

Optimización cuántica: quantum annealing y algoritmos variacionales

Elías F. Combarro (Universidad de Oviedo)
efernandezca@uniovi.es

Universidad de Almería - Octubre 2022



UNIVERSIDAD
DE ALMERÍA



Universidad de Oviedo

Parte I

Introducción y objetivos

La computación cuántica está de moda

The Nobel Prize in Physics 2022



III. Niklas Elmehed © Nobel Prize Outreach

Alain Aspect

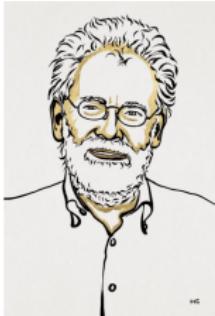
Prize share: 1/3



III. Niklas Elmehed © Nobel Prize Outreach

John F. Clauser

Prize share: 1/3



III. Niklas Elmehed © Nobel Prize Outreach

Anton Zeilinger

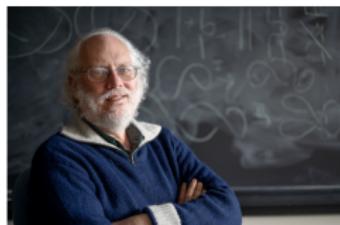
Prize share: 1/3

Peter Shor wins Breakthrough Prize in Fundamental Physics

MIT professor to share \$3 million prize with three others; Daniel Spielman PhD '95 wins Breakthrough Prize in Mathematics.

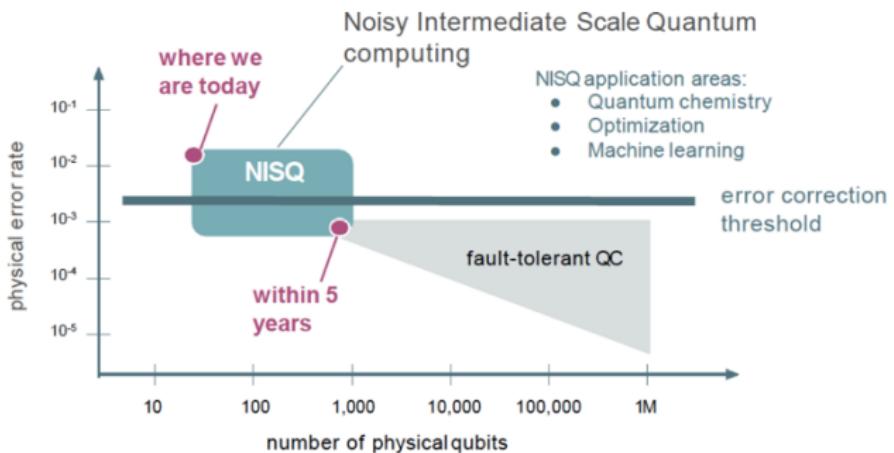
[Watch Video](#)

Jennifer Chu / MIT News Office
September 22, 2022



La era NISQ

- Noisy Intermediate-Scale Quantum computers
 - Afectados por el ruido
 - Número de qubits limitado (50-100)
 - Sin conexión completa
- A pesar de ello, estos ordenadores podrían ser útiles en ciertas tareas: optimización, machine learning...



"Quantum computing in the NISQ era and beyond" Preskill, 2018 <https://arxiv.org/abs/1801.00862>



Objetivos de este curso

- Profundizar en las aplicaciones de la computación cuántica en problemas de optimización (especialmente, optimización combinatoria)
- Estudiar en detalle los siguientes métodos:
 - Quantum Annealing
 - Quantum Approximate Optimization Algorithm (QAOA)
 - Variational Quantum Eigensolver (VQE)
- Aprender a resolver problemas de optimización con algoritmos cuánticos en:
 - Quantum annealers
 - Ordenadores cuánticos basados en puertas



Recursos y herramientas

- Jupyter Notebooks
 - Entorno para la creación y ejecución de documentos que combinan código, texto enriquecido e imágenes
 - Trabajaremos principalmente en Google Colab
- Acceso a ordenadores cuánticos reales
 - Quantum Annealers: D-Wave Leap
<https://cloud.dwavesys.com/leap/>
 - Ordenadores basados en puertas: IBM Quantum
<https://quantum-computing.ibm.com/>
- Notebooks del curso en <https://github.com/EliasCombarro/Almeria2022>

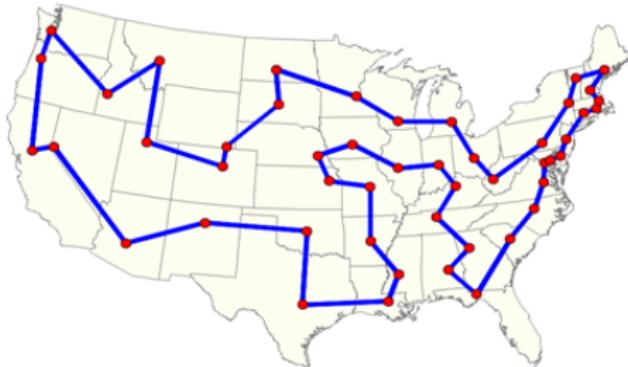


Parte II

QUBO: Quadratic Unconstrained
Binary Optimization

Problemas de optimización combinatoria

- Los problemas en los que nos vamos a centrar principalmente provienen de la optimización combinatoria
- Esto incluye problemas muy famosos:
 - El problema del viajante de comercio
 - El problema de la mochila
 - Hallar un coloreado mínimo en un grafo
 - Determinar si una fórmula es satisfacible
 - ...
- Se trata de problemas NP-hard (en algunos casos, NP-completos)



Reformulando problemas de optimización combinatoria

- Para usar métodos como QAOA o quantum annealing, necesitamos reformular nuestros problemas
- Los dos “formatos” principales que usaremos son:
 - QUBO (Quadratic Unconstrained Binary Optimization): Minimizar

$$\sum_{i,j} q_{ij} x_i x_j$$

con $x_i \in \{0, 1\}$ y q_{ij} coeficientes reales

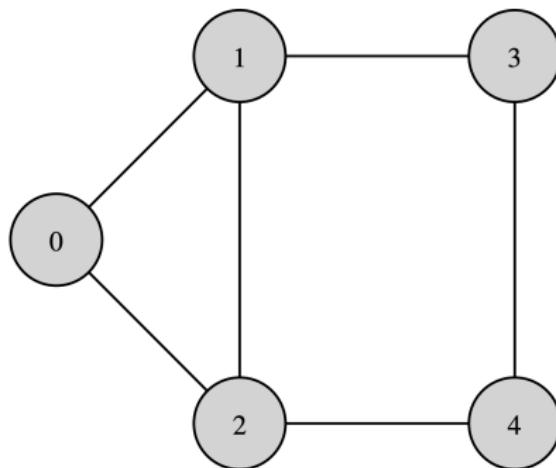
- Modelo de Ising: Minimizar

$$-\sum_{j,k} J_{jk} Z_j Z_k - \sum_j h_j Z_j$$

con $Z_i \in \{-1, 1\}$ y J_{jk}, h_j coeficientes reales

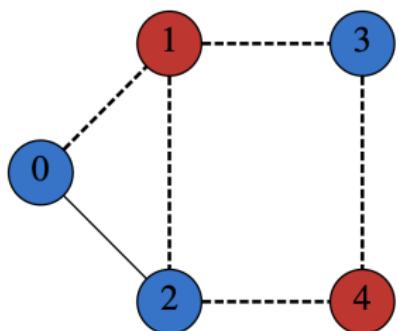
El problema del corte máximo

- Consideraremos el problema de dividir los vértices de un grafo en dos grupos maximizando los ejes cortados

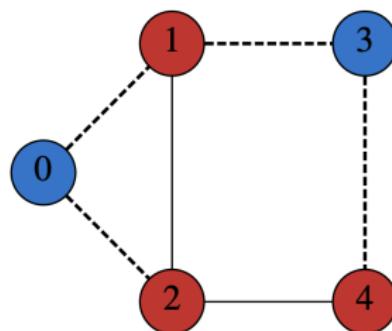


- Es un problema NP-hard

Dos posibles cortes



(a) Optimal cut.



