



***Facultad
de
Ciencias***

**RECONOCIMIENTO DE TRAZAS EN UN
DETECTOR DE PIXELS USANDO
ALGORITMOS DE COMPUTACIÓN
CUÁNTICA**

(Track reconstruction in a pixel detector
using quantum computing algorithms)

Trabajo de Fin de Grado
para acceder al

GRADO EN FÍSICA

Autor: Pablo Reoyo López

Director: Pablo Martínez Ruiz del Árbol

Septiembre - 2022

Resumen / Abstract

Resumen:

Con el presente trabajo se buscan explorar los fundamentos de la computación cuántica, a la par que se muestre el camino que ha tomado esta ciencia incipiente en los últimos años. Además, se expondrá su utilidad mediante la implementación de un algoritmo cuántico al proceso de reconstrucción de trazas, definidas a partir de los *hits* en un detector de partículas. Al finalizar el texto, uno será capaz de comprobar que, más allá de ser una curiosidad teórica, la computación cuántica se alza como una nueva forma de afrontar problemas físicos y que algunas de sus ventajas pueden disfrutarse en la actualidad.

Palabras clave: *Computación cuántica, Física de partículas, Mecánica cuántica, Qiskit, VQE*

Abstract:

With this work we aim to explore the basics of quantum computing, while showing the path this new technology is taking in the last few years. Furthermore, its utility will be showed by applying a quantum algorithm to the track reconstruction process in a particle detector. At the end of this work, one will be able to prove that, being more than just a simple theoretical curiosity, quantum computing has become a new framework for solving physics problems and that some of its advantages can be used in the present day.

Key words: *Quantum Computing, Particle Physics, Quantum Mechanics, Qiskit, VQE*

Índice general

Resumen	I
1 Introducción	1
2 Fundamentos de la computación cuántica	2
2.1 Postulados de la mecánica cuántica	3
2.2 Conceptos básicos y notación	5
2.2.1 <i>Qubits</i>	5
2.2.2 Puertas y circuitos cuánticas	5
2.2.3 Superposición y entrelazamiento	8
2.3 Implementación de los ordenadores cuánticos	9
2.3.1 Condiciones para su existencia	9
2.3.2 Mitigación de errores	13
3 Complejidad computacional y utilidad cuántica	15
3.1 Desde la perspectiva de la complejidad computacional	16
3.2 Introducción a los algoritmos híbridos	18
3.2.1 En la época NISQ	19
3.2.2 El algoritmo VQE	22
4 Algoritmo cuántico de reconstrucción	25
4.1 Características del detector	25
4.2 El simulador de eventos	27
4.3 El algoritmo de <i>tracking</i>	30
5 Resultados y análisis	34
5.1 Obtención de los parámetros de coste	35
5.2 Resultados	36

5.2.1	Análisis del rendimiento	36
5.2.2	Análisis del entrelazamiento	40
5.2.3	Análisis de la profundidad del circuito	44
6	Conclusiones	46
	Referencias	47

Capítulo 1

Introducción

Frente a la computación tradicional, construida en el marco de un estado físico que puede tener dos valores posibles, en los últimos años un nuevo paradigma está emergiendo en el mundo de la computación, la computación cuántica. A diferencia de la primera, el nuevo marco de trabajo consiste en emplear un estado físico que se encuentra en una superposición de dos estados: 0 y 1. En apariencia confuso y ocioso, el cambio a computación cuántica tiene un gran interés tanto teórico como práctico, pues abre las puertas a la instauración de una nueva rama de la lógica, al tiempo que permite, potencialmente, reducir un problema con complejidad computacional de orden exponencial a escalas polinomiales.

Así y todo, el objetivo del trabajo se centra en proporcionar una explicación breve pero rigurosa de los fundamentos de la computación cuántica, poniéndose de manifiesto su relevancia mediante la resolución de un problema relacionado con la física experimental de altas energías: la reconstrucción de trazas en un detector de partículas.

El trabajo se estructura de la siguiente manera. Con los libros *Quantum Computation and Quantum Information* ([1]) y *Principles of Quantum Computation and Information - Volume I: Basic Concepts* ([2]) se ha intentado construir una exposición breve y lo más completa posible sobre todos los conceptos básicos de la computación cuántica, lo cual conforma el segundo capítulo. A continuación, el tercer bloque se encarga de introducir, desde la perspectiva de la complejidad computacional, el algoritmo cuántico central para este trabajo. Después, en la cuarta parte del trabajo, se muestra la investigación que se ha llevado a cabo para conseguir aplicar los conocimientos adquiridos sobre computación cuántica al problema de la reconstrucción de trazas. Dejándose para los dos últimos capítulos los resultados obtenidos y la conclusión final.

Capítulo 2

Fundamentos de la computación cuántica

Los comienzos de la computación cuántica vienen narrados por R. Feynmann, cuando a finales de los años 80 se dio cuenta de que la simulación de sistemas cuánticos no se podía llevar a cabo de manera eficiente y estable en ordenadores clásicos. A continuación, científicos como Deutsch, Bernstein, Vazirani y Yao centraron su trabajo en intentar establecer las bases teóricas de esta nueva tecnología, demostrando la existencia de máquinas de Turing cuánticas universales capaces de reproducir circuitos equivalentes a los clásicos [3]. Sin embargo, lo más interesante aún estaba por llegar, pues en los años sucesivos se fueron encontrando cada vez más ejemplos de algoritmos donde la computación cuántica ideal se constituía como la clara vencedora, alcanzando, en contadas ocasiones, una ventaja exponencial frente a su rival clásico.

En la primera sección de este capítulo se introducen los postulados de la mecánica cuántica (2.1), haciendo un especial énfasis en los conceptos que se necesitarán para entender el funcionamiento de la computación cuántica. A continuación, se introducirá el concepto de *qubit* y se tratará la construcción de circuitos mediante puertas lógicas cuánticas (2.2). Además, en esta misma sección se remarcará el papel de la superposición y el entrelazamiento. Por último, para terminar el capítulo, se expondrá una mínima indagación sobre las bases físicas de la computación cuántica (2.3).

2.1. Postulados de la mecánica cuántica

La mecánica cuántica se erige como un marco en el que comprender, matemáticamente, el límite inferior del universo. En este sentido, los cuatro postulados siguientes se encargan de establecer la conexión entre los sistemas físicos y el formalismo matemático de esta disciplina.

POSTULADO 1 *Para cada sistema físico aislado S existe un espacio vectorial complejo con producto interno, o espacio de Hilbert, conocido como el espacio de estados \mathcal{H}_S . El sistema queda descrito por un vector unitario en este espacio, denominado vector de estado o función de onda $|\psi\rangle$.*

Ahora bien, la mecánica cuántica no proporciona ni este espacio ni el vector de estado, para ello hay que recurrir a teorías particulares como *Quantum Electrodynamics* o *Quantum Chromodynamics*. El estado más simple que puede construirse en el marco de una teoría cuántica lo proporciona un *qubit*, y no es más que un espacio de estados de dos dimensiones. Una base ortonormal viene dada por los vectores $|0\rangle$ y $|1\rangle$, y un estado arbitrario queda representado por $|\psi\rangle = a|0\rangle + b|1\rangle$, con $a, b \in \mathbb{C}$ verificando $|a|^2 + |b|^2 = 1$. Asimismo, decimos que una combinación lineal de estados de un espacio de Hilbert es una superposición de esos estados, por lo que el estado $|\psi\rangle$ no es más que una superposición de los estados $|0\rangle$ y $|1\rangle$.

POSTULADO 2 *La evolución de un sistema cuántico cerrado está descrita por una transformación unitaria, es decir, $|\psi(t_2)\rangle = U|\psi(t_1)\rangle$, con U un operador unitario¹.*

Mientras que en este postulado se requiere que el sistema esté cerrado, en la realidad esto no es factible, pues todo sistema interactúa, aunque sea muy ligeramente, con otros.

Una descripción equivalente y más habitual del postulado 2 suele venir dada por:

POSTULADO 2' *La evolución temporal del estado de un sistema cuántico cerrado está descrito por la ecuación de Schrodinger:*

$$i\hbar \frac{d|\psi\rangle}{dt} = H|\psi\rangle$$

Vemos que para conocer la dinámica de un sistema será suficiente con conocer el hamiltoniano de éste, H , lo cual no es una tarea fácil. Por otro lado, como el hamiltoniano

¹Que un operador U sea unitario quiere decir que su operador inverso es igual a su adjunto, a saber, $UU^\dagger = U^\dagger U = \mathbb{I}$.

es un operador hermítico², admite una descomposición espectral con valores propios reales $H = \sum_E E|E\rangle\langle E|$, donde a los estados $|E\rangle$ se les suele denominar estados estacionarios, pues su evolución temporal viene dada por el factor $|E\rangle \rightarrow e^{-iEt/\hbar}|E\rangle$. Hay que notar que, al aplicar un operador unitario al sistema estamos interactuando con él, por lo que dejaría de ser un sistema cerrado. Para solventar este sinsentido es habitual introducir términos adicionales en H , que describan la evolución temporal del sistema, teniendo en cuenta aquellas interacciones con el exterior y otros parámetros que reflejen el control del experimento.

Con esta segunda interpretación somos capaces de determinar la forma que tiene el operador unitario del postulado 2 sin más que escribiendo la solución de la ecuación de Schrodinger (que puede encontrarse en [1]), llegando a:

$$U(t_2, t_1) = e^{-\frac{iH(t_2-t_1)}{\hbar}}$$

El siguiente postulado se encarga de englobar todas aquellas interacciones del sistema con el exterior que hacen que el vector de estado colapse en uno de los términos de su superposición:

POSTULADO 3 *Las medidas de un sistema cuántico están representadas por un conjunto de operadores de medida $\{M_m\}$ que verifican la relación de completitud $\sum_m M_m^\dagger M_m = I$, donde la probabilidad de medir m viene dada por $\mathbb{P}(m) = \langle \psi | M_m^\dagger M_m | \psi \rangle$. El estado en el que se deja al sistema viene dado por la siguiente función de onda:*

$$|\psi'\rangle = \frac{M_m|\psi\rangle}{\sqrt{\langle \psi | M_m^\dagger M_m | \psi \rangle}}$$

Hay que observar que, para el caso de un solo *qubit*, únicamente se tendrían los siguientes operadores de medida $M_0 = |0\rangle\langle 0|$ y $M_1 = |1\rangle\langle 1|$, pero los vectores $|0\rangle, |1\rangle$ pueden ser cualquier par de estados ortogonales del conjunto de todos los estados.

Una aplicación interesante del Postulado 3 está relacionada con el problema de distinguir estados cuánticos: cuando se tiene una colección de estados que no son ortonormales, no existe una medición que nos diga el estado particular que se ha obtenido.

POSTULADO 4 *El estado de un sistema físico compuesto es el producto tensorial de los estados que lo conforman. Este enunciado puede denotarse por: $|\psi_1 \dots \psi_n\rangle = |\psi_1\rangle \otimes \dots \otimes |\psi_n\rangle$*

²Un operador es hermítico si es igual a su adjunto.

Por ejemplo, en un sistema de 2 *qubits* se tendrán los siguientes estados ³:

$$\begin{aligned}
 |00\rangle &= \begin{bmatrix} 1 & \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 0 & \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} & |01\rangle &= \begin{bmatrix} 1 & \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0 & \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \\
 |10\rangle &= \begin{bmatrix} 0 & \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 1 & \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} & |11\rangle &= \begin{bmatrix} 0 & \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 1 & \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
 \end{aligned}$$

2.2. Conceptos básicos y notación

2.2.1. *Qubits*

La computación cuántica se concibe como una alternativa a la computación clásica en la que se ponen en uso las propiedades de superposición y entrelazamiento, características de los sistemas cuánticos. En este sentido, y como se mencionaba en la discusión del postulado 1, el bloque fundamental de la computación cuántica es el *qubits*, que a diferencia del bit clásico, puede estar en cualquier estado $|\psi\rangle = a|0\rangle + b|1\rangle$ con $a, b \in \mathbb{C}$ verificando la condición $|a|^2 + |b|^2 = 1$, necesaria para que la probabilidad de encontrar el sistema en un estado determinado no sea mayor que la unidad. Generalmente, un *qubit* suele representarse en el espacio de tres dimensiones como una esfera (ver la figura 2.1), denominada esfera de Bloch. La idea de representar cada *qubit* como una esfera tiene su fundamento en que es una forma de mostrar la superposición, con coeficientes complejos, de los estados $|0\rangle$ y $|1\rangle$, al mismo tiempo que se muestre la condición de normalización. Por esta razón es muy común realizar el cambio a coordenadas esféricas dado por $a \rightarrow \cos(\theta/2), b \rightarrow e^{i\varphi} \sin(\theta/2)$, con (θ, φ) los ángulos del vector de estado con cada uno de los ejes.

2.2.2. Puertas y circuitos cuánticas

Siguiéndose con la tradición clásica, en computación cuántica se trabaja con puertas lógicas, capaces de transformar el estado de los *qubits*. Naturalmente, como queremos que

³Esta forma de generar los estados de un sistema físico compuesto proporciona la base de estados denominada *base computacional*.

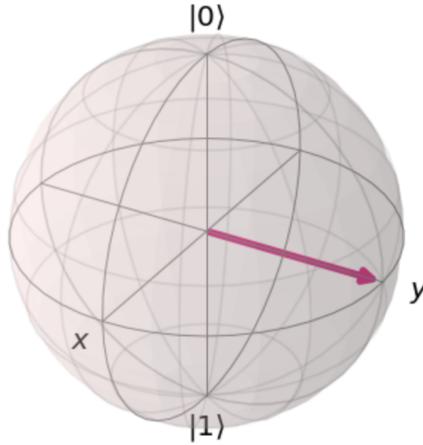


Figura 2.1: Representación del estado $|\psi\rangle = \frac{\sqrt{2}}{2}|0\rangle + \frac{\sqrt{2}}{2}e^{i\pi/2}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$ en la esfera de Bloch (Crédito: Qiskit [4])

se mantenga la condición de normalización, es necesario trabajar con puertas cuánticas que no modifiquen la longitud del vector de estado. A este tipo de puertas se las denomina unitarias, y es fácil demostrar que su determinante debe ser uno de los valores $\{-1, +1\}$. Con esto tenemos asegurado que todas las operaciones que se apliquen sobre los *qubits* de un circuito cuántico puedan entenderse como rotaciones sobre la esfera de Bloch, dejando invariante la amplitud de nuestro estado.

Algunas de las puertas cuánticas más importantes son las matrices de Pauli que, como veremos más adelante, son la base de las operaciones de rotación. Su representación matricial viene dada por las siguientes expresiones:

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad [\text{Eq. 2.1}]$$

También hay que destacar la puerta Hadamard, denotada por H , que transforma la base $\{|0\rangle, |1\rangle\}$ en la base de superposición $\{|+\rangle, |-\rangle\}$, la puerta fase, S , que gira $\pi/2$ radianes la componente $|1\rangle$ en el plano XY, y la puerta $\pi/8$, denotada como T , que introduce un desfase de $\pi/4$ entre las componentes $|0\rangle$ y $|1\rangle$ de un estado en superposición:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \quad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \quad [\text{Eq. 2.2}]$$

Cuando se llevan al exponente las matrices de Pauli, se obtienen los operadores de rotación

$$R_x(\theta) = e^{-i\theta\sigma_x/2}, \quad R_y(\theta) = e^{-i\theta\sigma_y/2}, \quad R_z(\theta) = e^{-i\theta\sigma_z/2}$$

consiguiendo la representación matricial mediante el desarrollo de Taylor. Sin pérdida de generalidad basta tomar $O = \sigma_x, \sigma_y, \sigma_z$ una puerta de Pauli cualquiera y tener en cuenta la

equivalencia $O^2 = I$. Así, es fácil verificar

$$e^{-i\alpha O} = \left(1 - \frac{1}{2!}\alpha^2 + \dots\right) \mathbb{I} - i \left(\alpha - \frac{1}{3!}\alpha^3 + \dots\right) O = \cos(\alpha)\mathbb{I} - i \sin(\alpha)O$$

sin más que haciendo uso del desarrollo en serie de potencias de las funciones seno y coseno.

