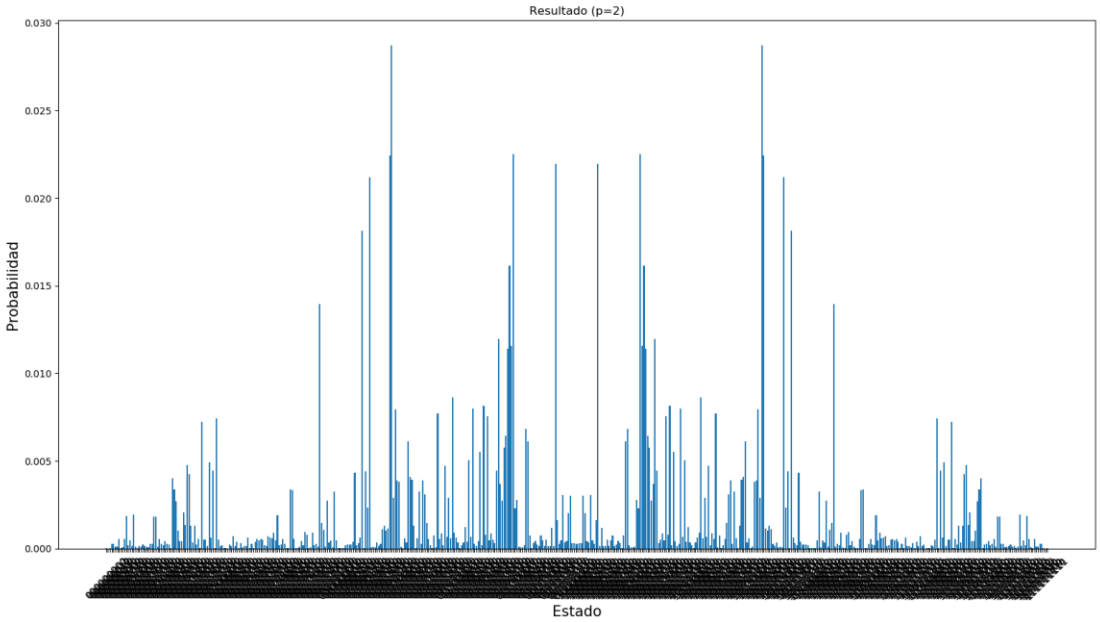


Figura 6-7. Resultados de la gráfica compleja para p=2

Para el caso de p=2 se obtienen múltiples posibles soluciones, entre las que destaca una y su complementaria, la cual, como se verá más adelante, es una solución óptima. A pesar de que existen una gran cantidad de posibles estados como solución, el algoritmo establece que todas estas soluciones, son óptimas o cercanas al valor óptimo, es por ello, que incluso utilizando un valor de p=2, se pueden obtener soluciones muy acertadas para gráficas más complejas, como sucede en este caso.

Sin embargo, se va a aumentar p para poder observar cómo evolucionan las probabilidades para cada estado, hasta poder obtener finalmente la solución óptima del problema.