(b) Non-optimal cut.

Planteando el problema del corte máximo con *spins*

- Identificamos cada vértice i del grafo con una variable Z_i que puede tomar valor 1 o -1
- Asignamos valor 1 a los vértices de un grupo y -1 a los del otro
- Entonces, si E es el conjunto de ejes, el problema se puede plantear como

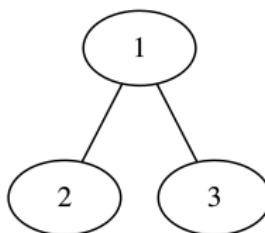
$$\text{Minimizar} \sum_{(i,j) \in E} Z_i Z_j$$

ya que vértices en distintos grupos aportan -1 a la suma y vértices del mismo grupo aportan 1

Ejemplo de corte máximo

- Para el grafo de la figura se trata de minimizar

$$H = Z_1 Z_2 + Z_1 Z_3$$



- Por inspección (o enumerando todas las posibilidades) se ve que las soluciones óptimas son 011 y 100

El modelo de Ising

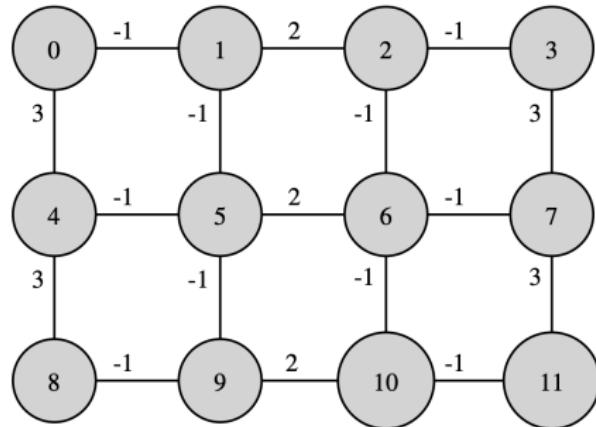
- Se tienen n partículas con spin, que interactúan entre sí con ciertas fuerzas de acoplamiento
- Su hamiltoniano es

$$H = - \sum_{1 \leq i < j \leq n} J_{ij} Z_i Z_j - \sum_{i=1}^n h_i Z_i$$

con J_{ij} y h_i coeficientes reales

- Queremos encontrar una asignación de valores de spins (1 o -1) que minimice la suma
- El problema general es NP-hard

Un ejemplo de modelo de Ising



Minimizar
$$\begin{aligned} & z_0z_1 - 2z_1z_2 + z_2z_3 - 3z_0z_4 + z_1z_5 + z_2z_6 - 3z_3z_7 \\ & + z_4z_5 - 2z_5z_6 + z_6z_7 - 3z_4z_8 + z_5z_9 + z_6z_{10} - 3z_7z_{11} \\ & + z_8z_9 - 2z_9z_{10} + z_{10}z_{11} - z_0 - z_1 - z_2 - z_3 - z_4 - z_5 \\ & - z_6 - z_7 - z_8 - z_9 - z_{10} - z_{11} \end{aligned}$$

sujeto a
$$z_j \in \{-1, 1\}, \quad j = 0, \dots, 11.$$

¿Y dónde metemos la computación cuántica en todo esto?

- Recordemos que la puerta Z tiene como matriz

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

y que el vector $|0\rangle$ tiene como coordenadas

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

- Entonces

$$(1 \ 0) \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 1$$

- Podemos ver el anterior producto de matrices y vectores como

$$\langle 0 | Z | 0 \rangle = 1$$

¿Y dónde metemos la computación cuántica en todo esto?

- Análogamente

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

- Así que

$$\langle 1 | Z | 1 \rangle = (0 \quad 1) \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -1$$

- Si tenemos más qubits, podemos utilizar productos tensoriales. Por ejemplo, denotando $Z_1 Z_2 = Z_1 \otimes Z_2$, tenemos

$$\langle 01 | Z_1 Z_2 | 01 \rangle = (\langle 0 | Z_1 | 0 \rangle) \cdot (\langle 1 | Z_2 | 1 \rangle) = 1 \cdot (-1) = -1$$

y, denotando $Z_1 Z_3 = Z_1 \otimes Z_3$, tenemos

$$\langle 101 | Z_1 Z_3 | 101 \rangle = (\langle 1 | Z_1 | 1 \rangle) \cdot (\langle 1 | Z_3 | 1 \rangle) = (-1) \cdot (-1) = 1$$

Volviendo al ejemplo de corte máximo

- Teníamos el problema de corte representado por

$$H = Z_1Z_2 + Z_1Z_3$$

- Podemos identificar un posible corte con $|011\rangle$ (tomar los vértices 2 y 3 y dejar fuera el 1) y evaluar su coste mediante

$$\begin{aligned}\langle 011 | H | 011 \rangle &= \langle 011 | (Z_1Z_2 + Z_1Z_3) | 011 \rangle \\ &= \langle 011 | Z_1Z_2 | 011 \rangle + \langle 011 | (Z_1Z_3) | 011 \rangle = -1 + (-1) = -2\end{aligned}$$

- Del mismo modo

$$\begin{aligned}\langle 010 | H | 010 \rangle &= \langle 010 | (Z_1Z_2 + Z_1Z_3) | 010 \rangle \\ &= \langle 010 | Z_1Z_2 | 010 \rangle + \langle 010 | (Z_1Z_3) | 010 \rangle = -1 + 1 = 0\end{aligned}$$

Extendiendo la búsqueda a todos los estados cuánticos

- Un estado cuántico genérico se puede escribir como

$$|\psi\rangle = \sum_x a_x |x\rangle$$

- Para cada x se verifica

$$Z_j Z_k |x\rangle = \pm |x\rangle$$

- Tenemos que

$$\begin{aligned}\langle\psi| Z_j Z_k |\psi\rangle &= \left(\sum_y a_y^* \langle y| \right) Z_j Z_k \left(\sum_x a_x |x\rangle \right) \\ &= \sum_y \sum_x a_y^* a_x \langle y| Z_j Z_k |x\rangle \\ &= \sum_x |a_x|^2 \langle x| Z_j Z_k |x\rangle\end{aligned}$$

Extendiendo la búsqueda a todos los estados cuánticos

- Por linearidad

$$\begin{aligned}\langle \psi | (Z_1 Z_2 + Z_1 Z_3) |\psi \rangle &= \langle \psi | Z_1 Z_2 |\psi \rangle + \langle \psi | Z_1 Z_3 |\psi \rangle \\&= \sum_x |a_x|^2 \langle x | Z_1 Z_2 |x \rangle \\&\quad + \sum_x |a_x|^2 \langle x | Z_1 Z_3 |x \rangle \\&= \sum_x |a_x|^2 \langle x | (Z_1 Z_2 + Z_1 Z_3) |x \rangle\end{aligned}$$

Extendiendo la búsqueda a todos los estados cuánticos

- Se sigue que

$$\begin{aligned}& \sum_x |a_x|^2 \langle x | (Z_1 Z_2 + Z_1 Z_3) |x\rangle \\& \geq \sum_x |a_x|^2 \langle x_{\min} | (Z_1 Z_2 + Z_1 Z_3) |x_{\min}\rangle \\& = \langle x_{\min} | (Z_1 Z_2 + Z_1 Z_3) |x_{\min}\rangle \sum_x |a_x|^2 \\& = \langle x_{\min} | (Z_1 Z_2 + Z_1 Z_3) |x_{\min}\rangle\end{aligned}$$

donde x_{\min} es el estado para el que se alcanza el mínimo.

El principio variacional

- Podemos extender el resultado a una matriz de Ising de la forma

$$H = - \sum_{1 \leq i < j \leq n} J_{ij} Z_i Z_j - \sum_{i=1}^n h_i Z_i$$

ya que $Z|x\rangle = \pm|x\rangle$.