Como se mencionaba anteriormente, al poseer las puertas unitarias un determinante con valor absoluto la unidad, podemos entenderlas como rotaciones en la esfera de Bloch:

TEOREMA 2.1 (*Descomposición en rotaciones*) *Sea U un operador unitario que actúa sobre un único qubit, entonces existen $\alpha, \beta, \gamma, \delta \in \mathbb{R}$ tales que $U = e^{i\alpha}R_z(\beta)R_y(\gamma)R_z(\delta)$*

Las operaciones que dependen del valor de un cierto *qubits* se denominan operaciones controladas, y son la base del entrelazamiento en un circuito cuántico. Entre ellas destaca la puerta *CNOT*, también llamada *CX*, que toma como entrada un *qubits* de control y un *qubits* objetivo, cambiando el valor de este último en caso de que el de control esté en el estado $|1\rangle$. Esta acción se puede escribir como $|c\rangle|d\rangle \rightarrow |c\rangle|c \oplus d\rangle$ o matricialmente como

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad [\text{Eq. 2.3}]$$

y se considera la segunda puerta cuántica más importante, después de la puerta *H*. Esta jerarquía en la importancia de las puertas puede ser entendida si pensamos en el uso que se les va a dar en un circuito cuántico. Mientras que la puerta Hadamard tiene por objetivo crear o destruir superposiciones, la puerta *CNOT* se establece como la puerta más simple capaz de crear o destruir entrelazamientos, siendo la introducción de esta interferencia controlada el pilar central de la computación cuántica.

De manera general, puede definirse la operación controlada de cualquier matriz unitaria U , denotándose por CU , como aquella que transforma una función de onda de esta forma $|c\rangle|d\rangle \rightarrow |c\rangle U^c |d\rangle$. Y, como puede verse en las páginas 180-184 de [1], tenemos asegurada la implementación de estas operaciones en un circuito cuántico.

Un caso que hay que destacar es la puerta Toffoli. Se trata de una particularización de C^2U , y se obtiene tomando $U = X$, por lo que cambia el estado del tercer *qubits* únicamente cuando los otros dos están en la posición $|1\rangle$.

Al introducir los circuitos cuánticos, es inevitable dedicar un espacio a la universalidad, la cual hace referencia a la capacidad de representar con precisión arbitraria cualquier operación unitaria con un conjunto finito de puertas cuánticas (p.189 [1]). En este sentido, veremos los dos resultados más importantes sobre la universalidad de los circuitos cuánticos:

TEOREMA 2.2 (*Universalidad 1*) *Cualquier operación unitaria en un espacio de Hilbert de n qubits puede descomponerse en puertas de un solo qubits y puertas CNOT.*

A pesar de que en la práctica este resultado no tiene mucho interés porque es ineficiente, unido al resultado siguiente nos proporciona una visión prometedora sobre la versatilidad de la computación cuántica.

TEOREMA 2.3 (*Universalidad 2*) *Cualquier operación de un solo qubits se puede aproximar por puertas H , P y $\pi/8$ a precisión arbitraria.*

2.2.3. Superposición y entrelazamiento

Hasta ahora, se ha expuesto que la superposición de los *qubits* es beneficiosa para la computación cuántica, pues la convierte en una alternativa potencialmente más competitiva que la computación clásica, pero no se ha explicado con claridad el papel que juega el entrelazamiento en todo ello. En este apartado, se intentará explicar con un ejemplo la base del entrelazamiento, su interés y las formas de medirlo.

Empecemos viendo un ejemplo habitual. Supongamos que tenemos el estado:

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Según el postulado 4, se trata de uno de los estados del sistema compuesto de 2 *qubits*, siendo en particular una superposición de dos estados de su base computacional, los estados $|00\rangle$ y $|11\rangle$. Ahora, notemos que no existen $|\psi_1\rangle$ y $|\psi_2\rangle$ tal que $|\psi\rangle = |\psi_1\rangle|\psi_2\rangle$, es decir, no podemos trabajar de manera separada sobre cada uno de los dos *qubits*: si medimos uno de ellos, el otro colapsará al mismo valor que el primero. Un estado de este estilo se dice que es entrelazado, y nos permite obtener información sobre un *qubit* sin hacer ninguna medición directa sobre él.

La superposición particular que se ha tomado para este ejemplo, $|\psi\rangle$, forma parte de un conjunto de estados que poseen el máximo entrelazamiento posible, denominados estados de Bell.

2.3. Implementación de los ordenadores cuánticos

2.3.1. Condiciones para su existencia

Desde una perspectiva práctica, uno puede preguntarse si existen dispositivos capaces de implementar los circuitos cuánticos o si se trata de mera especulación. El núcleo de esta sección consistirá en tratar de dilucidar esta cuestión.

Inicialmente, se verán cuáles son los requerimientos para conseguir un sistema cuántico capaz de satisfacer nuestros propósitos, dejando para un último punto algunas de las alternativas más alentadoras.

Naturalmente, para sacar partido de un computador cuántico se requiere de un número suficiente de *qubits* cuyas funciones de onda estén bien caracterizadas. Con bien caracterizadas no hago referencia a que no dependan unas de las otras, precisamente se busca que el sistema de *qubits* esté superpuesto, sino que cada uno de ellos pueda escribirse como la combinación compleja de los estados $|0\rangle$ y $|1\rangle$, sin que aparezcan inercias o impulsos indeseados hacia un estado o el otro. Hay que notar que, quitando los casos similares a los spines de los átomos, es difícil encontrar en la naturaleza un sistema cuántico con dos niveles, pues, generalmente, se dispondrá de un sistema cuantizado con infinitos niveles de energía.

En segundo lugar, es esencial poder inicializar todos los *qubits* en el estado deseado. Normalmente este estado será el de más baja energía, a saber, el $|0, 0, 0, 0, \dots\rangle$, y la precisión de esta configuración determinará cuán espuria es la función de onda en las puertas cuánticas venideras. Los protocolos de corrección de errores también se verán afectados si no se consigue una fuente inicial lo suficientemente uniforme. En la teoría, se presentan dos alternativas lógicas para alcanzar el estado de menor energía: dejar que el sistema no reciba energía externa y llegue a su mínimo de forma natural, o mediante medidas sucesivas que fuercen a los *qubits* a estar en el estado ideal o su opuesto, pudiéndose rotar este último.

La decoherencia ⁴ también sería una pieza angular para poder computar con estos sistemas, pues se la considera la principal fuente de supresión cuántica. Por este hecho, los tiempos de decoherencia deben ser lo más largos posibles, lo suficiente como para poder aplicar varias puertas cuánticas a un mismo sistema. Esto pone sobre la mesa la cuestión de si se pueden conseguir ventajas cuánticas o si, por el contrario, la decoherencia va a disminuir de forma proporcional al tiempo de computación, poniendo trabas inquebrantables a la via-

⁴Entiéndase aquí por decoherencia al opuesto de coherencia, siendo ésta la propiedad de un sistema por la cual diferentes ondas poseen fases relativas constantes, de forma que puedan interferir destructiva y constructivamente.

bilidad del circuito cuántico. Por lo que sabemos, existen protocolos de corrección de errores que mejoran la decoherencia y que son capaces de limitar el ruido de un sistema cuántico, aunque será la tecnología seleccionada la que más tenga que decir sobre este respecto.

Actualmente se sabe que la corrección de errores es viable mediante operaciones cuánticas y que los propios errores introducidos en la corrección, si no son muy frecuentes, no conllevarán una proliferación de los mismos. Esto es, la escalabilidad de un circuito cuántico equipado con corrección de errores resulta prometedora. Las limitaciones que se han encontrado en estos estudios establecen que es necesario una decoherencia de $10^4 - 10^5$ veces el tiempo de aplicación de una puerta cuántica ([5], [6]), con lo que, una vez superado este límite, la decoherencia no será un obstáculo para la escalabilidad.

Otro punto notable es la capacidad de trabajar con una base de puertas lógicas que sea universal. Mientras que ya se ha visto en los resultados 2.2 y 2.3 que sobre el papel se puede implementar cualquier operación unitaria en un conjunto de *qubits*, físicamente se requiere que la interacción sea instantánea. Empecemos viendo un esquema sencillo sobre un circuito cuántico para poder entender la importancia práctica de la universalidad. Un algoritmo cuántico, a fin de cuentas, no es otra cosa que una sucesión de puertas unitarias U_1, U_2, U_3 , actuando cada una de ellas sobre una pequeña fracción de los *qubits*, generalmente 1 o 2. Físicamente, estas puertas se traducen en las transformaciones que inducen los hamiltonianos H_1, H_2, H_3 de la forma $U_i = e^{iH_i t/\hbar}$. Tiene sentido querer exigir que los hamiltonianos se apliquen de forma consecutiva: el primero entre $[0, t]$, el segundo $[t, 2t]$, etc. Pero, ¿qué quiere decir que una interacción de un hamiltoniano se pueda iniciar o desactivar? La naturaleza no funciona como los interruptores, físicamente habrá de ejercer una serie de iteraciones que cambien las intensidades de las interacciones en cuestión.

Por otro lado, también se presenta la dificultad de simular un sistema cuántico con unas magnitudes clásicas que podamos modificar. Véase la intensidad de un láser o la tensión entre dos electrodos, que en general serán gobernados por fenómenos cuánticos. Cuando se establece la dependencia del sistema cuántico con un controlador clásico, se han de contemplar posibles interacciones cuánticas entre los dos aparatos, aunque en general puedan resultar despreciables. Un ejemplo de esta dependencia puede verse en aquellos ordenadores cuánticos donde no exista una interacción a pares, como es el caso de la trampa de iones. En este sistema, es necesario introducir un subsistema cuántico que permita a las parejas de *qubits* interactuar, lo cual se consigue con la vibración de los iones, y siendo todo ello muy sensible a verse influenciado con el entorno. ¿En qué se traduce todo ello? Efectivamente, en decoherencia. En ejemplos como éste, la decoherencia, creciendo con el número de puertas

cuánticas, se ve drásticamente perjudicada.

En definitiva, las puertas cuánticas no pueden implementarse sin introducir errores, por muy pequeños que sean, produciendo un efecto dominó en el esquema decoherencia-corrección de errores. La naturaleza de estos errores tendrá una componente puramente aleatoria y otra sistemática, siendo esta última la que definirá la tecnología que se alce vencedora.

Por último, cabe mencionar la capacidad que tenemos sobre la medida de los *qubits*. Idealmente se requiere que la medida de un *qubits* no perturbe a sus vecinos y que el estado en el que permanezca el *qubit* en cuestión, tras la medida, sea el propio de la amplitud de probabilidad que se ha medido. Sin embargo, esto no suele darse de forma exacta en la realidad... aunque puede que tampoco sea necesario. Veamos un ejemplo para clarificar este punto: si se posee una eficiencia del 90 % sobre el resultado de un circuito cuántico (a saber, tenemos una seguridad de que 9 de cada 10 medidas sean correctas), basta crear varios estados de salida para, estadísticamente, llegar a una solución correcta. A expensas de repetir medidas con seguridad del 90 %, uno puede acercarse todo lo que desee al 100 %. Esta ventaja recae en poder medir un sistema cuántico con la suficiente rapidez, del orden de 10^{-4} veces el tiempo de coherencia, quitando peso a los protocolos activos de corrección de errores. Aún así, a día de hoy los *qubits* se canjean caros, se tendrá que esperar a una época en la que este bien tanpreciado sea más abundante.

En cuanto a las tecnologías más prometedoras, podemos ver algunas de ellas agrupadas en la tabla 2.1 en función de sus tiempos de decoherencia y de operación:

Tecnología	τ_Q/s	τ_{op}/s	n_{op}
Spin nuclear	$10^{-2} - 10^8$	$10^{-3} - 10^{-6}$	$10^5 - 10^{14}$
Spin electrón	10^{-3}	10^{-7}	10^4
Ion trap (In^+)	10^{-1}	10^{-14}	10^{13}
Electrón - Au	10^{-8}	10^{-14}	10^6
Electrón - GaAs	10^{-10}	10^{-13}	10^3
Quantum dot	10^{-6}	10^{-9}	10^3
Cavidad óptica	10^{-5}	10^{-14}	10^9
Cavidad microondas	10^0	10^{-4}	10^4

Cuadro 2.1: Estimaciones para los tiempos de decoherencia τ_Q , tiempos de operaciones τ_{op} y máximo número de operaciones, definido como el cociente τ_Q/τ_{op} , para varias tecnologías cuánticas posibles. Las tecnologías Spin nuclear, Ion trap y Spin electrón utilizan como qubit los spines del núcleo y del los electrones, los Electrón-Au, Electrón-GaAs y Quantum Dot emplean la presencia o la ausencia de un electrón en una región confinada, y Cavidad óptica y Cavidad microondas usan los fotones en dos modos diferentes de un pozo de potencial.



Figura 2.2: IBM 5Q System. Criostato de un ordenador de IBM con 50 qubitss superconductores.
(Crédito: IBMq)

También se han de destacar aquellas que vienen de la mano de los *qubits* superconductores, pues aseguran pocas pérdidas disipativas, bajas temperaturas y no linealidad ([7]). Aun así, poseen una dificultad intrínseca en la escalabilidad, pues es técnicamente complejo aumentar el número de *qubits* a la par que se mantenga un sistema ideal donde puedan utilizarse sus propiedades cuánticas, siendo uno de los pilares fundamentales la capacidad de unir cada uno de ellos con los demás. Es justamente esta vía la que está tomando IBM (2.2), cuyo software utilizaremos para llevar a cabo las simulaciones de nuestro algoritmo.

2.3.2. Mitigación de errores

Al ser la computación cuántica una herramienta probabilística muy sensible a alteraciones tanto internas como externas, los errores que se comenten en sus cálculos son algo inherente a esta ciencia. En este sentido, se han desarrollado todo un conjunto de códigos que se encargan de corregir los distintos tipos de errores, algunos de estos pueden verse en el capítulo 7 de [8] o en artículos como [9], nosotros sólo discutiremos las principales diferencias con la computación clásica.