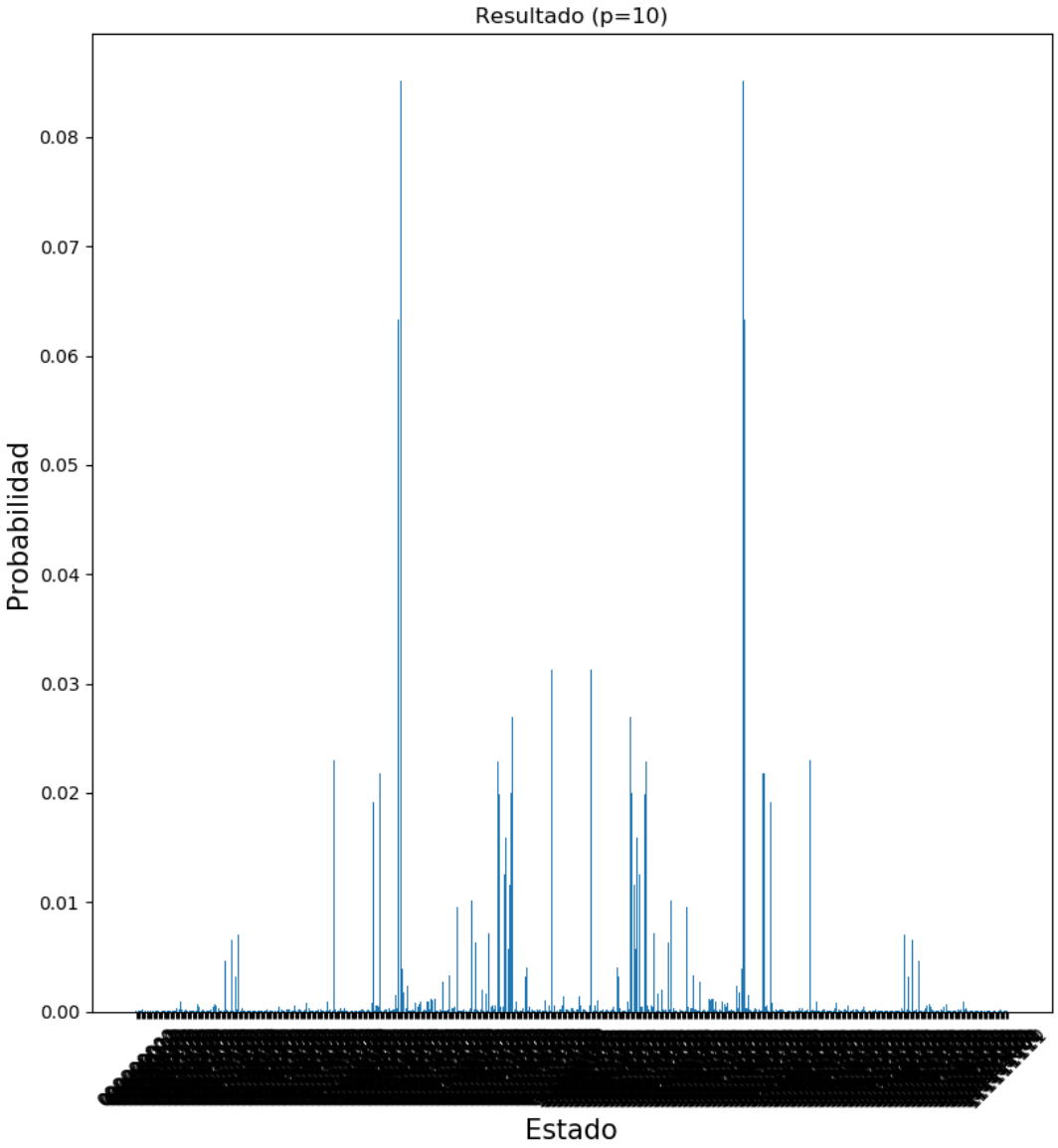


Figura 6-8. Resultados de la gráfica compleja para p=10

En esta figura, se ha aumentado p hasta 10, provocando un aumento del tiempo de ejecución considerable, y aunque se obtienen dos soluciones y sus complementarias de forma bastante destacada, siguen apareciendo probabilidades relativamente altas para otros posibles estados que no son óptimos. Estas dos soluciones (y sus complementarias) son las que se pueden apreciar en la figura 6-9.