- Tendremos que

$$\langle \psi | H | \psi \rangle = \sum_x |a_x|^2 \langle x | H | x \rangle$$

- Y también que

$$\langle \psi | H | \psi \rangle \geq \langle x_{min} | H | x_{min} \rangle$$

- A veces se denomina “principio variacional” a este resultado

El maravilloso mundo de los hamiltonianos

- Entonces, lo que nos interesa es hallar un estado cuántico $|\psi\rangle$ de forma que

$$\langle\psi|H|\psi\rangle$$

sea mínimo, con

$$H = - \sum_{1 \leq i < j \leq n} J_{ij} Z_i Z_j - \sum_{i=1}^n h_i Z_i$$

- Se trata de un caso particular de un problema muy importante en física: hallar el estado de energía mínima (**ground state**) de un hamiltoniano
- Un hamiltoniano es una matriz H hermitiana ($H = H^\dagger$)
- La energía de un estado $|\psi\rangle$ es

$$\langle\psi|H|\psi\rangle$$

Ejemplo: El problema del particionado de números

- Tenemos una lista de números (s_1, s_2, \dots, s_n)
- Queremos encontrar una partición de la lista en dos conjuntos de suma tan parecida como sea posible
- Es un problema NP-hard (aunque en casos prácticos se suele poder resolver eficientemente)
- Usamos variables Z_i que indican si el elemento i -ésimo pertenece a un conjunto (1) o al otro (-1)
- Podemos formular el problema como

$$\text{Minimizar} \left(\sum_{i=1}^n s_i Z_i \right)^2$$

y expandir y simplificar teniendo en cuenta que $Z_i^2 = 1$.

Un caso de particionado de números

- Consideramos la lista $(2, 5, 4, 7)$
- Queremos minimizar

$$(2Z_1 + 5Z_2 + 4Z_3 + 7Z_4)^2$$

es decir

$$\begin{aligned} 4Z_1^2 + 20Z_1Z_2 + 16Z_1Z_3 + 28Z_1Z_4 + 25Z_2^2 \\ + 40Z_2Z_3 + 70Z_2Z_4 + 16Z_3^2 + 56Z_3Z_4 + 49Z_4^2 \end{aligned}$$

- Es equivalente a minimizar

$$20Z_1Z_2 + 16Z_1Z_3 + 28Z_1Z_4 + 40Z_2Z_3 + 70Z_2Z_4 + 56Z_3Z_4$$

El problema Subset Sum

- El problema Subset Sum consiste en, dado un conjunto de números enteros $S = \{a_0, \dots, a_m\}$ y un valor objetivo T , hallar un subconjunto de S cuya suma sea lo más cercana posible a T .
- Es NP-hard
- Se podría escribir como un caso de minimizar un modelo de Ising
- Es más sencillo escribirlo como minimizar

$$c(x_0, x_1, \dots, x_m) = (a_0x_0 + a_1x_1 + \dots + a_mx_m - T)^2$$

donde $x_i \in \{0, 1\}$.

Ejemplo de Subset Sum

- $S = \{1, 4, -2\}$ y $T = 2$
- Tenemos que minimizar

$$(x_0 + 4x_1 - 2x_2 - 2)^2$$

- Expandiendo, tenemos

$$x_0^2 + 8x_0x_1 - 4x_0x_2 - 4x_0 + 16x_1^2 - 16x_1x_2 - 16x_1 + 4x_2^2 + 8x_2 + 4$$

un polinomio de grado dos en variables binarias

- Podríamos simplificar: eliminar el término independiente y tener en cuenta que $x_i^2 = x_i$
- Es un caso de un problema QUBO: Quadratic Unconstrained Binary Optimization

QUBO: Quadratic Unconstrained Binary Optimization

- Los problemas QUBO se plantean como

$$\text{Minimizar} \sum_{1 \leq i \leq j \leq n}^n w_{ij} x_i x_j$$

donde cada x_i es una variable binaria y los w_{ij} son coeficientes reales

- También se puede plantear como minimizar

$$x^T Q x$$

con x el vector columna de elementos (x_i), Q una matriz simétrica tal que $Q_{ij} = Q_{ji} = \frac{1}{2} w_{ij}$ y $Q_{ii} = w_{ii}$

- Se puede reescribir como un modelo de Ising con la transformación

$$x_i = \frac{1 - Z_i}{2}$$

y volver a QUBO con

$$Z_i = 1 - 2x_i$$

Ejemplo: De QUBO a Ising

- Consideramos el problema de minimizar

$$3x_1^2 - 4x_1x_2 + 4x_2^2$$

con x_1 y x_2 variables binarias

- Hacemos el cambio

$$x_1 = \frac{1 - Z_1}{2}$$

$$x_2 = \frac{1 - Z_2}{2}$$

- Obtenemos

$$\frac{3Z_1^2}{4} - Z_1Z_2 - \frac{Z_1}{2} + Z_2^2 - Z_2 + \frac{3}{4}$$

- Es equivalente a minimizar

$$-Z_1Z_2 - \frac{Z_1}{2} - Z_2$$

Binary Linear Programming

- Los problemas BLP (Binary Linear Programming) son del tipo

$$\begin{aligned} \text{Minimizar } & c_0x_0 + c_1x_1 + \dots + c_mx_m \\ \text{sujeto a } & Ax \leq b, \\ & x_j \in \{0, 1\}, \quad j = 0, \dots, m, \end{aligned}$$

- BLP es NP-hard
- ¡Tiene restricciones!

Problemas enteros con restricciones lineales

- Las restricciones se añaden como penalizaciones en la función a minimizar
- Una restricción de igualdad

$$\sum_{i=1}^n a_i x_i = b$$

se añade como el término de penalización

$$M \left(\sum_{i=1}^n a_i x_i - b \right)^2$$

donde M es una constante suficientemente grande

- Podemos aprovechar que $x_i^2 = x_i$ para simplificar

Problemas enteros con restricciones lineales (2)

- Una restricción de desigualdad

$$\sum_{i=1}^n a_i x_i \leq b$$

se transforma en una de igualdad añadiendo una variable de holgura entera $c \geq 0$

$$\sum_{i=1}^n a_i x_i + c = b$$

- Posteriormente, c (y las variables x_i , si son enteras) se transforma de acuerdo a

$$c = \sum_{k=0}^m c_k 2^k$$

donde c_i son variables binarias y m es suficientemente grande para poder expresar todos los valores válidos de c (el último valor podría ser menor que 2^m si hiciera falta)

Ejemplo: De BLP a QUBO

- Minimizar

$$2xy - y - 4zy$$

sujeto a

$$0 \leq x, y, z \leq 1$$

$$x + 2y + 3z \leq 5$$

- Añadimos $0 \leq c \leq 5$ y reescribimos

$$x + 2y + 3z + c = 5$$

- Sustituimos $c = c_0 + 2c_1 + 2c_2$ e incorporamos la penalización. Ahora queremos minimizar

$$2xy - y - 4zy + 100(x + 2y + 3z + c_0 + 2c_1 + 2c_2 - 5)^2$$

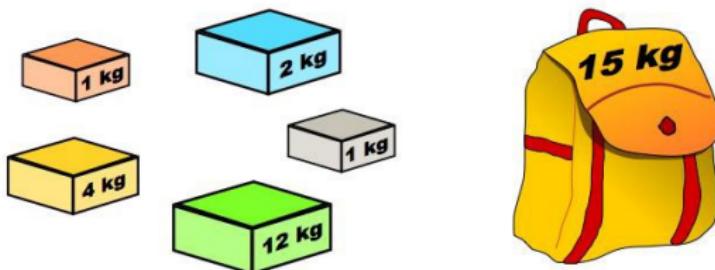
que es un problema QUBO (hemos elegido $M = 100$).