En la teoría clásica de la corrección de errores, la idea principal es introducir información redundante a los mensajes de tal forma que podamos utilizarla para comprobar si se ha cometido un error al recibir el mensaje. Sin embargo, en el caso de la computación cuántica hay tres resultados que hacen que esto sea algo distinto: los errores provenientes de fuentes muy diversas, enraizados en la naturaleza cuántica; la destrucción de información mediante las medidas; y las consecuencias del teorema de no clonado. A falta de más espacio, abordaremos únicamente este último punto:

TEOREMA 2.4 (*no clonado*) *Es imposible hacer una copia de un estado cuántico desconocido.*

Naturalmente, el teorema pone de manifiesto la inviabilidad de los códigos de repetición clásicos en una computadora cuántica. Veamos hasta que punto se produce un cambio con respecto a la computación clásica mediante un ejemplo. Supongamos que tenemos inicialmente un bit que queremos copiar en un *qubit*. Para ello, basta hacer uso de la puerta al estilo *CNOT*, pues, como ya se ha visto, es una cuestión de inicializar el *qubit* en cero y sumarle el valor del bit. Ahora, si nos ponemos en el caso cuántico y tratamos de copiar el estado $|\psi\rangle = a|0\rangle + b|1\rangle$, la puerta *CNOT* nos devolverá un estado que transforme

$$(a|0\rangle + b|1\rangle) |0\rangle = a|00\rangle + b|10\rangle$$

en el estado

$$a|00\rangle + b|11\rangle$$

pues se ha de negar el valor del segundo *qubit* en el único caso de que el primero valga 1.

Nos preguntamos en este punto, ¿se ha obtenido la copia deseada? Está claro que si a o b son nulos la copia se revela exitosa, es decir, cuando son simplemente un bit. La diferencia estriba en el caso de tener verdaderamente una superposición de estados, y es fácil comprobar que no se ha alcanzado el producto tensorial deseado:

$$|\psi\rangle|\psi\rangle = a^2|00\rangle + b^2|11\rangle + ab|01\rangle + ba|10\rangle$$

Aquel interesado puede encontrar la demostración del teorema de no clonado en la página 532 del libro [1].

Nótese también que este esquema puede contemplarse desde el punto de vista de los postulados de la Mecánica Cuántica. Si después de la primera medida pudiésemos obtener información sobre la superposición del otro *qubit*, estaríamos adentrándonos en la información perdida tras el colapso de la medición.

Capítulo 3

Complejidad computacional y utilidad cuántica

Históricamente, los hitos más destacables en el ámbito de la lógica cuántica vienen dados por los siguientes acontecimientos: en 1985 Deutsch definió un posible modelo de computación cuántica que denominó *Turing Quantum Machine*, más adelante, en 1993, Bernstein y Varizarni probaron que existe una máquina de este estilo eficiente. Ese mismo año, Yao extendió estos resultados al mostrar que las *Turing Quantum Machines* tienen un equivalente polinómico en circuitos cuánticos, que fueron propuestos por Deutsch en 1989. Todos estos avances teóricos, y sus subsiguientes, no consiguieron correr parejos a sus implementaciones reales, dando como resultado la instauración de una nueva época de la computación cuántica, aquella en la que nos encontramos actualmente.

En este capítulo se abordará la aplicabilidad de la computación cuántica en la actualidad, partiendo desde la teoría de la complejidad computacional, que se estudiará con brevedad en la primera sección (3.1). A continuación, se introducirán los algoritmos híbridos (3.2), donde se comprenderá la relevancia de la época NISQ (3.2.1). Asimismo, se asentarán las bases del algoritmo cuántico que se ha empleado para el proceso de reconstrucción de trazas, el algoritmo VQE (3.2.2).

3.1. Desde la perspectiva de la complejidad computacional

Para ver dónde reside la computación cuántica en el marco de las clases de complejidad, veamos una breve clasificación de la dificultad de un problema en función de su tamaño, introduciendo las nuevas clases que ha conseguido desbloquear esta nueva disciplina:

- Se recogen en P aquellos problemas que pueden resolverse en tiempo polinomial con respecto a la talla del input en un ordenador clásico.
- NP está definida como aquella clase donde evaluar una posible solución pertenece a P, pero sin tener por qué estar en esa clase el encontrar la solución.
- BPP (*Bounded-error Probabilistic Polynomial-time*) es la clase de problemas que se resuelve en tiempo polinomial mediante un ordenador probabilista.
- En contraposición a esta clase, los problemas BQP (*Bounded-error Quantum Polynomial-time*) son aquellos que se resuelven en tiempo polinomial por un ordenador cuántico.
- Por último, la clase QMA (*Quantum Merlin Arthur*) es el equivalente cuántico a la clase NP, esto es, la clase que recoge problemas donde la evaluación de una solución puede comprobarse en tiempo polinomial por un ordenador cuántico.

Es importante destacar que dentro de las clases NP y QMA suele hacerse la distinción entre los problema difíciles y los más sencillos, apostillando en el primer caso la etiqueta *Complete*. Justamente son estos problemas los que generan un mayor interés. El ejemplo más conocido es el de factorización de enteros, que se puede resolver en tiempo polinomial mediante computación cuántica. Sin embargo, se ha de destacar que no se espera que esta tecnología pueda resolver cualquier problema NP-*Complete* en tiempo polinomial.

Naturalmente, la comparación más evidente con los problemas que se pueden resolver polinómicamente en una máquina de Turing cuántica son los BPP. Bernstein y Vazirani probaron la inclusión $BPP \subset BQP$, esto es, todos los problemas que se puedan resolver clásicamente en tiempo polinomial por una máquina clásica probabilística también lo harán en un ordenador cuántico. Es más, dieron la primera evidencia en 1993 de que no se tiene la igualdad, a saber, las máquinas de Turing aplicadas a problemas que requieren de tiempo polinómico son más potentes que aquellas que sólo hacen uso de las maquinas de Turing probabilistas. De este modo, y al ser los BPP la clase de problemas que se pueden computar eficientemente, con

esta prueba quedo demostrado que los ordenadores cuánticos podrían convertirse en una herramienta significativamente más potentes para ciertos problemas específicos. Más adelante, Simon reforzó la evidencia probando la existencia de un oráculo ¹ donde los problemas BQP no pueden simularse por máquinas probabilistas por debajo de los $2^{n/2}$ pasos. El siguiente avance decisivo fue debido a Shor, que propuso en 1994 un algoritmo cuántico capaz de factorizar en primos un número entero en tiempo polinómico, alzando la cuestión $NP \subset BQP$. Si se diera esta inclusión, se podrían resolver problemas NP-Complete, como el *Travelling salesman problem*, en tiempo polinomial. Mientras que esta demostración no ha podido darse aún, sí que se ha llegado a que no puede resolverse con un método cuántico basado en los oráculos [3].

Aun así, esta incipiente supremacía cuántica no debería sorprender a nadie, mientras que la versatilidad ofrecida por los algoritmos probabilistas puede ser útil desde el punto de vista estadístico para manejar sistemas demasiado complejos, la mecánica cuántica da un paso más allá y proporciona las ventajas del mundo del azar junto con el impulso de la interferencia. Esto es, lo sustantivo recae en la capacidad de interferir constructiva y destructivamente para dar con el resultado buscado. En este punto es fácil caer en la tentación de creer que, al ser el número de estados creciente de manera exponencial con el número de *qubits* (de la forma 2^n) la ventaja que se obtendrá será exponencial. Nada más alejado de la realidad. Una idea de por qué esto no es así lo vamos a ver con el siguiente ejemplo:

Pongámonos en el caso de que estamos en una lista de tamaño N , que tomaremos por simplicidad como potencia de 2, $N = 2^n$. Clásicamente, para encontrar un elemento a la primera en esta larga lista tendríamos la probabilidad $1/N$. Sencillo. Poniéndonos en el caso de una superposición cuántica, si utilizáramos la misma estrategia nos veríamos condenados a seleccionar también un estado de entre 2^n . ¿Dónde queda la ventaja de la superposición? Pues aquí reside la estrategia/algoritmo que se quiera usar, cuya pieza central será un oráculo. Con este oráculo seremos capaces de explotar las verdaderas ventajas cuánticas, haciendo malabares con las probabilidades que nos interesen. La notación que se va a usar es la siguiente, denotaremos a la superposición de estados como $|\psi_0\rangle$ y a la solución buscada $|y\rangle$, ortogonal al resto.

$$|\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_x |x\rangle$$

¹En el sentido cuántico, un oráculo es un conjunto de operaciones cuánticas que permiten diferenciar una función de onda solución del resto de estados, generalmente mediante la introducción de un desfase con respecto al resto de estados de una superposición. La versatilidad que ofrecen los oráculos, sin parangón en la computación clásica, los hacen verse como el más eminente distintivo de los circuitos cuánticos.

El oráculo adecuado será capaz de introducir un desfase entre $|y\rangle$ y el resto de $|x\rangle$, de forma que será posible llegar a una función de onda:

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \neq y} |x\rangle - \frac{1}{\sqrt{2^n}} |y\rangle = |\psi_0\rangle - \frac{2}{\sqrt{2^n}} |y\rangle$$

Con una simple comprobación de la probabilidad encontramos:

$$\mathbb{P}(y) = |\langle y | \psi_1 - \psi_0 \rangle|^2 = \frac{4}{2^n}$$

Y esta ventaja cuadrática es efecto único de la primera iteración, una vez marcado como singular el término $|y\rangle$ de la superposición, es cuestión de desplazar los pesos de los términos base hacia la solución buscada. La forma en la que se vaya separando este estado del resto lo llevará a cabo una puerta que introduzca diferencia de amplitud allá donde exista diferencia de fase, y esto puede hacerse, por ejemplo, con el operador de Grover [10]. La velocidad a la que este operador hará que se aproxime la solución será cuadrática con el número de iteraciones, aproximadamente $4k^2/2^n$ para k pequeño. Es más, se ha probado en [3] que este proceso es óptimo, ningún algoritmo cuántico devolverá una probabilidad más alta que la cuadrática en k .

3.2. Introducción a los algoritmos híbridos

En los últimos años, gran parte de los científicos que investigan los límites de la computación cuántica han estado desarrollando algoritmos que sean capaces de beneficiarse de las propiedades cuánticas con los ordenadores imperfectos que se tienen en la actualidad. Los altos niveles de ruido, unidos a la naturaleza probabilística, hacen que los algoritmos cuánticos más famosos (como el de Grover o el de Shor) sean impracticables en la dimensión a la que superan a sus competidores clásicos. Por esta razón, se ha progresado en dirección a los algoritmos variacionales, que emplean la superposición y el entrelazamiento de forma iterativa, y donde un método clásico de optimización sea capaz de converger a la solución.

Desde un punto de vista computacional se está probando si se pueden aplicar estos circuitos híbridos para resolver problemas NP-*Complete*, como es el caso de los problemas de combinatoria. En particular, el algoritmo que se va a estudiar se ha probado que es eficiente para resolver problemas de optimización de tipo QUBO (*Quadratic Unconstrained Binary Problems*), y, como se verá en el capítulo siguiente, esta es la forma en la que se ha planteado el problema de la reconstrucción de trazas a partir de *hits*.

3.2.1. En la época NISQ

Actualmente, en la computación cuántica experimental se está intentando conseguir un conjunto de *qubits* que puedan controlarse por puertas cuánticas al mismo tiempo que se mantiene una coherencia lo suficientemente prolongada. Para que este proceso, en apariencia contradictorio, salga adelante, mucho esfuerzo se está destinando a la corrección de errores. Y es que aún no se puede decir que exista un ordenador cuántico que utilice la corrección de errores y se sostenga como una herramienta útil. En este presente incierto, en el que quizás deban pasar décadas hasta que un ordenador cuántico con puertas lógicas tenga la suficiente potencia como para ser relevante, se han ido confeccionado algoritmos que sean manejables en los ordenadores cuánticos ya presentes, los denominados dispositivos *Noisy Intermediate-Scale Quantum* (NISQ). Así, se está apostando por explotar al máximo las posibilidades de ordenadores cuánticos imperfectos y por ver cómo se pueden integrar en nuestros ordenadores clásicos. Estos algoritmos se recogen bajo el término *Variational Quantum Algorithms* (VQA), siendo el más famoso el *Variational Quantum Eigensolver* (VQE).

Más en detalle, los elementos que construyen un VQA pueden englobarse en los siguientes grupos:

Función de coste

Cuánticamente, esta función queda codificada por medio de un hamiltoniano, de tal forma que una medida de sus autovalores propine los elementos para la minimización. En un algoritmo variacional los autovalores dependerán de ciertos parámetros, que denotamos por el vector θ , de forma que la minimización pueda llevarse en un computador clásico en función de estos valores:

$$\min_{\theta} f(\theta, \langle \psi(\theta) | H | \psi(\theta) \rangle)$$

La función de coste no será una función de la cual sepamos demasiada información, por ejemplo, si tuviéramos propiedades de regularidad podríamos hallar el mínimo por otros métodos.

Una familia de problemas que pueden resolverse con un VQA son aquellos cuya función de coste pueda transformarse en un hamiltoniano formado por una combinación lineal de productos de matrices de Pauli [11]. Así, se podrá escribir

$$H = \sum c_k \hat{P}_k$$

donde c_k es un coeficiente complejo y P_k es producto de los elementos $\{I, \sigma_x, \sigma_y, \sigma_z\}$. Más adelante se verá cómo pueden transformarse los problemas QUBO en hamiltonianos de esta forma.

Circuito cuántico parametrizado

El siguiente miembro de un circuito cuántico es el circuito parametrizado, donde se utilizan como input los ángulos θ y se obtiene un posible estado solución. El circuito cuántico parametrizado se aplica sobre el estado inicial $|\psi^{(0)}\rangle$, siendo habitual tomar $|0\rangle^{\otimes n}$

En términos de puertas unitarias, podemos pensar en el circuito parametrizado como un conjunto de puertas que transforman el estado inicial en una función de onda del tipo siguiente

$$|\psi(\theta, \phi)\rangle = U(\theta)|\psi^{(0)}\rangle = U(\theta)P(\phi)|0\rangle^{\otimes n} = U(\theta, \phi)|0\rangle^{\otimes n}$$

El término $P(\phi)$, la preparación del estado inicial, suele ser simplemente una superposición de puertas de Hadamard, pues de este modo se gana la mayor ventaja cuántica: con una sola iteración estamos descartando el mayor número de estados no deseados. No obstante, si se posee información sobre la solución puede ser buena idea preparar el estado inicial de tal manera que los estados estén cerca, en términos de θ , de la solución real.

Una buena elección del circuito parametrizado determinará cuán rápido converge el método y cómo de buena será la aproximación final. Habrá de sopesar la información que quiera aplicarse en el circuito con la longitud del mismo, pues a mayor número de puertas lógicas mayores serán los errores introducidos.

Proceso de optimización

Los criterios que se siguen para dilucidar si un método de optimización clásico se puede aplicar en un VQA se pueden resumir en los siguientes puntos:

- Debido a los tiempos de coherencia cortos, no pueden ser implementados métodos del gradiente analíticos.
- Se favorecerá que se produzca el menor número de evaluaciones de función, pues las medidas tardan un tiempo considerable.
- El método que se utilice debe ser lo suficientemente flexible como para que los datos inexactos de los resultados cuánticos no estropeen las medidas.

Estos requerimientos tan exigentes harán emerger algunos métodos y permitirán descartar otros, por muy eficientes que puedan ser en computación clásica.

Una primera familia de métodos de optimización, la más popular, son los denominados métodos del gradiente. Veamos por qué esta alternativa puede no ser idónea para un algoritmo híbrido. Como el gradiente de una función multivariable nos indica la dirección de mayor cambio, muchos métodos clásicos hacen uso del gradiente para acercarse a la solución de manera iterativa. De forma que, dado un paso ρ , se pueda escribir

$$\theta^{(k+1)} = \theta^{(k)} + \rho \nabla f(\theta^{(k)})$$

Evidentemente, en el caso cuántico esta derivación debe darse de forma aproximada, por ejemplo, mediante diferencias finitas, haciendo que

$$\partial_i f(\theta^{(k)}) = \partial_i \langle 0 | U^\dagger H U | 0 \rangle$$

pase a ser

$$\frac{f(\theta^{(k)} + \epsilon e_i) - f(\theta^{(k)} - \epsilon e_i)}{2\epsilon}$$

Ha de notarse que, para conseguir una buena estimación del gradiente, el parámetro ϵ debe hacerse lo más pequeño posible, incrementando así el número de evaluaciones de función f . En definitiva, un proceso muy costoso. Otras alternativas han sido propuestas, como el *Quantum Natural Gradient* o el *Quantum Analytic Descent* [11].