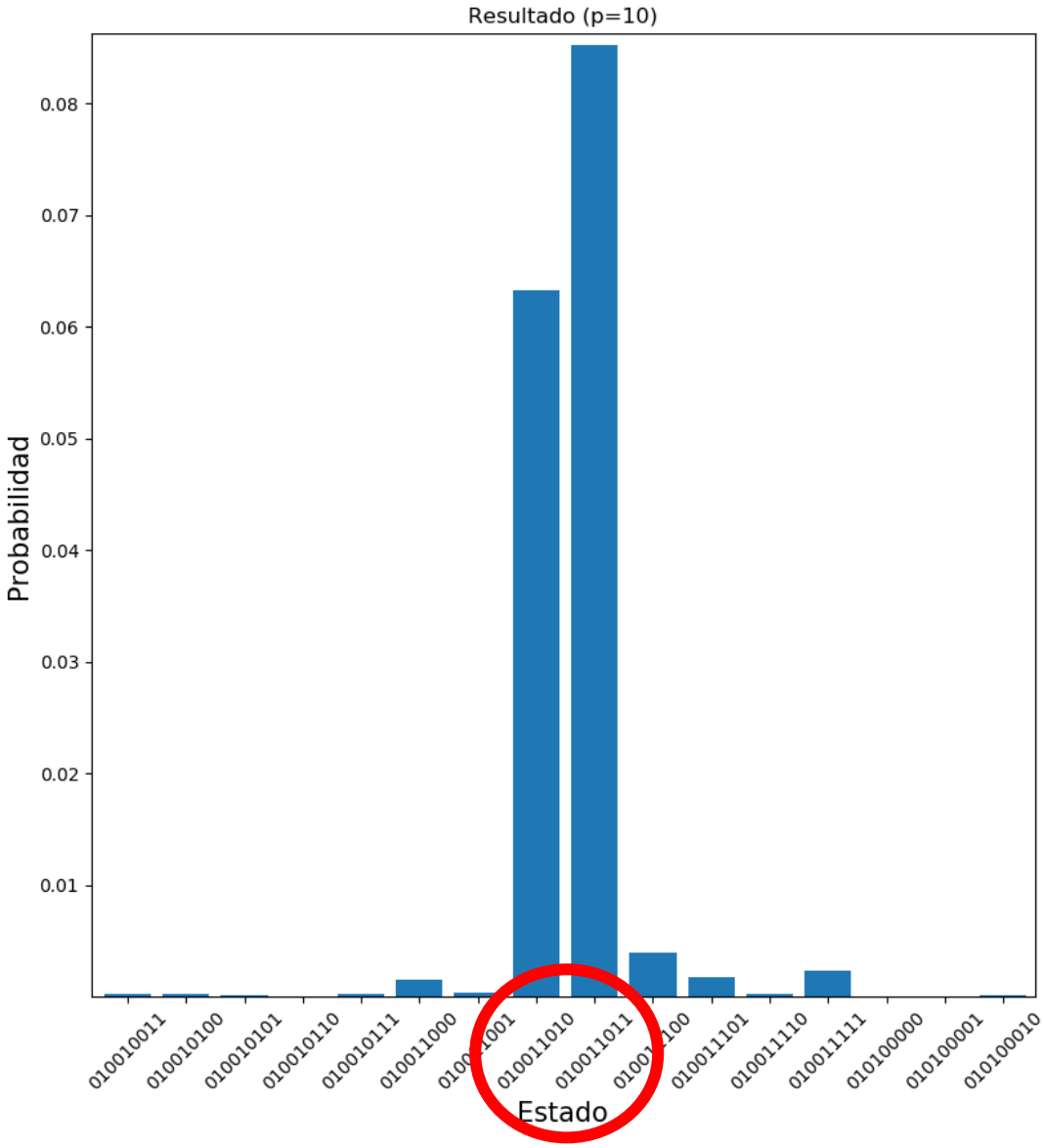


Figura 6-9. Resultados de la gráfica compleja para p=10 realizando zoom sobre las soluciones

Se observa como los estados 010011010 y 010011010 son los más probables de ser la solución óptima, aunque como se verá más adelante, solo uno de ellos es el óptimo, y en contra de lo que pueda pensarse, ese valor es el 010011010, el cuál es el que tiene menor probabilidad para p=10 entre estos dos, y esta es una de las razones por las que este es un algoritmo aproximado, puesto que permite obtener soluciones cercanas a la óptima con cierta facilidad (para p=2 ya se conseguían resultados similares), pero no es eficaz para obtener resultados exactos, ya que por definición estos solo se dan cuando p tiene a infinito, aunque para casos más simples como los que se han estudiado, el valor óptimo se consigue con valores menores de p.

Por último, se ha realizado una última simulación para p=25 para intentar buscar una solución en la que el estado óptimo obtenga unas probabilidades superiores, y que por lo tanto destaque más respecto a otros estados menos óptimos.