- También podríamos haber usado $c = c_0 + 2c_1 + 4c_2$

El problema de la mochila

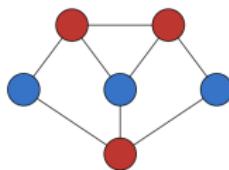
- Nos dan objetos $j = 0, \dots, m$, cada uno con peso w_j y valor c_j
- Queremos elegir objetos, maximizando el valor total, sin pasarnos de un peso máximo W
- Es un problema NP-hard
- Claramente, es un BLP:

$$\begin{aligned} & \text{Minimizar} && -c_0x_0 - c_1x_1 - \dots - c_mx_m \\ & \text{sujeto a} && w_0x_0 + w_1x_1 + \dots + w_mx_m \leq W, \\ & && x_j \in \{0, 1\}, \quad j = 0, \dots, m, \end{aligned}$$

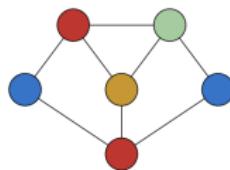


Coloreado de grafos

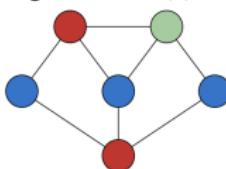
- Dado un grafo, queremos saber cuál es el menor número de colores necesarios para colorear sus vértices sin que dos vértices adyacentes reciban el mismo color
- Es NP-hard
- Decidir si un grafo se puede colorear con k colores es NP-completo (para k al menos 3)
- Tiene múltiples aplicaciones



(a) Invalid colouring



(b) Non-optimal colouring



(c) Optimal colouring

Versión QUBO del coloreado de grafos

- Queremos saber si un grafo de m vértices es coloreable con k colores
- Definimos variables binarias x_{jl} with $j = 0, \dots, m$ and $l = 0, \dots, k - 1$
- x_{jl} es 1 si y solo si el vértice j recibe el color l
- Restricción por cada vértice:

$$\sum_{l=0}^{k-1} x_{jl} = 1$$

- Restricción por cada par de vértices j y h adyacentes en el grafo:

$$\sum_{l=0}^{k-1} x_{jl} x_{hl} = 0$$

Versión QUBO del coloreado de grafos

- Versión QUBO:

$$\begin{aligned} \text{Minimizar} \quad & \sum_{j=0}^m \left(\sum_{l=0}^{k-1} x_{jl} - 1 \right)^2 + \sum_{(j,h) \in E} \sum_{l=0}^{k-1} x_{jl} x_{hl} \\ \text{sujeto a} \quad & x_{jl} \in \{0, 1\}, \quad j = 0, \dots, m, l = 0, \dots, k-1, \end{aligned}$$

- Si el mínimo es 0, el grafo es coloreable con k colores. Si no, no lo es.

El problema del viajante de comercio

- Hay una serie de ciudades $j = 0, \dots, m$ que tenemos que recorrer, minimizando el coste total
- El coste de viajar de j a l es w_{jl}
- Es un problema NP-hard



Versión QUBO del problema del viajante de comercio

- Variables binarias x_{jl}
- x_{jl} es 1 si y solo si la ciudad j se visita en la posición l
- Restricción por cada ciudad:

$$\sum_{l=0}^m x_{jl} = 1$$

- Restricción por cada posición:

$$\sum_{j=0}^m x_{jl} = 1$$

- Coste del viaje

$$\sum_{l=0}^{m-1} \sum_{j=0}^m \sum_{k=0}^m w_{jk} x_{jl} x_{kl+1}$$

Versión QUBO del problema del viajante de comercio

- Formulación completa:

$$\begin{aligned} \text{Minimizar} \quad & \sum_{l=0}^{m-1} \sum_{j=0}^m \sum_{k=0}^m w_{jk} x_{jl} x_{kl+1} \\ & + B \left(\sum_{l=0}^m x_{jl} - 1 \right)^2 + B \left(\sum_{j=0}^m x_{jl} - 1 \right)^2 \\ \text{sujeto a} \quad & x_{jl} \in \{0, 1\}, \quad j, l = 0, \dots, m, \end{aligned}$$

- Elegimos B suficientemente grande. Por ejemplo:

$$B = 1 + \sum_{j,k=0}^m w_{jk}$$

Otros problemas

frontiers in
PHYSICS

REVIEW ARTICLE
published: 12 February 2014
doi: 10.3389/fphy.2014.00005



Ising formulations of many NP problems

Andrew Lucas*

Lyman Laboratory of Physics, Department of Physics, Harvard University, Cambridge, MA, USA

Edited by:

Jacob Biamonte, ISI Foundation,
Italy

Reviewed by:

Mauro Faccin, ISI Foundation, Italy
Ryan Babbush, Harvard University,
USA

Bryan A. O'Gorman, NASA, USA

***Correspondence:**

Andrew Lucas, Lyman Laboratory of
Physics, Department of Physics,
Harvard University, 17 Oxford St.,
Cambridge, MA 02138, USA
e-mail: lucas@fas.harvard.edu

We provide Ising formulations for many NP-complete and NP-hard problems, including all of Karp's 21 NP-complete problems. This collects and extends mappings to the Ising model from partitioning, covering, and satisfiability. In each case, the required number of spins is at most cubic in the size of the problem. This work may be useful in designing adiabatic quantum optimization algorithms.

Keywords: spin glasses, complexity theory, adiabatic quantum computation, NP, algorithms

Quantum Information Processing (2022) 21:67
<https://doi.org/10.1007/s11128-021-03405-5>



Unconstrained binary models of the travelling salesman problem variants for quantum optimization

Özlem Salehi¹ · Adam Glos¹ · Jarosław Adam Miszczak¹

Received: 28 June 2021 / Accepted: 26 December 2021 / Published online: 22 January 2022
© The Author(s) 2022

Ejercicios

- Escribir la versión QUBO/Ising de los siguientes problemas:
 - Determinar si un grafo tiene un **camino Hamiltoniano** o no
 - **Vertex cover**: dado un grafo, encontrar un conjunto de vértices del menor tamaño posible tal que cada eje sea incidente en, al menos, uno de los vértices elegidos.
 - **Maximum clique**: dado un grafo, encontrar un conjunto de vértices de tamaño máximo tal que todos estén conectados entre sí
- ¿Cómo se podría utilizar un algoritmo para el problema de decisión del coloreado de grafos para determinar el **número cromático** de un grafo?

Parte III

Quantum Annealing

Computación cuántica adiabática

- ¿Cómo obtener el *ground state* de H ?
- Una solución natural es aplicar el propio hamiltoniano H para llegar a la solución
- El **teorema adiabático** nos asegura que si comenzamos en el estado de mínima energía de un hamiltoniano y lo vamos variando lentamente, nos mantendremos siempre en el estado de mínima energía
- La idea de la computación cuántica adiabática es:
 - Comenzar en el estado de mínima energía de un hamiltoniano sencillo H_i
 - Evolucionar el sistema hacia el estado de mínima energía del hamiltoniano del problema H_f
 - Para ello se aplica el hamiltoniano dependiente del tiempo

$$H(t) = \left(1 - \frac{t}{T}\right)H_i + \frac{t}{T}H_f$$

durante tiempo T

Computación cuántica adiabática (2)

- Para garantizar la adiabaticidad, T debe crecer como el inverso del cuadrado del *spectral gap* de $H(t)$ (diferencia entre el primer y segundo nivel de energía)
- El spectral gap es **difícil** de calcular
- En la práctica, se usa el *quantum annealing*:
 - Se toma $H_i = - \sum_{i=1}^n X_i$ (con ground state $\sum_{x=0}^{2^n-1} |x\rangle$)
 - Como H_f se toma un hamiltoniano de Ising
 - Se deja evolucionar durante un tiempo T (no necesariamente adiabático)
 - Se mide para obtener una solución
 - Se repite un cierto número de veces y se devuelve la mejor solución obtenida
- Es la base de los ordenadores cuánticos de D-Wave

Los ordenadores cuánticos de D-Wave

- Son ordenadores de propósito específico: resolver el modelo de Ising
- Accesibles gratuitamente (1 minuto/mes) a través de <https://www.dwavesys.com/take-leap>