En el presente trabajo se utilizará un método que no requiere del gradiente, el algoritmo optimizador COBYLA. El método COBYLA se centra en la aproximación de un problema de optimización del tipo

$$\begin{cases} \text{mín } F(x), & x \in \mathbb{R}^n \\ c_i(x) \geq 0, & i = 1, 2, \dots, m \end{cases} \quad [\text{Eq. 3.1}]$$

por otro donde las funciones $F(x)$ y $c_i(x)$ sean estimaciones lineales de las dadas. El proceso que se sigue para obtener la linearización consiste en tomar un conjunto de puntos, llamados vértices, $\{x^{(j)} : j = 0, 1, \dots, n\}$, e interpolar tanto la función a minimizar como las restricciones. La justificación para utilizar este método se encuentra en [12], donde se concluye que, frente a otros optimizadores que sólo requieren de evaluaciones de función, es el que más rápido converge. Finalmente, comentar que el funcionamiento de este optimizador puede verse en el artículo [13], donde se introdujo este método.

3.2.2. El algoritmo VQE

En esta sección nos centraremos en explicar el método que se va a utilizar para la reconstrucción de trazas, el *Variational Quantum Eigensolver*, cuyo esquema se encuentra en la figura 3.1:

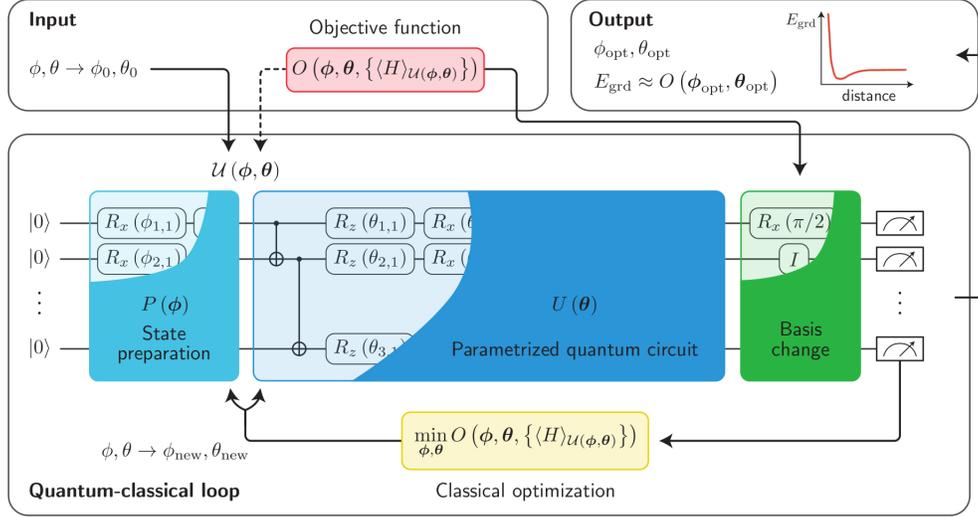


Figura 3.1: Esquema general del algoritmo VQE donde se resuelve el problema de la configuración más estable de una molécula. Notar que nosotros hemos tomado la notación $\theta_{a0} = \phi$ para las rotaciones del estado inicial. Crédito: [11]

La familia de funciones que se van a utilizar se denominan QUBO y pueden escribirse de la forma siguiente:

$$f(s) = \sum_i \sum_j w_{i,j} s_i s_j = s^T W s$$

estando definidas sobre los vectores binarios $s \in \{0, 1\}^n$ y siendo W una matriz simétrica.

La desigualdad variacional que define el problema de optimización es la siguiente

$$\lambda_{min} \leq E(\theta) = \langle \psi(\theta) | H | \psi(\theta) \rangle$$

por lo que, idealmente, es posible conseguir una buena estimación de aquel vector θ_{min} que minimice el valor esperado del hamiltoniano.

Asimismo, la decisión de trabajar con una función tipo QUBO hace posible que se pueda utilizar la correspondencia QUBO-Ising [14], que traduce la búsqueda del mínimo de f en la búsqueda del autovalor más pequeño de la matriz H , mediante el uso de la transformación

$$s_i \rightarrow \frac{1}{2}(1 + \sigma_i^z)$$

es decir, no es más que el cambio de las coordenadas binarias en spines cuánticos. Además, como se consigue representar el hamiltoniano como la suma de productos de observables que conmutan, basta con medir los *qubits* en su estado final para hallar el vector $s \in \{0, 1\}^n$ y un coste asociado.

En cuanto a esto último, dada la naturaleza cuántica del circuito de evaluación, será necesario medir los *qubits* un número de veces lo suficientemente elevado como para reducir los errores aleatorios, obteniéndose así un coste promedio para cada vector de ángulos:

$$E(\theta^{(k)}) = \frac{1}{m} \sum_{l=1}^m E_l$$

donde $E_l = f(s_l^{(k)}) \cdot Ab_l$ es el coste ponderado de cada estado resultante² y m representa el número de *shots* (veces que se ha ejecutado el circuito paramétrico). Una alternativa a este coste promedio entre todos los estados resultantes viene dado por el *Conditional Value at Risk*, donde la energía de la iteración k depende de la distribución de energías $P(E, \theta)$, y donde se toma la fracción $\rho \in (0, 1)$ de estados más pequeños:

$$E(\theta^{(k)}) = \int_{-\infty}^{\infty} P(E, \theta) E dE \approx \frac{1}{\rho m} \sum_{l=1}^{\rho m} \hat{E}_l$$

Este método puede adaptarse para trabajar únicamente con las energías más bajas y así concentrar la configuración del circuito cuántico en aquellos estados que generan una energía más cercana a la mínima [15].

La confección de un circuito cuántico que prepare un estado iterable se centra en el uso de puertas cuánticas de rotación, acompañadas por las correspondientes puertas de control, que serán las encargadas de introducir las dependencias de unos estados con otros. Así, cuando se tiene inicialmente el estado $|0\rangle$ de n *qubits*, la función de onda resultante tras aplicar el circuito será

$$|\psi(\theta^{(k)})\rangle = U(\theta^{(k)})|0\rangle$$

siendo U una matriz unitaria que se encarga de rotar cada *qubit* la cantidad que ha calculado el algoritmo de optimización clásico, y que puede escribirse como

$$U(\theta) = \prod_{l=1}^L \left(\prod_{a=1}^n i e^{i\theta_{al}\sigma_a^y} U_{ent} \right) \prod_{a=1}^n e^{i\theta_{a0}\sigma_a^y}$$

siendo U_{ent} cada etapa de rotación y superposición.

²Pues en cada una de las medidas l del circuito cuántico se obtendrá una función de onda s_l y una abundancia Ab_l .

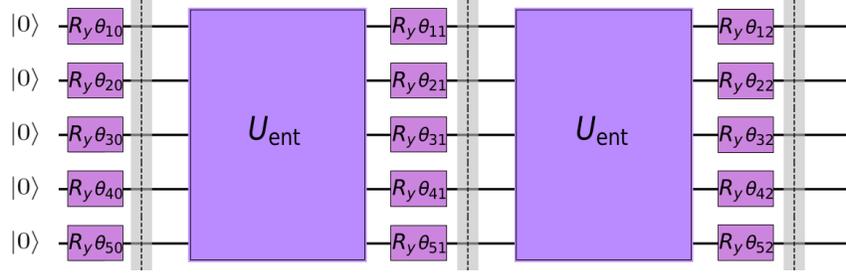


Figura 3.2: Ejemplo de un circuito parametrizado de 5 qubits con dos capas ($L=2$) para el algoritmo VQE. Crédito: [16]

El conjunto de todos los ángulos de rotación $\theta = (\theta_{ab})$ comprende a todos los parámetros que se han de optimizar, siendo la entidad variable en todo el proceso. En cuanto a la puerta unitaria encargada del entrelazamiento, el algoritmo deja total libertad a la hora de seleccionar su estructura, siendo posible tanto el entrelazamiento lineal, el entrelazamiento definido por el problema en cuestión y el dado de forma aleatoria. De este modo, si se quiere una estructura lineal bastaría con aplicar puertas CNOT sobre dos *qubits* consecutivos, quedando la puerta unitaria

$$U_{ent} = \prod_{j=1}^n e^{i\pi/4} (\mathbb{I} - \sigma_i^z) (\mathbb{I} - \sigma_{i+1}^z)$$

En cambio, si se busca reflejar la estructura del problema, bastará considerar parejas de *qubits* cuyos correspondientes pesos en el problema QUBO sean no nulos, digamos que las parejas (i, j) , y proceder de la misma manera

$$U_{ent} = \prod_{(i,j)} e^{i\pi/4} (\mathbb{I} - \sigma_i^z) (\mathbb{I} - \sigma_j^z)$$

Para el entrelazamiento aleatorio se tendría la misma unitaria, con la diferencia de que esta pareja (i, j) no tendría que reflejar ningún aspecto físico del problema.

También es interesante remarcar que es posible generar un estado cuántico sin entrelazamiento, donde el estado global no sería más que el producto de los estados individuales:

$$|\psi(\theta)\rangle = \otimes_{j=1}^N |\psi(\theta_j)\rangle$$

Este producto de estados permite destacar si para un determinado problema las propiedades cuánticas generan algún beneficio perceptible frente a su alternativa clásica, pues este último caso se computaría fácilmente en un ordenador clásico probabilista.

Capítulo 4

Algoritmo cuántico de reconstrucción

4.1. Características del detector

El proceso de reconstrucción de trazas, o *tracking*, hace referencia a la agrupación de los *hits* medidos en un detector de partículas tipo *tracker* con el objetivo de obtener la trayectoria que han seguido las partículas tras la colisión. Así, determinado el momento y la carga que debe poseer cada partícula, de tal forma que se satisfaga la trayectoria estimada en todas las cámaras del detector, se puede predecir de qué partícula se trata.

Todo este proceso de agrupación de *hits* se revela muy costoso computacionalmente, por lo que la eficiencia a la hora de detectar las partículas resultantes es un aspecto esencial para su buen funcionamiento. El coste se debe a la magnitud de datos que se están tratando, por ejemplo, en el LHC la tasa de colisiones es de $4 \cdot 10^8$ grupos de protones cada segundo, produciéndose una elevada cantidad de partículas por cada colisión, y se prevé que el número de protones de cada *bunch* siga aumentando con el tiempo. Con todo, es esencial tener unos métodos de *tracking* lo más actualizados posibles, que se apoyen en tecnologías pioneras y prometedoras, como es el caso de la computación cuántica.

Naturalmente, esta reconstrucción no es tarea sencilla, y menos si se quiere mantener una eficacia alta y a la par que siendo capaces limitar el número de trazas falsas. Respecto a este punto, se ha de mencionar que el fenómeno de las trazas falsas es altamente probable, y se ve agravado por la abundancia de señales espurias que se van generando en los detectores en forma de ruido. Es más, a parte de los *hits* debidos al ruido, si no se trabaja con un algoritmo reconstructor lo suficientemente ágil, también es posible analizar *hits* reales pero que se han formado instantes antes o después del que se está analizando, convirtiendo este proceso de

reconstrucción en una tarea no menos que imposible [17].

A estos estándares de calidad generales hay que sumarles los requerimientos científicos, como puede serlo el umbral de resolución para el momento transverso ¹, pues habrá partículas que requieran de un umbral lo suficientemente alto para poder detectarlas y para otras será mejor un momento mucho más bajo para que se produzca una reconstrucción eficiente [17].

En este trabajo se van a utilizar el detector de trazas del experimento CMS (*Compact Muon Solenoid*) instalado en el LHC del CERN, así como sus algoritmos de reconstrucción. Si bien la jerga es específica del detector, las ideas fundamentales son similares en todos los detectores de partículas, y por lo tanto, se trata de un desarrollo muy general.

Brevemente, el procedimiento de *tracking* funciona del modo siguiente: se comienza reconstruyendo aquellas trazas más sencillas, esto es, las que tienen un momento muy alto y apenas se curvan en el campo magnético, priorizando también las que se han formado en el centro de la interacción; a continuación, se eliminan las trazas reconstruidas, reduciendo la complejidad del problema; para finalizar, se llevan a cabo los preparativos para una nueva iteración. Asimismo, cada una de las reconstrucciones seguiría los siguientes pasos:

- El seeding, proceso de reconstrucción con los *hits* del Pixel, produce unos candidatos a traza con unos pocos *hits*, definiendo una trayectoria aproximada.
- El filtro Kalman [18] extrapola las posibles trayectorias al resto de capas y trata de identificar posibles *hits* en las inmediaciones de la predicción.
- El algoritmo de ajuste mide los mejores candidatos para las trayectorias y, por último, se seleccionan aquellos *hits* mejor posicionados.

En la figura 4.1 puede verse un esquema que representa la estructura de las capas internas del detector CMS.

Ahora bien, podría pensarse que al estar el *Pixel Detector* mucho más cerca del punto de colisión, éste debería ser un detector mucho más saturado que aquellos más alejados, por lo que resultaría más interesante una reconstrucción de trazas empezando desde la capa más externa del CMS. Sin embargo, esto no se produce debido a que el PD tiene una mayor granularidad de sensores, haciendo que la ocupancia de trazas por sensor individual sea más baja. Esta variación de la sensibilidad con la profundidad de los detectores tiene un significado físico, las partículas cargadas interaccionan perdiendo energía a través del proceso de

¹Esto es, la componente perpendicular al *beam* del momento lineal. La importancia de este parámetro recae en el hecho de que es el momento asociado a las interacciones de las partículas que se han creado, pues inicialmente se tiene un sistema con momento transverso nulo.