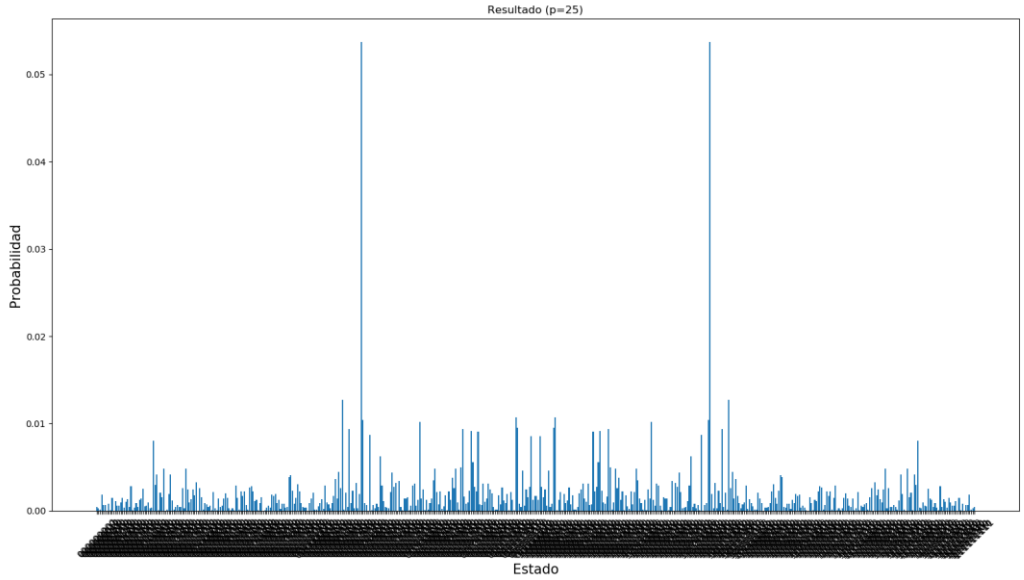


Figura 6-10. Resultados de la gráfica compleja para p=25

En esta ocasión, si se consigue que el valor óptimo destaque por encima del resto, pero por otra parte, como se ha mencionado antes, el tiempo empleado para conseguir dicha solución es muy superior al de todos los casos anteriores, incluso para p=10, ya que cuando se implementa este algoritmo en un ordenador clásico, el coste computacional en relación a p aumenta exponencialmente.

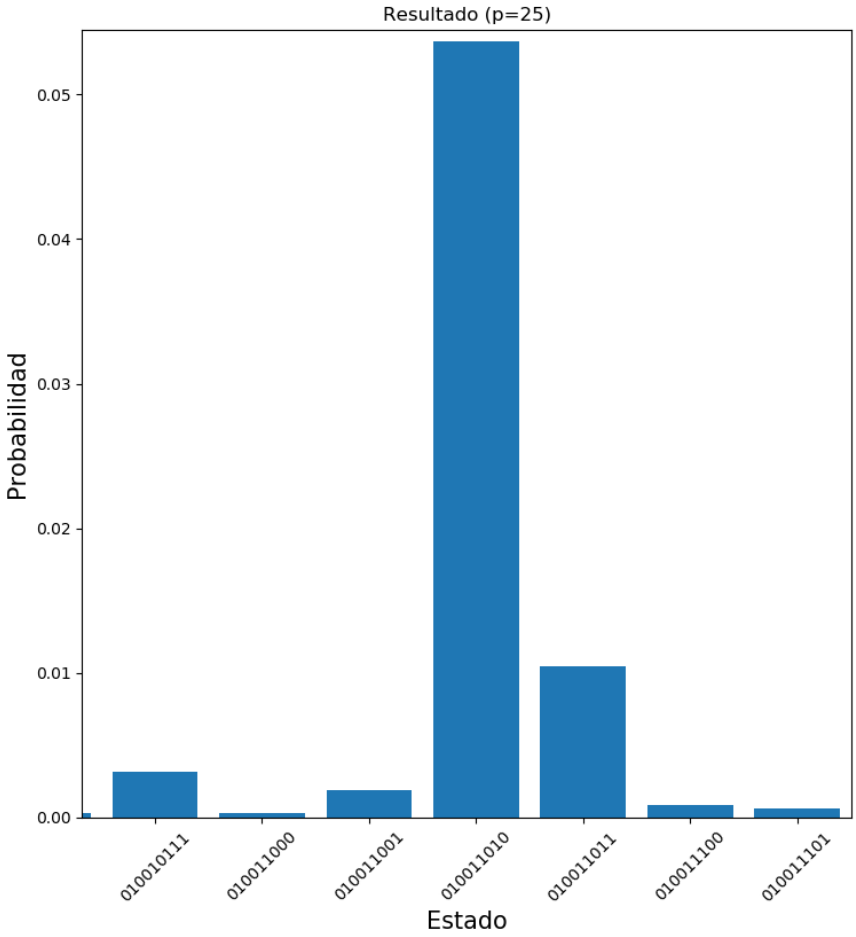


Figura 6-11. Resultados de la gráfica compleja para p=25 realizando zoom sobre la solución

El resultado final es por tanto el estado 010011010, cuya representación gráfica de la distribución de los nodos en los dos conjuntos (S y $\overline{S}$) y los segmentos que aportan al máximo corte quedaría de la siguiente forma:

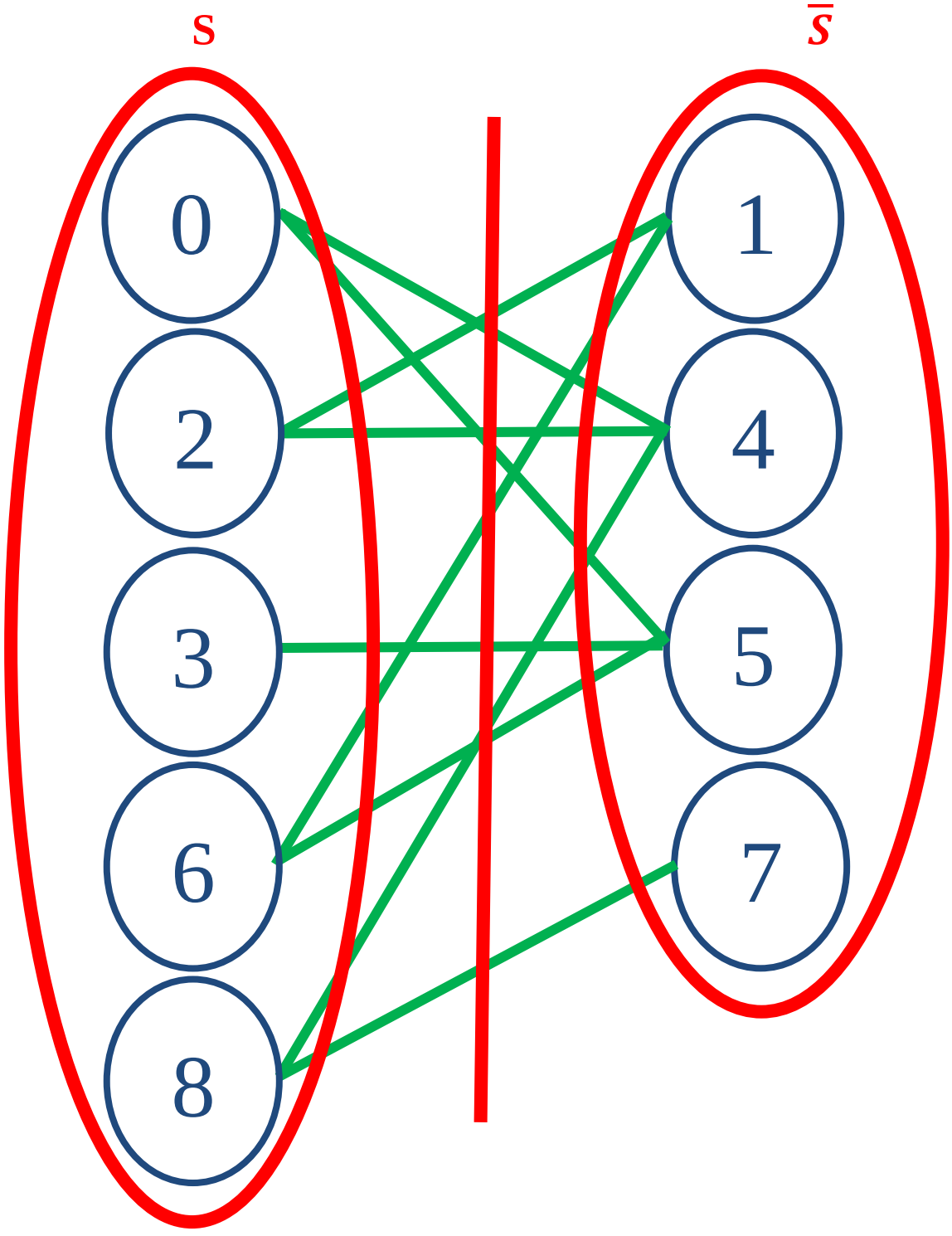


Figura 6-12. Solución gráfica de los nodos y sus conjuntos

A partir de esta figura se puede obtener fácilmente el valor óptimo de máximo corte de la gráfica, y este es 9 (contar el número de segmentos verdes que son cortados por la línea roja). Como se ha mencionado anteriormente, esta solución no es trivial, pero no es compleja para un algoritmo clásico, por lo que lo que justifica la utilización de este algoritmo frente a otros ya existentes es su capacidad de conseguir resultados aproximados con un buen ratio de cercanía al valor óptimo para valores de p bajos, y por lo tanto, bajo tiempo de ejecución. Además, a pesar de que en el ordenador clásico sea más costoso simular este algoritmo, puesto

que no se puede aprovechar todas las virtudes de un ordenador cuántico, se consiguen resultados prometedores y que pueda permitir que cuando se desarrollen los ordenadores cuánticos con una mayor cantidad de qubits, algoritmos como este puedan ofrecer una gran mejora frente a lo establecido con algoritmos clásicos.