Resolviendo problemas de optimización con D-Wave

- Utilizar los quantum annealers de D-Wave es muy sencillo
- Por ejemplo, tomemos el hamiltoniano

$$Z_0 Z_1 + Z_0 Z_2$$

- Lo podemos definir con el siguiente código:

```
import dimod
J = { (0,1):1, (0,2):1 }
h = {}
problem = dimod.BinaryQuadraticModel(h, J,
                                         0.0, dimod.SPIN)
```

Resolviendo problemas de optimización con D-Wave

- Para ejecutarlo, podemos usar las siguientes instrucciones:

```
from dwave.system import DWaveSampler  
from dwave.system import EmbeddingComposite  
sampler = EmbeddingComposite(DWaveSampler())  
result = sampler.sample(problem, num_reads=10)  
print(result)
```

- Obtendremos algo como:

```
    0   1   2   energy  num_oc.  chain_.  
0  +1  -1  -1     -2.0          6      0.0  
1  -1  +1  +1     -2.0          4      0.0  
['SPIN', 2 rows, 10 samples, 3 variables]
```

Trabajando con QUBO en D-Wave

- Podemos ejecutar problemas QUBO en D-Wave usando `dimod.BINARY` (las soluciones vendrán dadas como ceros y unos)
- También podemos definir directamente problemas declarando:
 - Variables
 - Función objetivo
 - Restricciones
- Por ejemplo, un problema como

$$\begin{aligned} \text{Minimize} \quad & -5x_0 + 3x_1 - 2x_2 \\ \text{subject to} \quad & x_0 + x_2 \leq 1, \\ & 3x_0 - x_1 + 3x_2 \leq 4 \\ & x_j \in \{0, 1\}, \quad j = 0, 1, 2 \end{aligned}$$

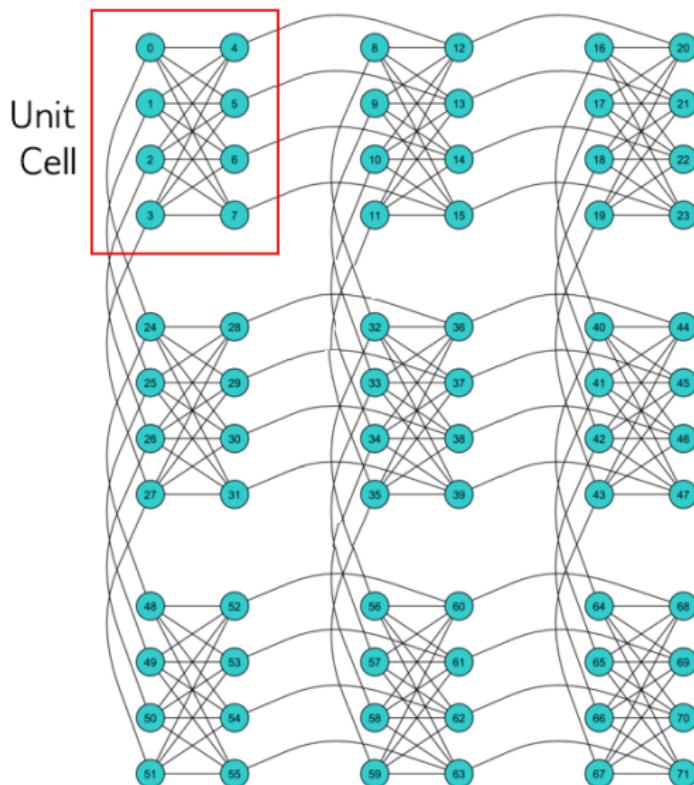
- Se aplican automáticamente las transformaciones que hemos estudiado

Los annealers de D-Wave

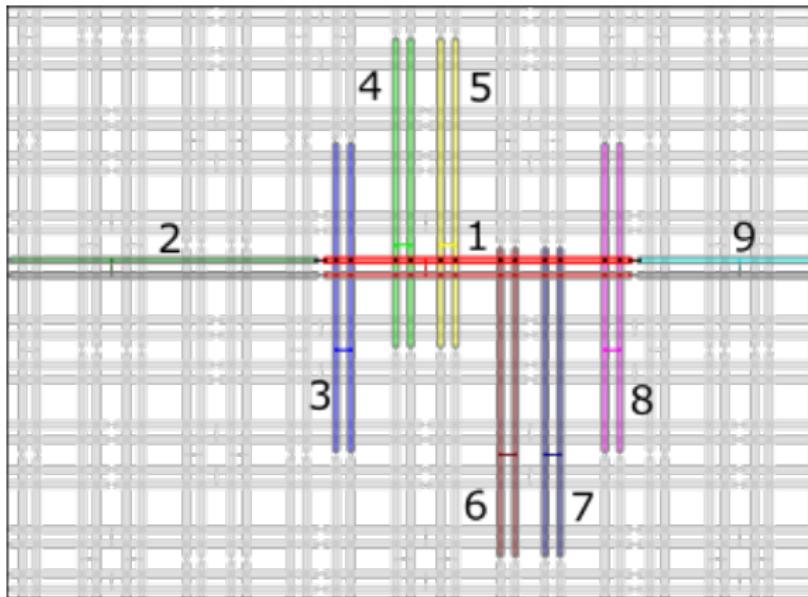
- D-Wave ofrece distintos annealers a través de Leap:
 - DWAVE 2000Q: 2048 qubits, topología Chimera
 - Advantage System: 5760 qubits, topología Pegasus
 - Advantage 2 (prototipo): 576 qubits, topología Zephyr
 - Hybrid solvers
- La topología es extremadamente importante:
- **NO** hay conexiones completas
- En algunos casos, no hay conexiones 3 a 3 (Chimera)
- Se necesitan cadenas para representar variables lógicas



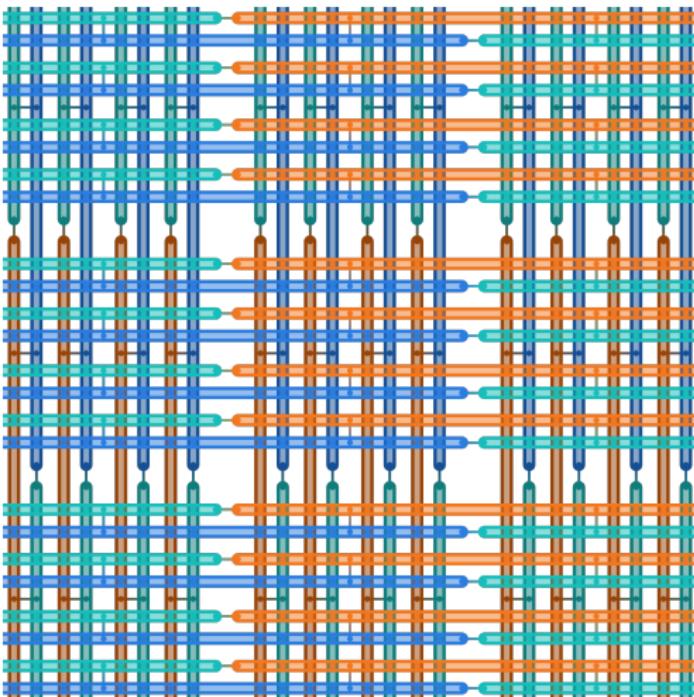
La topología Chimera



La topología Pegasus



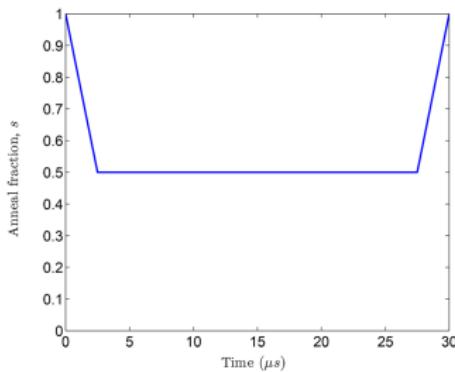
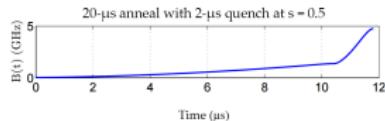
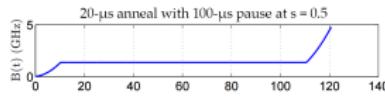
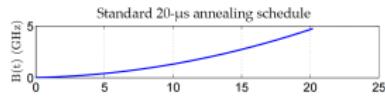
La topología Zephyr