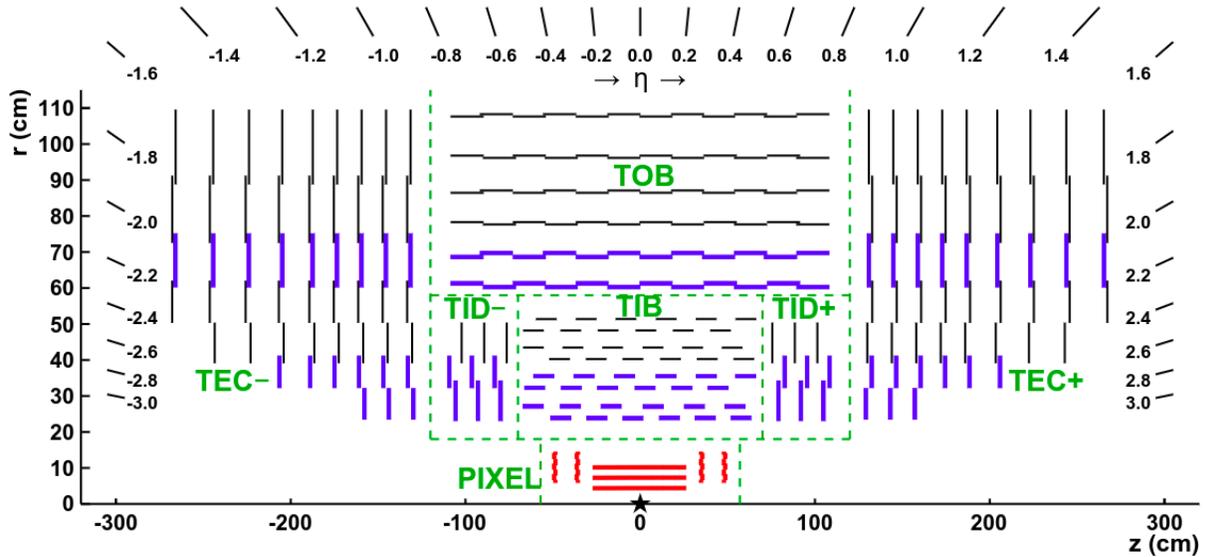


Figura 4.1: Esquema representando los diferentes detectores de tracking en CMS. Se muestra la distancia al eje del beam (vertical) frente a la distancia en este eje (horizontal) centrada en el punto de colisión entre los dos haces de protones, que se indica con una estrella. En la parte superior de la figura se presenta la pseudorapidez η , que se define como $\eta = -\ln(\tan(\theta/2))$, aunque por simplicidad trabajaremos directamente con θ , que mide el desplazamiento angular en el plano de la figura, respecto de la vertical al origen.

ionización, perdiéndose información a medida que se van atravesando los detectores. Los electrones, por ejemplo, emiten fotones de Bremsstrahlung y modifican su trayectoria, mientras que otras partículas sufren interacciones nucleares y pueden decaer a otras diferentes.

4.2. El simulador de eventos

En aras de poder llevar a cabo pruebas de detección y valorar el rendimiento del algoritmo de *tracking*, se optó por la construcción de un simulador del detector de pixels de CMS. Para este propósito, el programa debía ser capaz de inicializarse con una configuración aleatoria, tanto de ruido como de trazas reales, y de proporcionar una lista final de *hits* en cada una de las capas.

Para dar más realismo al proceso de reconstrucción, el simulador se construye en tres dimensiones, aunque, como ya se verá, el algoritmo que se ha programado trabaja sobre el plano. La estructura de un evento particular del simulador se puede ver en la figura 4.2.

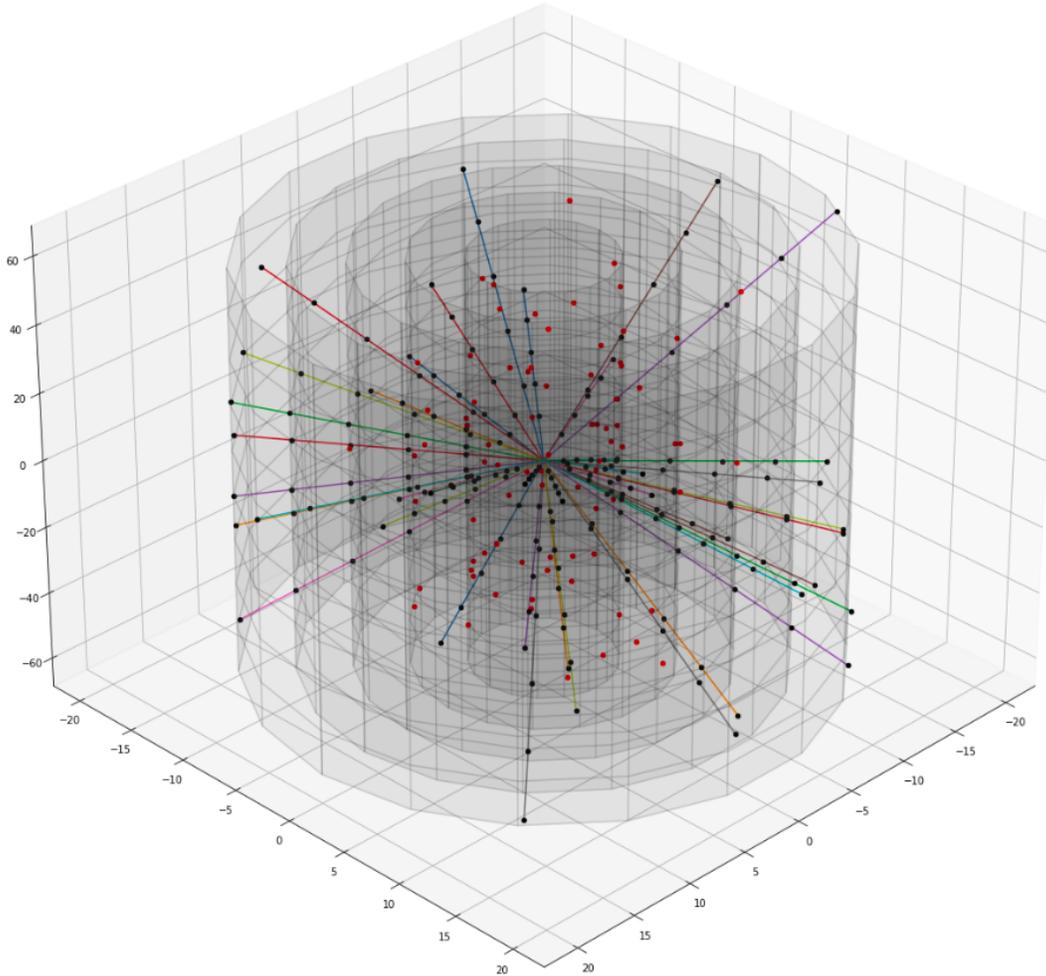


Figura 4.2: Ejemplo de una ejecución del simulador para un sistema de detección de 5 capas, 40 trazas y 80 hits de ruido.

De manera simplificada, este software funciona creando un *hit* inicial en el *beam* del acelerador de partículas, en la región de colisión ², y un *hit* en la última capa del detector. En virtud de ofrecer un evento realista, la posición de este *hit* se escoge con una distribución descentralizada en el eje Z. Así, serán más abundantes las trazas que tiendan a salir próximas a movimiento de los protones. Notar también que se toma una distribución uniforme en la coordenada angular φ . Mediante estos dos puntos $(\varphi_0, z_0, 0)$, (φ_c, z_c, c) , con c la última capa, es posible determinar un vector director para la traza correspondiente. A su vez, será posible hallar los *hits* en el resto de capas $\{(\varphi_1, z_1, 1), (\varphi_2, z_2, 2), \dots, (\varphi_{c-1}, z_{c-1}, c-1)\}$. Es justamente en este paso de intersección de la traza con los detectores donde se introducen los términos de error, que deben reflejar las precisiones que se tienen actualmente en los detectores.

²Notar que los *hits* no salen exactamente del punto $(0, 0, 0)$ sino que esta información es desconocida, y es parte de los objetivos de la reconstrucción determinar el inicio de la traza.

En cuanto a los *hits* espurios, estos se han calculado de manera aleatoria utilizando la misma distribución angular y longitudinal que las trazas, so pretexto de que se acumulen en los laterales del detector.

Las dimensiones están basadas en las del esquema 4.1, con la peculiaridad de que todos los detectores sean equidistantes, siendo la separación:

$$\Delta r = \frac{r_{max} - r_{min}}{c - 1}$$

Asimismo, la longitud de todos los cilindros se corresponde con la media de los detectores del Strip Tracker, unos $120cm$. Una vez más, llevando a cabo la simplificación de que todas las capas poseen la misma longitud.

En cuanto a las verificaciones del algoritmo, se ha de comentar que de todas las trazas que se simularon en tres dimensiones finalmente solo se seleccionaron las más cercanas en la coordenadas φ , pues el algoritmo no discriminará *hits* por esta coordenada. Así, por ejemplo, dos eventos analizables serían los de la figura 4.3. Obsérvese que, además de los *hits* generados, se muestran todas aquellas aristas que los unen. Notar que sólo se unen los de una capa con los de la siguiente, pues será la selección de pares de aristas concatenadas el resultado del algoritmo de reconstrucción.

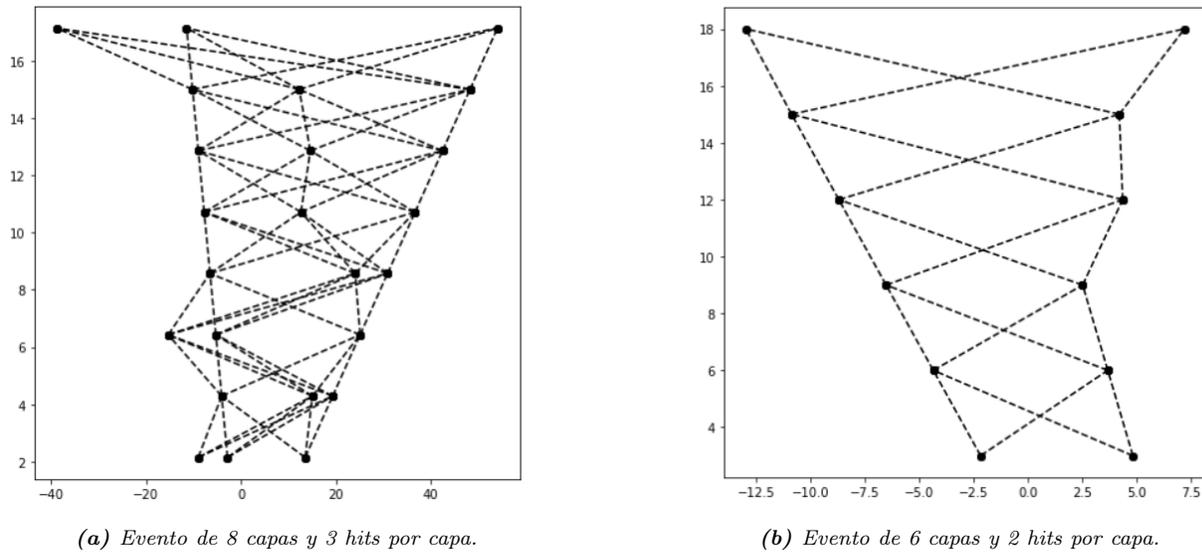


Figura 4.3: Ejemplos de diagramas a los que se les puede aplicar el algoritmo de reconstrucción.

4.3. El algoritmo de *tracking*

El algoritmo trabaja con las aristas que unen a los *hits*, y su objetivo es el de asignar a dichas aristas un 0 o un 1 en función de si esa arista no forme parte de la trayectoria de una traza o que sí lo haga. Así, el algoritmo de *tracking* comienza con las preparaciones del vector de aristas, combinación de dos *hits* posibles, y de la matriz de pesos, combinación de dos aristas posibles.

En cuanto a lo primero, se toma la notación de las aristas numerando en una matriz A , desde la capa más baja y desde la izquierda, las parejas de *hits* involucrados. Asimismo, se guarda en esa matriz las posiciones de los *hits* en el eje Z y de la capa a la que pertenecen, pues estas cantidades serán de utilidad a la hora de definir el peso entre pares de aristas. Naturalmente, en un evento de c capas y N *hits* por cada capa, el número de aristas vendrá dado por

$$|A| = (c - 1)N^2 \quad [\text{Eq. 4.1}]$$

y será éste el número de *qubits* necesario para programar el algoritmo: $|A| = n$.

En cuanto al método de agrupación de aristas, se toma como input la matriz A y se recorre la misma en busca de aristas en las que el último *hit* de la primera aristas sea el *hit* inicial de la segunda, generándose la matriz $s_i s_j$. Nótese que el número de pares de aristas crecerá de manera cúbica:

$$|s_i s_j| = (c - 2)N^3 \quad [\text{Eq. 4.2}]$$

Asimismo, es en este mismo programa donde se define la función de coste, que no será otra cosa que la matriz capaz de establecer cuán rectas están dos trazas. Y como hemos visto, se habrán de hacer $(c - 2)N^3$ operaciones para rellenar esta matriz.

Antes de seguir con el funcionamiento del algoritmo, es adecuado destacar la ventaja o ganancia que puede conseguirse en la agrupación de pares de aristas en función del coste computacional que nos ha costado definir los pesos. Lógicamente, si el problema sólo tiene tres capas de detección la propia función de pesos nos dará la combinación de las dos aristas más rectas, siendo ésta la solución final. Se ha de ir a un mayor número de detectores y a un mayor número de *hits* por capa para contemplar la potencial ventaja. Al igual que se hizo con la combinación de *hits* para generar aristas, y con la agrupación de aristas para determinar el peso, ahora se quiere ver cuántas combinaciones posibles de pares $(s_i s_j, s_i' s_j')$ hay, en el caso general de c capas y N *hits* por capa. El cálculo combinatorio nos arroja la fórmula 4.3, y puede verse en la figura 4.4 el momento a partir del cual se consigue la

ganancia buscada

$$g(c, N) = N^3 N^{c-3} = N^c \quad [\text{Eq. 4.3}]$$

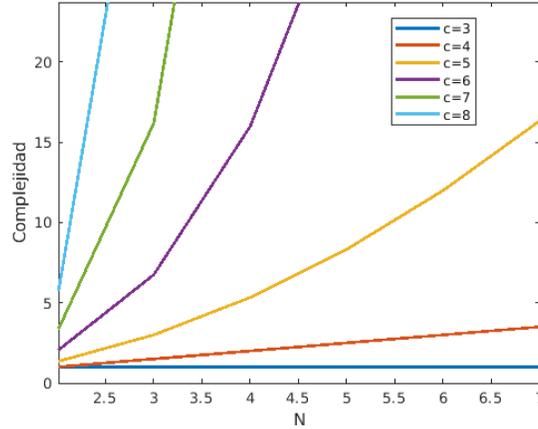


Figura 4.4: Representación de la complejidad en función del número de hits por capa N y del número de capas c . La complejidad se ha definido como relación entre el gasto de resolver el problema combinatorio frente al gasto en definirlo, a saber, el cociente entre 4.3 y 4.2. Se puede observar como se consigue una ventaja a partir de $c = 4$, creciendo esta de manera exponencial con c . A su vez, dado un número de capas, el crecimiento en N es polinómico de grado c .

Como ya se ha mencionado, será la combinación adecuada de aristas la que determine la correcta configuración de trazas, siendo el vector que lo determine la función de onda de los n qubits. Por ejemplo, un vector de onda del tipo $(1, 0, 0, 0, 1, \dots)$ nos mostrará que s_1 y s_5 están unidas. Ahora, el tipo de grafo con el que se está trabajando (ver figura 4.3) no es una mera unión de todos los hits con los demás, se ha de introducir en la función de coste una estructura que especifique una regla de combinación. Esto se hará forzando a que la matriz de costes esté formada por ceros en todas las posiciones menos en los puntos (i, j) donde la arista i pueda ir unida a la j , y, cabe mencionar, no será una tarea complicada por trabajar con aristas ordenadas ³.

³Esta regla no se cumple de manera estricta, como se indicará más adelante.