# 7 CONCLUSIÓN Y FUTURAS AMPLIACIONES

Tras haber realizado un recorrido desde los puntos más básico de la computación cuántica, así como de sus bases teóricas; se ha alcanzado el objetivo del trabajo, aplicar un algoritmo cuántico a un problema combinatorio, como es el caso del problema del máximo corte. Esto se debe a que este tipo de problemas llevan asociados largos tiempo de ejecución para poder obtener resultados que se acerquen al óptimo. Sin embargo, la aparición de la tecnología cuántica aplicada a la computación ha abierto una nueva posibilidad de resolver estos tipos de problemas de forma que se obtenga un coste computacional inferior al necesario en los ordenadores clásicos.

Uno de los ejemplos más notorios es el del algoritmo de Schor, el cual ha despertado el interés y a su vez, la inquietud, ya que su aplicación al ámbito de la encriptación podría permitir desencriptar algunos de los protocolos más conocidos en la actualidad como es el caso de RSA.

A pesar de ello, como se ha podido ir observando a lo largo de este trabajo, QAOA es un algoritmo prometedor, ya que evita algunos de los problemas más importantes que existen en la actualidad con los algoritmos cuánticos y con la propia construcción de un ordenador cuántico, como es la corrección de errores, puesto que como se ha explicado, este algoritmo no necesita realizar un complejo procedimiento de corrección de errores una vez finalizadas las operaciones, a diferencia de otros algoritmos cuánticos, que necesitan invertir una gran cantidad de qubits para ello. En cambio, en contra tiene el punto de que es necesario emplear un qubit por cada nodo de la red (vértice), y sumado a que el ordenador cuántico con un mayor número de qubits en la actualidad tiene solamente 72, esto restringe enormemente su aplicación y utilización para casos prácticos reales. Es por ello, que este algoritmo todavía necesita que se desarrollen ordenadores cuánticos con un mayor número de qubits para poder comprobar realmente su funcionamiento para casos complejos. En caso contrario, quizás sea necesario implementar otro tipo de algoritmo en el que el número de qubits no dependa directamente del número de vértices de la red, aunque es difícil pensar una solución a este problema.

Por otra parte, QAOA es uno de los algoritmos que se puede utilizar para demostrar la supremacía cuántica, concepto acuñado a la posibilidad de que en algún momento, los ordenadores cuánticos sean capaces de obtener resultados mejores que los ordenadores clásicos cuando se enfrenten a problemas complejos como el que se aborda en este trabajo. Esto se debe a que se ha demostrado que es imposible realizar una adaptación de este algoritmo para un ordenador clásico y que esta sea capaz de respetar todos los procedimientos que se hacen a bajo nivel, ya que en caso contrario, se aseguraría que P=NP, o lo que es lo mismo, que los problemas considerados actualmente como de tiempo no polinomial, pasarían a ser de tiempo polinomial, permitiendo ser resueltos en una cantidad de tiempo limitada y que en ningún momento crecería exponencialmente cuando se complica el problema, resolviéndose uno de los Problemas del Milenio, y la trasecendencia que esto tendría.

Una vez tenidos todos estos puntos en cuenta, se ha realizado la simulación del algoritmo y a pesar de que los resultados obtenidos no se ajusten a los que se obtendrían en un ordenador cuántico, por lo dicho anteriormente, se ha observado como el valor del entero p es importante a la hora de implementar y ejecutar el algoritmo, ya que el tiempo de ejecución en un ordenador clásico crece exponencialmente cuando crece el valor de p. Es por ello, que para simulaciones como para p=25 se han obtenido tiempo de ejecución superiores a una hora para casos relativamente simples en comparación con los que se llevarían a cabo en casos reales como el de la aplicación del máximo corte a la evaluación de un circuito integrado, donde el número de transistores MOS, y por lo tanto de nodos de la red es muy superior a los utilizados en los ejemplos de este trabajo (n=9).

En cambio, un aspecto positivo se puede observar claramente al realizar simulaciones para p bajos, como p=2, ya que a pesar de que se obtienen soluciones que no son la óptima, estas soluciones toman valores cercanos a la óptima, y de hecho, anteriormente se ha mencionado como para un caso concreto como es el de gráficas de la clase p=1 y de grado 3 (máximo 3 segmentos unidos a un solo nodo), se obtiene que como mínimo el valor óptimo obtenido por el algoritmo será más cercano al máximo de la gráfica que si se realizase un procedimiento aleatorio, por lo que solo queda esperar que esto mismo se pueda esperar para casos más complejos, permitiendo al algoritmo obtener valores cercanos al óptimo (valores aproximados) para valores de p bajos, por lo que el tiempo de ejecución será bajo.