- Si la topología no soporta directamente las conexiones de las variables lógicas del problema, hay que usar un embedding
- Encontrar un embedding es un problema NP-hard
- El software de D-Wave proporciona distintos métodos (heurísticos) para encontrar embeddings:
 - EmbeddingComposite (calcula un embedding en cada ejecución)
 - AutoEmbeddingComposite (primero intenta NO utilizar embedding)
 - FixedEmbeddingComposite (utiliza un embedding pasado por parámetro)
 - LazyEmbeddingComposite (calcula un embedding solo la primera vez y lo almacena)

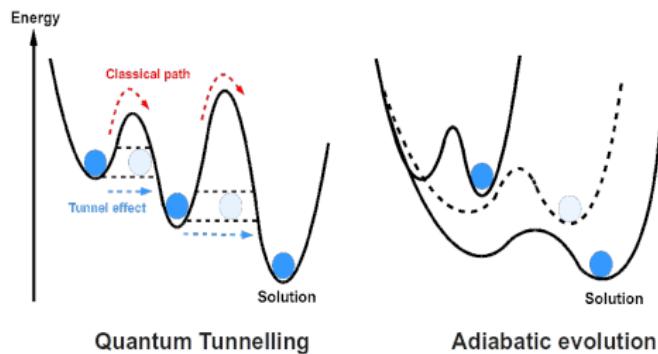
Parámetros del annealing

- Para realizar el annealing, se pueden tener en cuenta los siguientes parámetros:
 - Annealing time
 - Annealing schedule
 - Forward
 - Reverse
 - Coupling strength
 - Lagrange multiplier (en problemas con restricciones)



Hybrid solvers

- El software de D-Wave ofrece solvers clásicos:
 - Exact Solver
 - Simulated Annealing
 - Steepest Descent Solver
 - Tabu Solver
- Y solvers híbridos, que combinan recursos clásicos y cuánticos
- Para los solvers híbridos, es interesante la propiedad `quota_conversion_rate`



Ejercicios

- Crear un problema QUBO o Ising (por ejemplo, una instancia de MaxCut) con un número grande de variables (100 o más)
- Ejecutarlo en:
 - DWave 2000Q
 - Advantage System
 - Advantage System 2
 - Simulated annealing y otros solvers clásicos e híbridos
- Comparar:
 - Calidad de las soluciones
 - Tiempo de ejecución
- Repetir, incrementando el número de variables

Parte IV

QAOA: Quantum Approximate
Optimization Algorithm

Discretizando la computación adiabática

- En el modelo adiabático, partimos de un ground state y llevamos a cabo una evolución con

$$H(t) = A(t)H_0 + B(t)H_1$$

- H_1 es, por ejemplo, un hamiltoniano de Ising
- Podemos intentar discretizar la evolución mediante

$$e^{i\Delta t(A(t_c)H_0+B(t_c)H_1)}$$

en el intervalo $[t_c, t_c + \Delta t]$

- Tendríamos

$$\left(\prod_{m=0}^p e^{i\Delta t(A(t_m)H_0+B(t_m)H_1)} \right) |\psi_0\rangle,$$

con $t_m = m\frac{\Delta t}{T}$ y $p = \frac{T}{\Delta t}$

Discretizando la computación adiabática

- Realizamos una aproximación más:

$$e^{i\Delta t(A(t_c)H_0+B(t_c)H_1)} \approx e^{i\Delta tA(t_c)H_0} e^{i\Delta tB(t_c)H_1}$$

- Finalmente, tendremos

$$\prod_{m=0}^p e^{i\Delta tA(t_m)H_0} e^{i\Delta tB(t_m)H_1} |\psi_0\rangle$$

- Esta es la inspiración para el QAOA, que considera estados parametrizados de la forma

$$|\beta, \gamma\rangle = e^{i\beta_p H_0} e^{i\gamma_p H_1} \dots e^{i\beta_2 H_0} e^{i\gamma_2 H_1} e^{i\beta_1 H_0} e^{i\gamma_1 H_1} |\psi_0\rangle$$

con $p \geq 1$

QAOA: el algoritmo

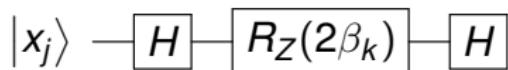
- QAOA es un método híbrido en el que se usan tanto un ordenador clásico como uno cuántico
- Los pasos son:
 - 1 Elegir un valor para p y unos ángulos iniciales β, γ
 - 2 Preparar el estado $|\beta, \gamma\rangle$
 - 3 Estimar la energía $E(\beta, \gamma) = \langle\beta, \gamma| H_1 |\beta, \gamma\rangle$ de $|\beta, \gamma\rangle$
 - 4 Si se cumplen los criterios de parada, terminar. Si no, variar β y γ para minimizar $E(\beta, \gamma)$ y volver a 2
- El paso 2 se realiza en el ordenador cuántico. El resto, en uno clásico.

Cómo preparar el estado $|\beta, \gamma\rangle$

- Tomamos $H_0 = \sum X_i$
- El estado $|\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |x\rangle$ se puede preparar con puertas de Hadamard
- Se cumple que

$$e^{i\beta_k H_0} = e^{-i\beta_k \sum_{j=0}^{n-1} X_j} = \prod_{j=0}^{n-1} e^{-i\beta_k X_j}$$

- Cada $e^{-i\beta_k X_j}$ es una rotación $R_X(2\beta_k)$ o equivalentemente



Cómo preparar el estado $|\beta, \gamma\rangle$

- Tomamos $H_1 = -\sum_{j,k} J_{jk} Z_j Z_k - \sum_j h_j Z_j$
- Entonces, tenemos

$$e^{i\gamma_I H_1} = e^{-i\gamma_I (\sum_{j,k} J_{jk} Z_j Z_k - \sum_j h_j Z_j)} = \prod_{j,k} e^{-i\gamma_I J_{jk} Z_j Z_k} \prod_j e^{-i\gamma_I h_j Z_j}.$$

- Podemos generalizar y considerar casos de la forma

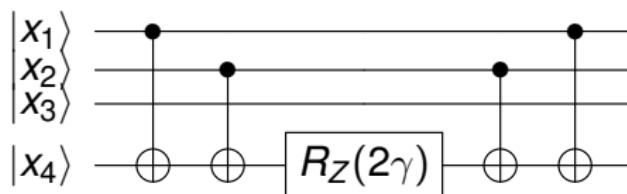
$$e^{-i\gamma_k Z_{i_1} \cdots Z_{i_j}}$$

Implementando $e^{-i\gamma_k Z_{i_1} \cdots Z_{i_j}}$

- Nótese que $e^{-i\gamma_k Z_{i_1} \cdots Z_{i_j}}$ es diagonal en la base computacional
- De hecho, para una cadena binaria $x = x_1 \dots x_n$ su acción en $|x\rangle$ es
 - $|x\rangle \rightarrow e^{-i\gamma_k} |x\rangle$ if $x_{i_1} \oplus \dots \oplus x_{i_j} = 0 \pmod{2}$
 - $|x\rangle \rightarrow e^{i\gamma_k} |x\rangle$ if $x_{i_1} \oplus \dots \oplus x_{i_j} = 1 \pmod{2}$
- Como se ve, es muy similar a una rotación R_Z
- Entonces, podemos:
 - Computar la paridad de $x_{i_1} \oplus \dots \oplus x_{i_j}$ con puertas CNOT
 - Aplicar $R_Z(2\gamma_k)$ en el qubit sobre el que hemos computado la paridad
 - Descomputar la paridad