Por otro lado, la definición por la que se optó para la función de coste para rellenar la matriz es la siguiente:

$$\begin{aligned}
 f(s) = \mu \sum_i \sum_j \left((\cos(\gamma_{ij}))^k + 1 - TOL \right) + \alpha \sum_a \sum_b \sum_{c \neq b} s_{ab} s_{ac} + \alpha \sum_a \sum_{b \neq a} \sum_c s_{ac} s_{bc} + \\
 + \eta \sum_a \sum_b \sum_c \left(z_b - r_a \frac{z_b - z_a}{r_b - r_a} + z_c - r_b \frac{z_c - z_b}{r_c - r_b} \right) + \beta s
 \end{aligned}
 \tag{Eq. 4.4}$$

Analicemos cada uno de estos términos:

- **Término de curvatura.** Se trata del término dominante por introducir pesos negativos, que serán aquellos que discriminen la solución global del evento. La decisión de descentrar la función coseno tiene precisamente este fin, mientras que para ángulos cercanos a π el peso será negativo, para algún valor lo suficientemente alejado, $\pi \pm \epsilon$, la función de coste crecerá. Asimismo, se introduce también un exponente k que variará la pendiente del coseno. Seremos más estrictos en las capas más bajas del detector y más laxos a medida que vayamos aumentando la coordenada r , en aras de dar margen a posibles indicios de curvatura. Por último, la constante μ no será otra cosa que el peso relativo de este primer término, reflejando cuán importante es la curvatura en función del resto de términos.
- **Término de repetición.** Este sumando hace referencia a la imposibilidad de que aparezcan dos trazas que compartan una misma arista, pues es muy improbable que esto ocurra. Obsérvese que el sumatorio se lleva a cabo entre *hits* (a, b, c) , no entre aristas (i, j) , pero esto no da ningún problema al tener la correspondencia directa a partir de la matriz A . Este término hace que la matriz de pesos W luzca ligeramente más complicada, pues aparecerán pesos positivos en regiones diferentes a (i, j) , siendo s_i, s_j trazas que pueden ir unidas. Así, los pesos de repetición asociados, por ejemplo, a $(1, 6)$ y $(3, 6)$ aparecerán en las posiciones $(1, 3)$ y $(3, 1)$.
- **Término de origen.** Mientras que el algoritmo es capaz de unir todo tipo de *hits* generados por una traza, conviene dar más realismo al método y obligarle a reconstruir las trazas que provengan del origen. Obtenido con la simple ecuación de la pendiente de una recta, este sumando fuerza a que si una combinación de dos aristas no pueden prolongarse hasta el origen, entonces aparezca un peso positivo que sea proporcional a la distancia de discrepancia en el *beam*. Naturalmente, el peso relativo de este término

en la función de coste debe ser pequeño, pues al ir aumentando la capa sobre la que se está trabajando se conseguirán mayores desplazamientos con respecto al origen.

- **Término de singularidad.** Con este sumando, introducido *ad hoc*, se pretenden eliminar las aristas sueltas, de forma que se priorice sobre las combinaciones de aristas negativas. Es decir, costará más que una determinada arista aparezca como solución, y deberá combinarse con otras generando un peso más negativo que β para que aparezca.

En este punto es pertinente destacar que, al tener una función de coste que va a involucrar a todo un abanico de estados cuánticos, es muy complicado converger a una la solución con una amplitud cercana a la unidad, pues no es el mismo el coste mínimo del problema que el mínimo coste que se consigue evaluando las funciones de onda experimentales.

Finalmente, como ya se ha comentado en la sección del algoritmo VQE (3.2.2), esta función de coste de tipo QUBO será transformada por el ordenador cuántico en un hamiltoniano de Ising, y el minimizador será capaz de encontrar los parámetros θ que optimicen el menor autovalor del mismo. En cuanto al circuito cuántico que se ha empleado para cada iteración, se han seguido dos estrategias: el entrelazamiento lineal y el compatible. Asimismo, se han llevado a cabo simulaciones con circuitos cuánticos de diferentes longitudes, para estudiar el efecto de la profundidad del circuito en la convergencia. Será en los resultados donde se determine cuál fue la configuración más adecuada.

Capítulo 5

Resultados y análisis

En un primer lugar, se estudia la efectividad del método propuesto mediante el análisis de las trazas que se consiguen reconstruir para distintos tipos de eventos. Con este estudio seremos capaces de discernir si el método es capaz de converger al vector que representa las trazas del evento. Para ello, previamente, se han de ajustar los parámetros de los diferentes tipos de términos de la función de coste que vimos en la sección 4.3, de tal forma que la solución quede lo suficientemente aislada como para que el minimizador COBYLA halle el vector de estado final. En un segundo estudio, se llevan a cabo los cambios pertinentes para observar los efectos del número de *shots*, el tipo de entrelazamiento y la profundidad del circuito cuántico.

Antes de comenzar con las simulaciones, se ha de comentar que el software utilizado ha sido Qiskit [4], el lenguaje de programación de IBM. En particular, se han llevado a cabo todas las ejecuciones en un simulador de un ordenador cuántico ideal llamado *Qasm Simulator*, que proporciona un máximo de 32 *qubits*. Con esto en mente, y recordando la fórmula 4.1, se puede inducir que lo más interesante es trabajar con una pareja de *hits* por cada capa, pues se dará cabida a experimentar con hasta 8 capas de detectores sin sobrepasar el número de *qubits*. Aun así, se ha de remarcar que el método propuesto funciona para cualesquiera números de *hits* por capa y capas del detector¹.

¹Todos los códigos pueden encontrarse en GitHub, donde se recoge el simulador de eventos y el algoritmo de reconstrucción: <https://github.com/PabloReoyo/Qiskit/blob/main/EventoVQE.ipynb>

5.1. Obtención de los parámetros de coste

A falta de un método capaz de encontrar los pesos óptimos para los diferentes términos, se optó por la selección de unos parámetros que reflejaran los requisitos físicos del problema y asilaran al máximo la función de onda solución, que es el mínimo de la función:

$$f(s) = \mu \sum_i \sum_j \left((\cos(\gamma_{ij}))^k + 1 - TOL \right) + \alpha \sum_a \sum_b \sum_{c \neq b} s_{ab} s_{ac} + \alpha \sum_a \sum_{b \neq a} \sum_c s_{ac} s_{bc} + \\ + \eta \sum_a \sum_b \sum_c \left(z_b - r_a \frac{z_b - z_a}{r_b - r_a} + z_c - r_b \frac{z_c - z_b}{r_c - r_b} \right) + \beta s$$

Tras varias medidas quedó claro que la tolerancia debía ser tan pequeña como para seleccionar los ángulos $\alpha_{i,j}$ tales que su coseno no sea mayor a $-0,825$, dejando ese breve margen (con respecto a la total verticalidad) para posibles indicios de curvatura. Por ello, se ha trabajado con $TOL = 0,175$. Asimismo, se introdujo un cambio en la pendiente del coseno al aplicar la potencia k , que va tomando los valores 11, 9, 5, 1, 1, ..., siempre impar. Así, en la primera capa se exigen ángulos que no se alejen más de 10^0 para empezar a restar, en la segunda 12^0 , etc, restando un total de $-0,175$ cuando $\alpha_{i,j} = 180$. Debido a que este desplazamiento en el mínimo del coseno no es notable, se decidió multiplicar todos los sumandos del término de curvatura por una constante $\mu = 25$, factor que estará relacionada con el resto de los términos de f .

Para tratar de mantener una simetría entre los pesos positivos y los negativos, se redujo la magnitud del término de origen a su máxima expresión, pues un la combinación de un triplete de *hits* que no van a formar traza se traducirían en un peso $w_{i,j}$ demasiado positivo. Así, en casi todos los ejemplos se trabajó con $\eta = 0,0025$.

Por otro lado, los términos de no repetición se tomaron de tal forma que fuese inviable que salieran aristas repetidas, manteniendo siempre la armonía con el resto de contribuciones. La elección fue de $\alpha = 5\mu TOL$, a saber, cada arista repetida contribuirá tanto como 5 aristas que formen traza.

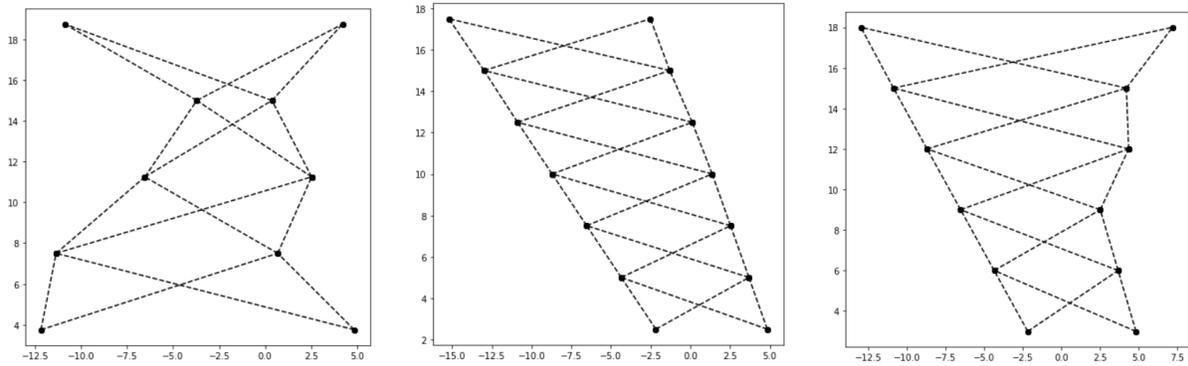
Finalmente, se introdujo un valor de $\beta = \frac{1}{3}\mu TOL$. La razón por la cual no se puede ser más estricto con este término es que va a ir multiplicado por todas y cada una de las aristas que aparezcan, haciendo que la solución final deje de ser tan negativa. En los resultados siguientes se verá cómo el término de singularidad no ha sido capaz de desactivar aristas que no forman una traza completa.

5.2. Resultados

5.2.1. Análisis del rendimiento

En esta primera sección de los resultados se estudia la convergencia del método, tanto desde el punto de vista de la efectividad para encontrar soluciones como desde un punto de vista práctico.

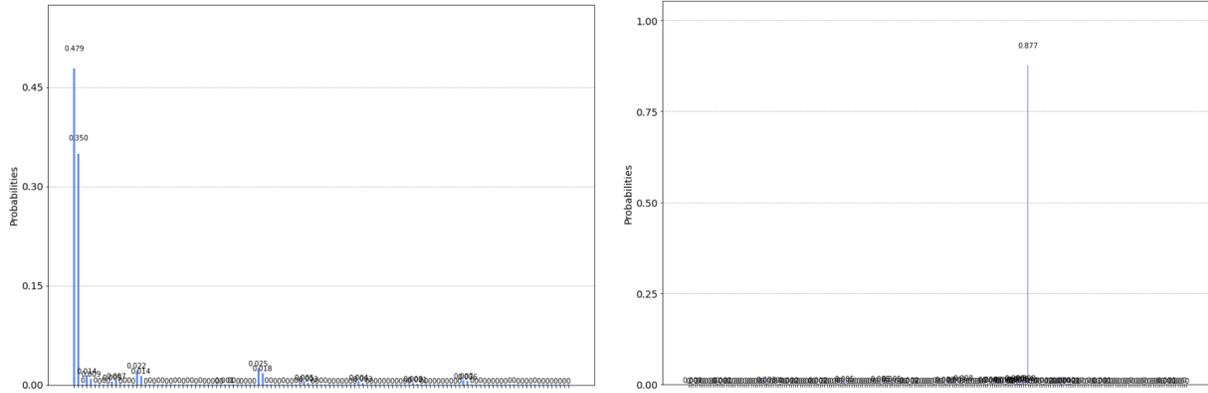
La eventos que se van a estudiar pueden verse en las figuras 5.1a, 5.1b y 5.1c:



(a) Evento de 5 capas y 2 hits por capa, (b) Evento de 7 capas y 2 hits por capa, (c) Evento de 6 capas y 2 hits por capa, sin ninguna traza. con dos trazas. con una traza.

Figura 5.1: Pareja de eventos (5.1a, 5.1b) que se van a resolver con el algoritmo de tracking para comprobar su rendimiento, junto con el evento que se va a utilizar para estudiar el efecto del número de shots (5.1c).

Para llevar a cabo la optimización se fijó un número de *shots*, las veces que se ejecuta el circuito cuántico por cada iteración de la optimización, de 30000 y un entrelazamiento compatible, para ambos eventos. Asimismo, se tomó una profundidad de 5 para el evento de 5 capas y otra de 3 para el de 7 capas. En ambos casos el algoritmo de tracking cuántico fue capaz de hallar la solución global, sendos histogramas finales se pueden ver en las figuras 5.2a y 5.2b:



(a) La solución más abundante a la que se consigue llegar es el vector nulo. (b) La solución más abundante a la que se consigue llegar es el vector de las dos trazas.

Figura 5.2: Histogramas solución para los eventos 5.1a y 5.1b. Las abundancias de ambas soluciones son 4789/10000 y 8770/10000, respectivamente.

A su vez, se muestra en los gráficos 5.3a y 5.3b la velocidad, definida como el coste frente al número de iteraciones, a la que el método ha conseguido converger a la solución.

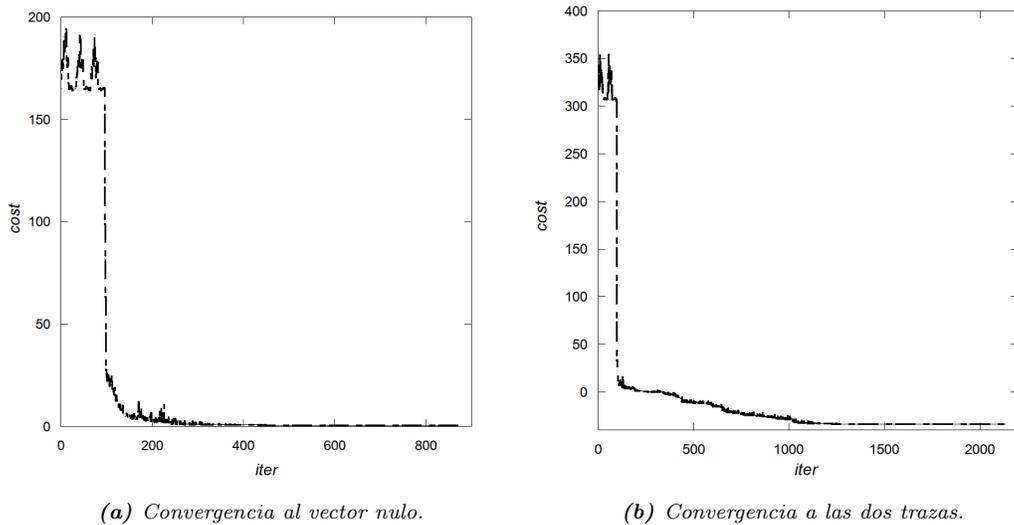


Figura 5.3: Representaciones de la convergencia de los eventos 5.1a y 5.1b. En el primer paso se halla la solución en 871 iteraciones, mientras que en el segundo en 2136.

El perfil de convergencia, tan marcado en torno a las 100 iteraciones y tan lento al aproximarse al mínimo, es el característico del método COBYLA, como se verá en el resto de simulaciones.

En esta misma sección donde se analiza el rendimiento, se comprueba cómo afecta el número de *shots* al proceso de convergencia. Para ello, se han estudiado el evento 5.1c.