Por otra parte, en cuanto a posibles ampliaciones del trabajo se podrían escoger distintas vertientes, ya que la computación cuántica es uno de los aspectos más novedosos que existen en la actualidad en lo referente a

investigación de una nueva forma de realizar algoritmos y en general, de estructurar la informática y las telecomunicaciones. Esto se debe a que continuamente aparecen nuevos algoritmos con aplicaciones en distintos ámbitos, ya sea la optimización como en este caso, aplicaciones médicas o incluso para resolver problemas de la propia mecánica cuántica que en la actualidad se encuentran en estudio.

Otro de los mayores puntos a favor de la mecánica cuántica es que el problema de la medida; es decir, que la función de onda colapsa a un valor concreto cuando esta es medida y no puede volver a ser recuperada en su forma anterior (la medida no es un proceso reversible), y junto a que no se puede realizar copias de estados cuánticos debido al teorema de no clonación como ya se ha visto, esto implica que sea una gran ventaja a la hora de realizar protocolos de encriptación, ya que si se quiere enviar un mensaje entre dos sujetos separados, si existiese un intruso en el medio de comunicación, este no podría estar espiando los mensajes, ya que una vez que mide un qubit del protocolo, la función de onda colapsa, y el receptor puede darse cuenta de dicho problema y por lo tanto descartar ese medio de comunicación y utilizar otro, siendo en dicho caso imposible que exista un intruso en la comunicación y que este pase desapercibido para los dos sujetos que se están comunicando. Por eso, este campo es muy interesante para poder realizar un trabajo, aunque como pasa en todos los algoritmos que se siguen desarrollando hoy en día, tienen una fuerte dependencia de conocimientos en áreas como la mecánica cuántica y sus propiedades, por lo que muchos de los expertos son físicos y no ingenieros o informáticos, ya que es necesario tener estos conocimientos para poder realizar trabajos al respecto.

# APÉNDICE A: CÓDIGO PYTHON

```
1. import numpy as np
2. from grove.pyqaoa.maxcut_qaoa import maxcut_qaoa
3. import pyquil.api as api
4.
5. import networkx as nx
6. import matplotlib.pyplot as plt
7.
8.
9. qvm_connection = api.QVMConnection()
10.
11.       #Valores a cambiar-----------------
12.       grafica =[(0,1),(1,2),(2,3),(3,0)]
13.       p = 2
14.       #---------------------------------
15.
16.       grafica_maximocorte = nx.Graph()
17.       for j in grafica:
18.           grafica_maximocorte.add_edge(*j)
19.       red = grafica_maximocorte.copy()
20.
21.       nx.draw(red)
22.
23.       #Se inician los cálculos de los angulos beta y gamma
24.       inst = maxcut_qaoa(graph=grafica, steps=p)
25.
26.       betas, gammas = inst.get_angles()
27.
28.
29.       #Se calculan los estados a partir de los angulos y sus respectivas
    funciones de onda
30.       t = np.hstack((betas, gammas))
31.       parametros = inst.get_parameterized_program()
32.       prog = parametros(t)
33.       wf = qvm_connection.wavefunction(prog)
34.       wf = wf.amplitudes
35.
36.       for i in range(inst.nstates):
37.           print(inst.states[i], np.conj(wf[i])*wf[i])
38.
39.
40.       plt.figure(2)
41.       plt.bar(inst.states,abs(wf*wf))
42.       plt.xlabel('Estado', fontsize=15)
43.       plt.ylabel('Probabilidad', fontsize=15)
44.       plt.xticks(inst.states, fontsize=10, rotation=45)
45.       plt.title('Resultado para un cuadrado (p=2)')
46.       plt.show()
```