Un ejemplo

- Imaginemos que estamos trabajando con 4 qubits y queremos implementar $e^{-i\gamma Z_1 Z_2 Z_4}$
- Podemos usar el siguiente circuito:

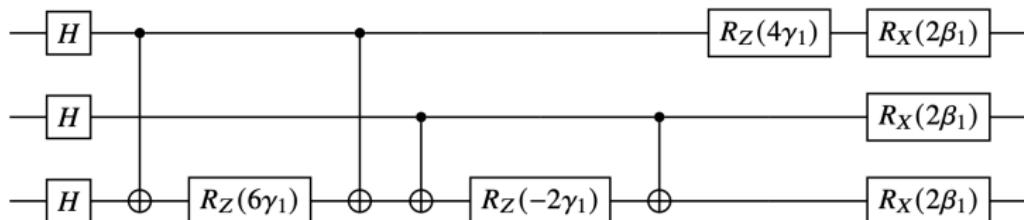


Ejemplo de circuito QAOA completo

- Consideramos el caso

$$3Z_0Z_2 - Z_1Z_2 + 2Z_0$$

- Obtenemos el siguiente circuito para $p = 1$



Estimando la energía

- Estimar la energía en el QAOA es muy sencillo
- Suponemos que partimos de un problema QUBO con función de coste $C(x)$, de donde obtenemos H_1
- Repetimos el siguiente proceso un número fijo de veces
 - ① Preparamos el estado $|\beta, \gamma\rangle$
 - ② Lo medimos para obtener un resultado x
 - ③ Calculamos $C(x)$y obtenemos la media de los resultados
- Este proceso funciona porque como

$$|\beta, \gamma\rangle = \sum_{x \in \{0,1\}^n} a_x |x\rangle$$

entonces

$$\langle \beta, \gamma | H_1 | \beta, \gamma \rangle = \sum_{x \in \{0,1\}^n} |a_x|^2 C(x)$$

- También es interesante guardar la x con el valor de $C(x)$ mínimo entre las que hemos medido

QUBO y HOBO

- Consideramos la fórmula booleana

$$(x_0 \vee \neg x_1 \vee x_2) \wedge (\neg x_0 \vee x_1 \vee \neg x_2) \wedge (x_0 \vee x_1 \vee x_2)$$

- A partir de ella, podemos obtener el polinomio

$$(1 - x_0)x_1(1 - x_2) + x_0(1 - x_1)x_2 + (1 - x_0)(1 - x_1)(1 - x_2)$$

- La fórmula original es satisfacible si y solo si el mínimo del polinomio es 0
- Es un polinomio de grado 3
- Tenemos un problema HOBO (Higher Order Binary Optimization) o PUBO (Polynomial Unconstrained Binary Optimization)
- Lo podemos resolver también con QAOA (nuestros desarrollos se aplican también a estos casos)

Algunos comentarios sobre el QAOA

- Para que el proceso sea eficiente, H_1 debe tener un número de términos $e^{-i\gamma_k Z_{i_1} \cdots Z_{i_j}}$ que sea polinomial en el número de qubits n y en el número m de términos de $C(x)$
- Si un término $C_a(x)$ sólo involucra k bits, su traducción H_a involucrará sólo k matrices Pauli Z_i
- Por tanto, si p es constante e independiente de n y m , y todas los términos involucran a lo sumo k bits (constante también independiente de n y m) entonces el número de puertas será polinomial en n y m
- Es el caso, por ejemplo, de Max-Cut
- Cuando $p \rightarrow \infty$, el estado de mínima energía de $|\beta, \gamma\rangle$ tiende al estado de mínima energía de H_1
- En algunos casos, se han obtenido resultados interesantes incluso para p pequeño

Nuevos resultados sobre la eficacia del QAOA

The Quantum Approximate Optimization Algorithm at High Depth for MaxCut on Large-Girth Regular Graphs and the Sherrington-Kirkpatrick Model

✉ Joao Basso¹, Edward Farhi^{1,2}, ✉ Kunal Marwaha³,
✉ Benjamin Villalonga¹, and ✉ Leo Zhou⁴

¹ Google Quantum AI, Venice, CA 90291

² Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139

³ Department of Computer Science, University of Chicago, Chicago, IL 60637

⁴ Walter Burke Institute for Theoretical Physics, California Institute of Technology, Pasadena, CA 91125

October 26, 2021

Abstract

The Quantum Approximate Optimization Algorithm (QAOA) finds approximate solutions to combinatorial optimization problems. Its performance monotonically improves with its depth p . We apply the QAOA to MaxCut on large-girth D -regular graphs. We give an iterative formula to evaluate performance for any D at any depth p . Looking at random D -regular graphs, at optimal parameters and as D goes to infinity, we find that the $p = 11$ QAOA beats all classical algorithms (known to the authors) that are free of unproven conjectures. While the iterative formula for these D -regular graphs is derived by looking at a single tree subgraph, we prove that it also gives the ensemble-averaged performance of the QAOA on the Sherrington-Kirkpatrick (SK) model. Our iteration is a compact procedure, but its computational complexity grows as $O(p^2 4^p)$.

Variantes de QAOA

PHYSICAL REVIEW RESEARCH 4, 033029 (2022)

www.nature.com/scientificreports/

Adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer

Linghua Zhu^{1,2}, Ho Lun Tang¹, George S. Barron³, F. A. Calderon-Vargas,¹ Nicholas J. Mayhall,¹ Edwin Barnes,¹ and Sophia E. Economou,^{1,4}
¹Department of Physics, Virginia Tech, Blacksburg, Virginia 24061, USA
²Department of Chemistry, Virginia Tech, Blacksburg, Virginia 24061, USA

(Received 25 May 2020; revised 22 December 2020; accepted 17 June 2022; published 11 July 2022)

The quantum approximate optimization algorithm (QAOA) is a hybrid variational quantum-classical algorithm that solves combinatorial optimization problems. While there is evidence suggesting that the fixed form of the standard QAOA Ansatz is not optimal, there is no systematic approach for finding better Ansätze. We address this problem by developing an iterative version of QAOA that performs gradient descent. This Ansatz can be applied to specific quantum computers. We simulate the algorithm on a class of Max-Cut graph problems and show that it converges much faster than the standard QAOA, while simultaneously reducing the required number of CNOT gates and optimization parameters. We provide evidence that this speedup is connected to the concept of short-cut adiabaticity.

DOI: 10.1103/PhysRevResearch.4.033029

I. INTRODUCTION

Many important computationally hard combinatorial optimization problems such as Max-Cut, graph coloring, traveling salesman, and scheduling problems [1–3] can be reduced to Ising Hamiltonians whose ground state provide the solutions. One can in principle solve these optimization problems on a quantum computer by initializing the quantum device in the ground state of a Hamiltonian that is easy to prepare and adiabatically tuning the latter into the problem Hamiltonian. In a digital quantum computer, this translates into a Trotterized version of the adiabatic evolution operator, which is the alternating product of two evolution operators corresponding to the Trotterizer and the problem (Ising) Hamiltonians. In the limit of an infinite product, this Trotterized form becomes exact.

QAOA is a hybrid quantum-classical variational algorithm that uses a finite-order version of this evolution operator to prepare wavefunction ansatze on a quantum processor [4–11]. QAOA is performed by iteratively minimizing the expectation value of the Ising Hamiltonian with respect to the parameters in the Ansatz. The quantum processor is also used to measure energy expectation values, while the optimization is done on a classical computer. There has been a lot of

QAOA may provide a significant quantum advantage over classical algorithms [14–16], and that it is computationally universal [17,18].