Los resultados obtenidos para dos ejecuciones con 10000 y 1000 *shots*, respectivamente, se muestran en las figuras 5.4a y 5.4b:

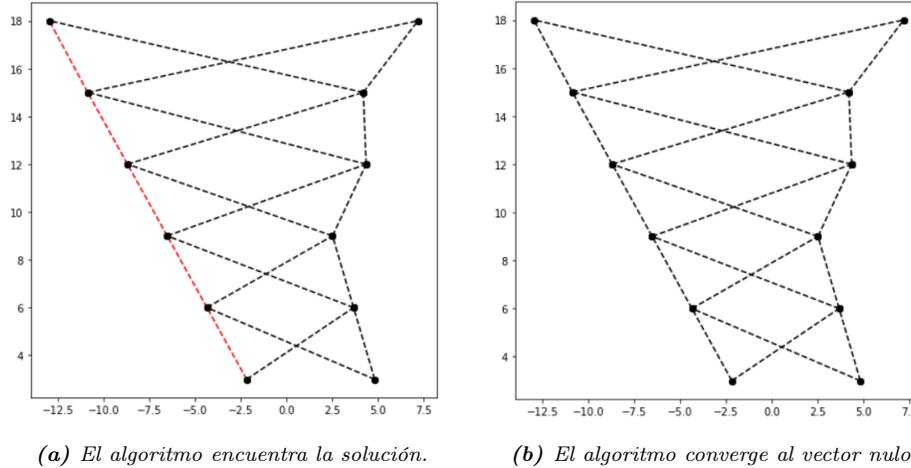


Figura 5.4: Representación de la solución obtenida para 5.1c con 10000 y 1000 *shots*.

En cuanto a la velocidad de convergencia de cada uno de estos casos:

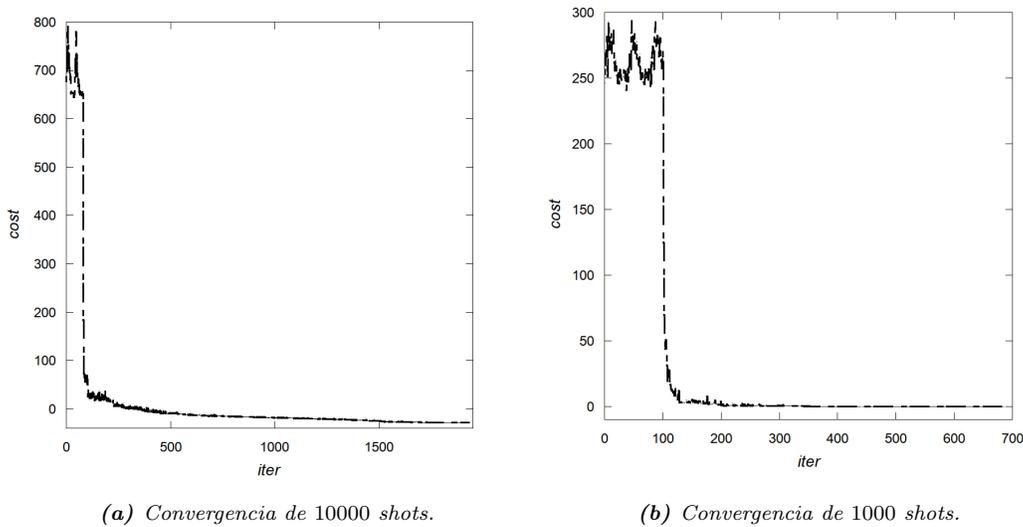


Figura 5.5: Representación de los dos perfiles de convergencia para el estudio en el número de *shots*. En el primer caso se alcanza la solución en 1930 iteraciones, mientras que en el segundo no se llega a la solución, convergiendo en 689 iteraciones.

Con esta comparación somos capaces de concluir que el número de *shots* que se han de aplicar en cada evaluación del circuito cuántico es esencial para que se produzca la conver-

gencia. Así, en el resto de análisis las simulaciones se han llevado a cabo con un número mayor o igual 10000 *shots*.

Para terminar la sección dedicada a la convergencia del método, cabe mencionar que la forma más eficiente que se ha encontrado para determinar el coste de cada iteración viene dada por un híbrido entre los dos costes vistos en la sección 3.2.2: el coste convencional y el *Conditional Value at Risk*. Así, se ha optado por favorecer aquellos estados más abundantes en la superposición solución frente al resto de estados, que tienden a entorpecer la convergencia. La fórmula de coste utilizada viene dada por la expresión 5.1, y no es más que el resultado de elevar al cuadrado las abundancias de la fórmula convencional.

$$E^{(k)} = \frac{1}{s} \sum_{l=1}^s f(s_l^{(k)}) \cdot Ab_l^2 \quad [\text{Eq. 5.1}]$$

Con esta nueva fórmula se ha conseguido una convergencia más rápida, a la par que se ha aumentado la efectividad de la reconstrucción, pues todos los estados finales que generan ruido han pasado a ser despreciables.

5.2.2. Análisis del entrelazamiento

En esta sección se comprueba si es beneficioso el empleo de un entrelazamiento que refleje las características del problema o si basta con introducir un entrelazamiento lineal. Los dos eventos que se van a emplear son 5.1a y 5.1b, pues se quiere verificar si existe un entrelazamiento más beneficioso en función del tamaño de la función de onda. Empecemos por el caso de las cinco capas de detectores. En las figuras 5.6 puede verse la comparación entre los dos entrelazamientos, donde se ha mantenido una profundidad de cinco niveles en el circuito cuántico y se ha ejecutado cada evaluación de función 30000 veces.

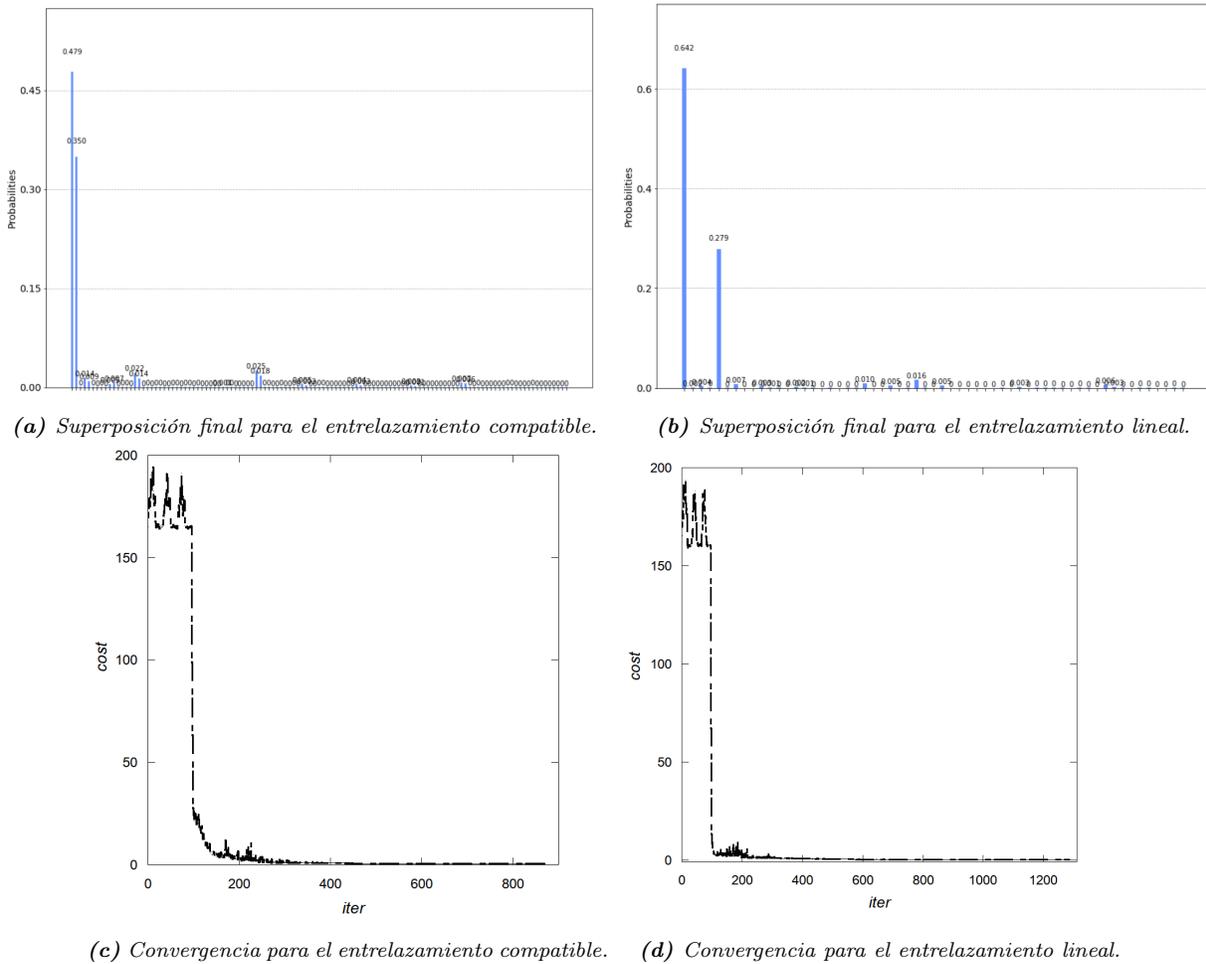
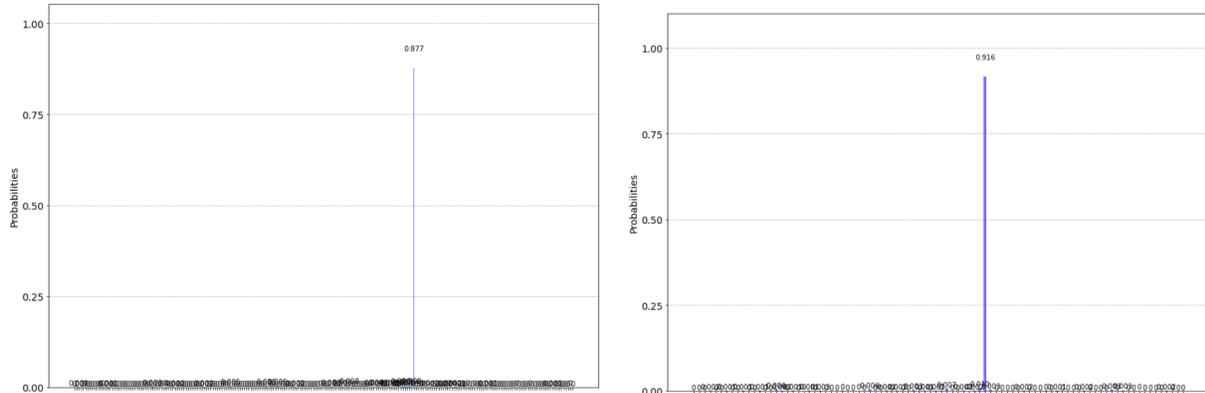


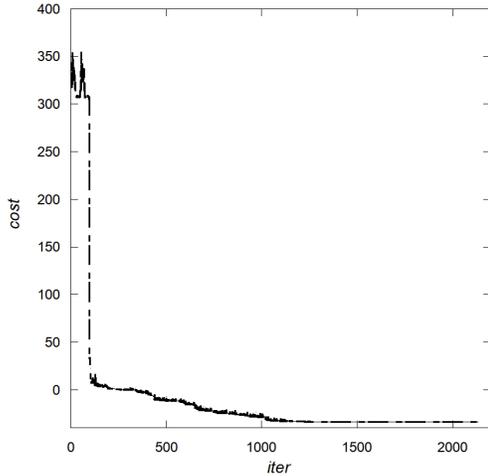
Figura 5.6: Histogramas y perfiles de convergencia para estudiar el efecto del entrelazamiento en el evento 5.1a. Se observa cómo en ambos casos la solución del problema es el vector de onda más abundante, 4786/10000 para el caso compatible y 6419/10000 para el lineal. En el caso compatible se cumple el test de parada en 871 iteraciones, mientras que para el entrelazamiento lineal se han de ejecutar 1284 iteraciones.

En cuanto al evento de 7 capas (diagrama 5.1b), la comparación se muestra en los dos pares de figuras siguientes (5.7), donde se ha trabajado con una profundidad del circuito cuántico de 3 niveles y con 30000 *shots*:

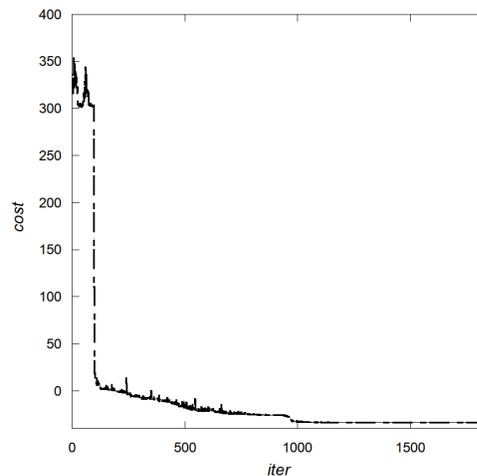


(a) Superposición final para el entrelazamiento compatible.

(b) Superposición final para el entrelazamiento lineal.



(c) Convergencia del algoritmo para el entrelazamiento compatible.



(d) Convergencia del algoritmo para el entrelazamiento lineal.

Figura 5.7: Histogramas y perfiles de convergencia para estudiar los efectos del entrelazamiento del evento 5.1b. La abundancia de la solución para el entrelazamiento compatible es de 8770/10000, mientras que para el caso lineal se consigue 9165/10000. En el primer caso se llega a la solución en 2136 iteraciones, mientras que para el segundo se han de ejecutar 1818 iteraciones.

Debido a que esta pareja de eventos no nos permite establecer un juicio perentorio sobre el carácter del entrelazamiento en nuestro algoritmo, se han llevado a cabo dos comparaciones adicionales, que pueden verse resumidas en la tabla 5.1.

Simulación	$depth$	$shots$	Entrelazamiento	Abundancia	Iteraciones
1	2	10000	Compatible	9671/10000	924
2	2	10000	Lineal	—	498
3	5	50000	Compatible	9562/10000	1431
4	5	50000	Lineal	—	878

Cuadro 5.1: Simulaciones para comprobar el efecto del tipo de entrelazamiento en el circuito cuántico. El evento que se ha estudiado en todas las simulaciones se caracteriza por poseer $N = 2$ y $c = 5$, con una sola traza. Mientras que en los dos casos donde se aplica el entrelazamiento compatible se converge a la solución, en el caso del entrelazamiento lineal se converge al vector nulo, pues no se consiguen encontrar los vectores de estado con pesos negativos.

Finalmente, con el estudio de estos todos estos eventos podemos hacernos una idea de que el tipo de entrelazamiento que implemente el circuito cuántico es relevante, siendo más eficiente un entrelazamiento compatible que uno meramente lineal. Pues, a pesar de que el coste en iteraciones sea más elevado, la eficacia será significativamente más elevada. Asimismo, con estos resultados corroboramos los obtenidos en [16], que pueden verse en las gráficas 5.8. Se ha de notar que en ese artículo se introduce una distinción entre grafos poco densos y muy densos, donde la densidad viene dada por la relación entre el número de aristas posibles entre parejas de $hits$ y las aristas verdaderamente presentes, y puede verse en la fórmula 5.2.