# ÍNDICE DE TABLAS

Tabla 2-1. Principales QPUs según nº de qubits 11

# ÍNDICE DE FIGURAS


Figura 1-1. Ejemplo de suma de 2 qubits (puertas CCNOT y CNOT) 2
Figura 1-2. Ejemplo de entrelazamiento de 2 qubits (puerta H+CNOT) 3
Figura 1-3. Puerta de medición 5
Figura 2-1. Representación de la esfera de Bloch para los estados de un qubit. 8
Figura 2-2. Circuito cuántico para puerta Hadamard 12
Figura 2-3. Resultado de la simulación de una puerta Hadamard 12
Figura 3-1. Casuística de la puerta Hadamard (H) 15
Figura 3-2. Puerta X de Pauli (NOT) 16
Figura 3-3. Puerta Y de Pauli 17
Figura 3-4. Puerta Z de Pauli 17
Figura 3-5. Puerta SWAP 18
Figura 3-6. Circuito con una puerta Hadamard y una CNOT 18
Figura 4-1. Ejemplo de máximo corte aplicado a un cuadrado 21
Figura 4-2. Ejemplo de máximo corte aplicado a un cuadrado (resultado óptimo) 22
Figura 4-3. Ejemplo de máximo corte complejo 23
Figura 4-4. Branch and Bound aplicado al problema de la mochila 24
Figura 4-5. Comparativa de una figura plana (izquierda) con una no plana (derecha) 25
Figura 5-1. Representación del estudio del máximo corte de un eje para p=2 30
Figura 5-2. Gráficas posibles para p=1 y 3 segmentos 31
Figura 5-3. Estado inicial de superposición uniforme 33
Figura 5-4. Pasos del algoritmo QAOA aplicado al problema de máximo corte 35
Figura 5-5. Comando quilc –S en el command prompt de Windows 37
Figura 5-6. Comando qvm –S en el command prompt de Windows 37

Figura 6-1. Gráfica de un cuadrado 39
Figura 6-2. Resultados de la gráfica del cuadrado para p=1 40
Figura 6-3. Resultados de la gráfica del cuadrado para p=2 40
Figura 6-4. Gráfica de una línea recta 41
Figura 6-5. Resultados de la gráfica de la recta para p=2 42
Figura 6-6. Gráfica de una red compleja 43
Figura 6-7. Resultados de la gráfica compleja para p=2 43
Figura 6-8. Resultados de la gráfica compleja para p=10 44
Figura 6-9. Resultados de la gráfica compleja para p=10 realizando zoom sobre las soluciones 45
Figura 6-10. Resultados de la gráfica compleja para p=25 46
Figura 6-11. Resultados de la gráfica compleja para p=25 realizando zoom sobre la solución 46
Figura 6-12. Solución gráfica de los nodos y sus conjuntos 47

# REFERENCIAS


[1] Giuliano,B., Giulio,C., & Giuliano, S. (2004). Principles of quantum computation and information (Volume I: Basic concepts). Singapore: World Scientifi, 108-111

[2] Benenti, G., Casati, G., & Strini, G. (Eds.). (2007). Principles of quantum computation and information. Volume II: Basic Tools and Special Topics. World Scientific Publishing Company

[3] Nielsen, M. A., & Chuang, I. L. (2001). Quantum computation and quantum information. Phys. Today, 54, 60-2.

[4] Schlosshauer, M. (2005). Decoherence, the measurement problem, and interpretations of quantum mechanics. Reviews of Modern physics, 76(4), 1267.

[5] Aharonov, D. (1999). Quantum computation. In Annual Reviews of Computational Physics VI (pp. 259-346).

[6] Venegas-Andraca, S. E., Cruz-Santos, W., McGeoch, C., & Lanzagorta, M. (2018). A cross-disciplinary introduction to quantum annealing-based algorithms. Contemporary Physics, 59(2), 174-197.

[7] Kadowaki, T., & Nishimori, H. (1998). Quantum annealing in the transverse Ising model. Physical Review E, 58(5), 5355.

[8] Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D. P., Margolus, N., Shor, P., ... & Weinfurter, H. (1995). Elementary gates for quantum computation. Physical review A, 52(5), 3457.

[9] Scribe, S., & Karimi, S. (2007). Max-cut Problem.

[10] Rendl, F., Rinaldi, G., & Wiegele, A. (2010). Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. Mathematical Programming, 121(2), 307.

[11] Commander, C. W. (2009). Maximum cut problem, max-cut. Encyclopedia of Optimization, 1991-1999.

[12] Hadlock, F. (1975). Finding a maximum cut of a planar graph in polynomial time. SIAM Journal on Computing, 4(3), 221-225.

[13] Farhi, E., Goldstone, J., & Gutmann, S. (2014). A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028.

[14] Farhi, E., & Harrow, A. W. Quantum supremacy through the quantum approximate optimization algorithm, 2016. arXiv preprint arXiv:1602.07674.

[15] Wang, Q., & Abdullah, T. (2018). An Introduction to Quantum Optimization Approximation Algorithm.

[16] [Grove es una librería de Python de código abierto destinada a la programación de algoritmos cuánticos] (s.f.). n/a. Instalación desde https://grove-docs.readthedocs.io/en/latest/installation.html