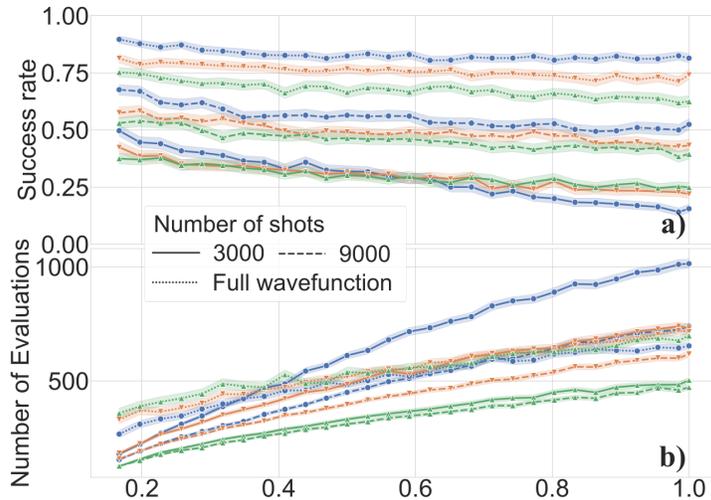


Figura 5.8: Representación de la tasa de acierto del algoritmo VQE y el número de iteraciones necesarias para converger a la solución frente la densidad del grafo que se quiere resolver. Se muestra en azul el entrelazamiento compatible, en amarillo el lineal y en verde el estado sin entrelazamiento, que no estudiamos en este trabajo. Crédito: [16].

$$D = \frac{2a}{Nc \cdot (Nc - 1)} = \frac{2(c - 1)N}{Nc^2 - c} \quad [\text{Eq. 5.2}]$$

Notar que el grafo que se ha estudiado para los resultados de la tabla 5.1 posee una densidad $D \approx 0,36$, luego, según la figura 5.8, la superposición compatible es la más efectiva independientemente del número de *shots*. Por otro lado, podemos ver en esa misma figura que, en general, también es la que más iteraciones necesita para converger. Con todo, podemos concluir esta sección mostrando que se han obtenidos los resultados previstos.

5.2.3. Análisis de la profundidad del circuito

Con respecto al análisis de la importancia del tamaño del circuito cuántico, comenzaremos viendo los resultados que hemos obtenido para discutir, en un último bloque, si se obtienen unos resultados acordes con [16]. El evento que se ha estudiado es el de la figura 5.1c ($N = 2$ y $c = 6$, con densidad $D \approx 0,3$), mediante 10000 *shots* y un entrelazamiento compatible, tanto para una profundidad de cinco capas como para una única capa. Las superposiciones finales y la convergencia pueden verse en las siguientes figuras (5.9):

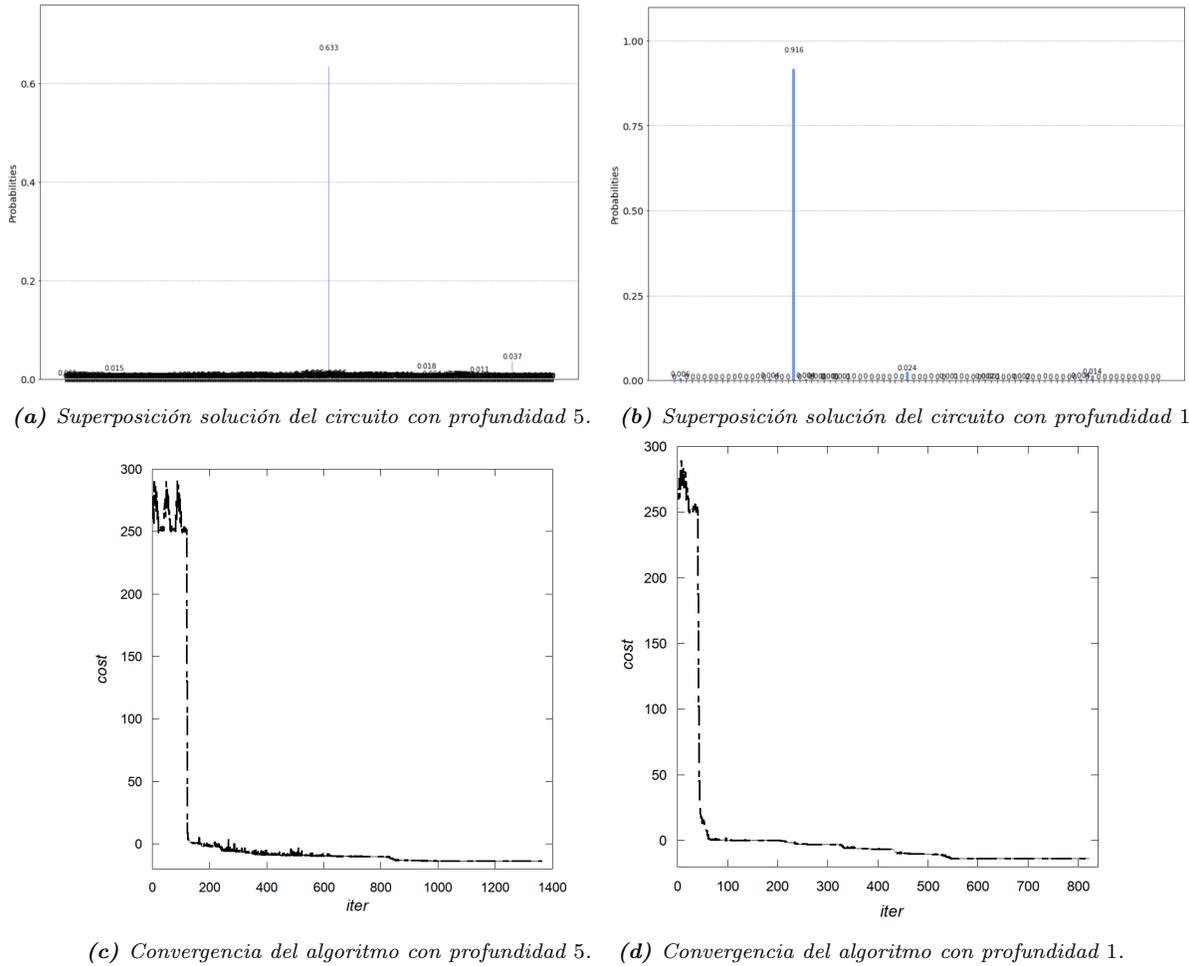


Figura 5.9: Histogramas resultantes para el estudio del evento 5.1c y sus correspondientes perfiles de convergencia. En ambos casos se llega a la solución, que posee una abundancia 6333/10000 y 9163, respectivamente. En cuanto a la convergencia, se cumplen los test de parada en 1361 y 821 iteraciones.

En definitiva, la profundidad del circuito cuántico se ha reflejado en un número de parámetros a optimizar más elevado, por lo que se podría llegar a pesar que ha de ser más preciso al determinar la solución. Asimismo, al aumentar la profundidad también aumentan

las puertas de entrelazamiento, que de manera ideal conllevaría una convergencia más rápida. A pesar de ello, se ha visto en los resultados que la longitud del circuito cuántico no es verdaderamente significativo para ninguna de las dos pretensiones. Con estos resultados reafirmamos el estudio [16], pues en éste se obtenía una dependencia que se estabiliza a partir de la profundidad $depth = 1$ para una densidad similar a la de nuestro problema. Se pueden ver los resultados de este estudio en las figuras 5.10.

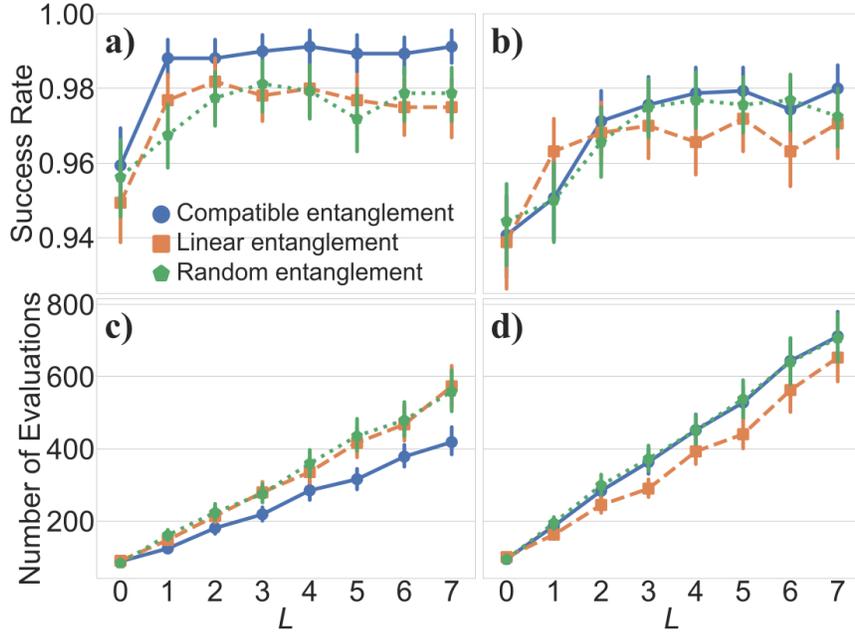


Figura 5.10: Tasa de acierto y número de iteraciones necesarias para la convergencia en función de la longitud del circuito cuántico ($depth$). La primera columna se corresponde con un evento de densidad $D \approx 0,26$, mientras que la segunda con uno de $D \approx 0,89$.

Crédito: [16].

Capítulo 6

Conclusiones

Veamos primero alguna conclusión referente al método de detección de partículas implementado, para discutir, a continuación, algunos puntos de la propia tecnología de la computación cuántica.

Lejos de sostenerse como una alternativa competitiva a los algoritmos de reconstrucción actuales, optimizados durante largos años, y a donde no podemos aspirar, la propuesta de un método de *tracking* que utilice el algoritmo VQE ha resultado exitosa. La capacidad para encontrar la correcta configuración de aristas ha mostrado que con este proceso se pueden reconstruir diversos tipos de diagramas, desde aquellos con *hits* sueltos hasta los que están conformados por *hits* provenientes de trazas, evidenciándose así su versatilidad. Asimismo, se ha conseguido remarcar la diferencia entre dos tipos de entrelazamiento, demostrándose que aportar más información sobre el problema a resolver se traduce en una convergencia más eficaz. También se ha determinado el papel de la longitud del circuito cuántico, que se ha revelado útil hasta con una profundidad mínima. Este hecho es de suma importancia, si se quisieran hacer simulaciones en un ordenador cuántico actual, la decoherencia sólo nos dejaría trabajar con circuitos de unas pocas puertas cuánticas. Con esto nos aseguramos que el método desarrollado es implementable en la época NISQ. También se ha de notar que se ha logrado programar una aproximación al *Conditional Value at Risk* como una ligera variación al método convencional, consiguiéndose con ello una mejora en la convergencia. De este modo, el coste de cada iteración ha estado marcado por los estados menos energéticos que se han alcanzado en la superposición, y los más positivos han sido rápidamente descartados.

En contraposición, una breve mirada a los perfiles de convergencia nos deja entrever que el método de optimización no converge todo lo rápido que se esperaría. La caída pronunciada en torno a las 100 iteraciones, seguida por un lento proceso de minimización, sugiere que hay que buscar alguna innovación para desencadenar una convergencia más uniforme. En este sentido, se podría experimentar con unos test de parada más adecuados a los resultados que estamos observando, o con otros optimizadores que no se vean tan alterados por fluctuaciones debidas a la superposición.

Por otro lado, y mirando hacia el futuro, se podría tratar de adaptar este método para que sea capaz de trabajar en 3 dimensiones. Bastaría con introducir otro término para la fórmula de coste que tenga en cuenta la variable angular φ , definida en el plano transversal, de forma similar a lo que hacemos con el término de curvatura. Así, abandonando el plano, obtendríamos un método más realista y en el cual pudiéramos sacar pleno partido al simulador de eventos que se ha programado.

En definitiva, la sucinta indagación sobre las bases de la computación cuántica nos ha permitido comprobar que esta herramienta puede llegar a ser muy útil en el futuro. Mientras que en la actualidad nos hemos de limitar a contemplar sus capacidades con simuladores o con computadores reales de unos pocos *qubits*, se ha explorado el novedoso camino de la época NISQ, poniéndose de manifiesto las propiedades cuánticas incluso en un momento marcado por la alta decoherencia. Es posible que se hayan de esperar varias décadas hasta que emerja una tecnología de computación cuántica que verdaderamente desencadene un cambio de paradigma en el marco de la computación. Mientras tanto, lo más sensato es no declarar ninguna supremacía.

Bibliografía

- [1] M. Nielsen e I. Chuang, *Quantum Computation and Quantum Information*. 2011, ISBN: 9781107002173.
- [2] G. Benenti, G. Casati y G. Strini, *Principles of Quantum Computation and Information - Volume I: Basic Concepts*. 2004, ISBN: 9789812563453.
- [3] C. H. Bennett, E. Bernstein, G. Brassard y U. Vazirani, “Strengths and weaknesses of quantum computing,” *SIAM J. Comput.*, 1997. DOI: 10.1137/S0097539796300933. arXiv: 9701001 [quant-ph].
- [4] QiskitTeam, “Qiskit: An Open-source Framework for Quantum Computing,” 2021. DOI: 10.5281/zenodo.2573505.
- [5] J. Preskill, “Reliable quantum computers,” *Proc. R. Soc. A Math. Phys. Eng. Sci.*, 1998. DOI: 10.1098/rspa.1998.0167. arXiv: 9705031 [quant-ph].
- [6] P. W. Shor, “Fault-tolerant quantum computation,” *Proc. 37th Symp. Found. Comput. Sci.*, 1996. arXiv: 9605011 [quant-ph].
- [7] Y. Makhlin, G. Schön y A. Shnirman, “Quantum-state engineering with Josephson-junction devices,” *Rev. Mod. Phys.*, 2001. DOI: 10.1103/RevModPhys.73.357. arXiv: 0011269 [cond-mat].
- [8] G. Benenti, G. Casati y G. Strini, *Principles of Quantum Computation and Information-Volume II: Basic Tools and Special Topics*. 2007, ISBN: 9789812563453.
- [9] D. Bultrini, M. H. Gordon, E. López y G. Sierra, “Simple Mitigation Strategy for a Systematic Gate Error in IBMQ,” *J. Appl. Math. Phys.*, 2021. DOI: 10.4236/jamp.2021.96083. arXiv: 2012.00831.
- [10] L. K. Grover, “A fast quantum mechanical algorithm for database search,” *Proceedings, STOC*, 1996.

-
- [11] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W. K. Mok, S. Sim, L. C. Kwek y A. Aspuru-Guzik, “Noisy intermediate-scale quantum algorithms,” *Rev. Mod. Phys.*, 2021. DOI: 10.1103/RevModPhys.94.015004. arXiv: arXiv:2101.08448v1.
- [12] Y.-H. Oh, H. Mohammadbagherpoor, P. Dreher, A. Singh, X. Yu y A. J. Rindos, “Solving Multi-Coloring Combinatorial Optimization Problems Using Hybrid Quantum Algorithms,” 2019. arXiv: 1911.00595.
- [13] M. J. D. Powell, “A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation,” *Adv. Optim. Numer. Anal.*, 1994. DOI: 10.1007/978-94-015-8330-5_4.
- [14] H. D. Pi, G. Hahn, G. Chapuis y G. Rizk, “Efficient Combinatorial Optimization Using Quantum Annealing,” 2016. arXiv: arXiv:1801.08653v2.
- [15] P. K. Barkoutsos, “Improving Variational Quantum Optimization using CVaR,” *Quantum*, 2020.
- [16] P. Díez-Valle, D. Porrás y J. J. García-Ripoll, “Quantum variational optimization: The role of entanglement and problem hardness,” *Phys. Rev. A*, 2021. DOI: 10.1103/PhysRevA.104.062426. arXiv: 2103.14479.
- [17] CMS-Colaboration, “Description and performance of track and primary-vertex reconstruction with the CMS tracker,” *J. Instrum.*, 2014. DOI: 10.1088/1748-0221/9/10/P10009. arXiv: 1405.6569.
- [18] R. E. Kalman, “A New Approach to Linear filtering and Prediction Problems,” *Transactions of the ASME—Journal of Basic Engineering*, n.º Series D, 1960.