Despite these advances, there are limitations and potential issues with this algorithm. The performance improves with the number of alternating layers in the Ansatz, but the latter is limited by coherence times in existing and near-term quantum processors. Moreover, more layers implies more operations, which increases the cost function evaluated by the classical optimizer [19]. Furthermore, Ref. [20] points out that the locality and symmetry of QAOA can also severely limit its performance. These issues can be attributed to, or are at least exacerbated by, the form of the QAOA Ansatz. In particular, short-depth QAOA is not really the digitized version of the Trotterized procedure used at adiabatic Ansatz, and its small should not be expected to perform well, usually even well. A short-depth Ansatz that is further taskified to a given combinatorial problem could therefore address the issues with the standard QAOA Ansatz. However, identifying such an alternative is a highly nontrivial problem given the vast space of possible Ansätze.

In this work, we propose an iterative version of QAOA termed adaptive derivative assembled problem tailored-

scientific reports

OPEN

Multi-angle quantum approximate optimization algorithm

Rebekah Herrman^{1,2}, Phillip C. Lotshaw^{2,3}, James Ostrowski¹, Travis S. Humble², and George Siopsis⁴

The quantum approximate optimization algorithm (QAOA) generates an approximate solution to combinatorial optimization problems using a variational circuit circuit defined by parameterized layers of quantum evolution. In theory, the algorithm can find an optimal solution among circuits with both gate sets and different complexity orders. In practice, however, we investigate a multi-angle ansatz for QAOA that reduces circuit depth and improves the approximation ratio by increasing the number of classical parameters. Even though the number of parameters increases, our results indicate that good parameters can be found in polynomial time for a test dataset we consider. This new ansatz is also shown to be significantly more efficient than the standard QAOA ansatz in terms of circuit resources over QAOA. The thermal performance is lower bounded by the conventional ansatz, and we present empirical results for graphs on eight vertices that one layer of the multi-angle ansatz is comparable to three layers of the traditional ansatz on MaxCut problems. Similarly, multi-angle QAOA yields a higher approximation ratio than QAOA on a collection of MaxCut instances on fifty and one-hundred vertex graphs. Most of the optimal parameters are forced to be zero, so their associated circuit can be removed from the circuit, further decreasing the circuit depth. These results indicate that multi-angle QAOA requires shallower circuits to solve problems than QAOA, making it more viable for near-term intermediate-scale quantum devices.

Among several quantum algorithms implemented on noisy intermediate-scale quantum (NISQ) devices^{1,2}, the quantum approximate optimization algorithm (QAOA) offers a promising route to approximately solve combinatorial optimization problems such as Max-Cut, Max-Satisfiability, and Max-k-Clique. QAOA takes a set of classical parameters to optimize the cost function and an initial state. Variations to the original algorithm include alternative operators and initial states³, while many classical aspects such as the parameter optimization and problem encoding are shared. However, the original QAOA has a major limitation: it requires many layers of QAOA to reach large numbers of qubits and deep circuits⁴. For example, a recent study has developed a systematic set of parameters that are argued to require $p = 30$ layers of QAOA to reach performance comparable to the conventional Goemans-Williamson algorithm on MaxCut⁵, while another paper has argued that hundreds of qubits are needed to outperform QAOA⁶. Not only does the number of layers increase exponentially with circuit depth and affects the fidelity of the prepared quantum state so the performance that can be achieved from near-term quantum computers at these depths is questionable.

One way to address this issue is to increase the number of classical parameters introduced in each layer, a variation that we term multi-angle QAOA (ma-QAOA). This approach was originally introduced by Ref. [7]. Increasing the number of classical parameters allows for fine-grain control over the optimization of the cost function and the approximation ratio, which measures optimality relative to the known best solution. The main idea behind ma-QAOA is that the additional parameters, which correspond to the responding reduction in circuit depth preserves the critical resource of the quantum state. In addition, finding the absolute optimal angles is not necessary in order to see an improvement over QAOA.

Here, we quantify the advantages of using multiple parameters for each layer of QAOA. First, we prove that

- Formular como QUBO/HOBO el problema del **Set Cover**: Dada una colección de conjuntos $S = \{S_1, \dots, S_n\}$, encontrar el menor número de conjuntos cuya unión incluye todos los elementos incluidos en al menos uno de los S_i .
- Tomar un problema sencillo (por ejemplo, una instancia de Max Cut en un grafo no muy grande)
- Ejecutarlo
 - Con QAOA y distintos valores de p en simuladores
 - Con quantum annealing
- Comparar:
 - Calidad de las soluciones (en función de p)
 - Tiempos (solo QAOA)

Parte V

VQE: Variational Quantum
Eigensolver

VQE: Variational Quantum Eigensolver

- QAOA puede verse como un caso particular de un algoritmo más general: el Variational Quantum Eigensolver (VQE)
- Ahora, tenemos un hamiltoniano general H_f (con un número polinomial de términos) y queremos aproximar su estado de mínima energía
- En lugar del estado parametrizado $|\beta, \gamma\rangle$ del QAOA usaremos
 - Un estado inicial $|\psi\rangle$ fácil de preparar (podría ser simplemente $|0\rangle$)
 - Una transformación unitaria parametrizada $U(\theta)$ que se suele llamar **forma variacional**
- Con ellos, creamos un ansatz

$$|\psi(\theta)\rangle = U(\theta) |\psi\rangle$$

y tratamos de minimizar su energía con respecto a H_f variando los parámetros

El principio variacional

- Como H_f es una matriz hermitiana, tiene autovalores reales λ_i y una base asociada de autovectores ortonormales $|\phi_i\rangle$
- Así, podemos escribir $|\psi(\theta)\rangle$ como una combinación lineal

$$|\psi(\theta)\rangle = \sum_i \alpha_i |\phi_i\rangle$$

- La energía de $|\psi(\theta)\rangle$ es

$$\langle \psi(\theta) | H_f | \psi(\theta) \rangle = \sum_i |\alpha_i|^2 \lambda_i$$

- Si λ_{min} es el menor de esos autovalores, entonces

$$\min_{\theta} \langle \psi(\theta) | H_f | \psi(\theta) \rangle \geq \lambda_{min}$$

Aproximando el estado de mínima energía con VQE

- VQE es también un algoritmo híbrido en el que se usa el ordenador cuántico en colaboración con un ordenador clásico
- Los pasos son:
 - ① Elegir un estado inicial $|\psi\rangle$, una forma variacional $U(\theta)$ y un vector de parámetros inicial θ
 - ② Preparar el estado $|\psi(\theta)\rangle = U(\theta) |\psi\rangle$
 - ③ Estimar la energía $E(\theta) = \langle\psi(\theta)| H_f |\psi(\theta)\rangle$ de $|\psi(\theta)\rangle$
 - ④ Si se cumplen los criterios de parada, terminar. Si no, variar θ para minimizar $E(\theta)$ y volver a 2
- El paso 2 se realiza en el ordenador cuántico. El resto, en uno clásico.

Estimando la energía de un estado

- El hamiltoniano siempre puede ser expresado como una combinación lineal de productos tensoriales de Paulis
- Por ejemplo

$$H_f = \frac{1}{4} Z_1 Z_3 - 3 X_1 Y_3 Z_4$$

- Dado $|\psi\rangle$, podemos usar la linearidad y evaluar

$$\langle \psi | H_f | \psi \rangle = \frac{1}{4} \langle \psi | Z_1 Z_3 | \psi \rangle - 3 \langle \psi | X_1 Y_3 Z_4 | \psi \rangle$$

- Para estimar $\langle \psi | Z_1 Z_3 | \psi \rangle$ podemos medir $|\psi\rangle$ en la base computacional y promediar las energías de los resultados (que serán 1 o -1 para cada resultado individual).

Estimando la energía de un estado (2)

- Para estimar $\langle \psi | X_1 Y_3 Z_4 | \psi \rangle$ podemos tener en cuenta que

$$X = HZH$$

y

$$Y = SHZHS^\dagger$$

- Entonces $\langle \psi | X_1 Y_3 Z_4 | \psi \rangle$ es igual

$$\langle \psi | (H \otimes I \otimes SH \otimes I) Z_1 Z_3 Z_4 (H \otimes I \otimes HS^\dagger \otimes I) | \psi \rangle$$

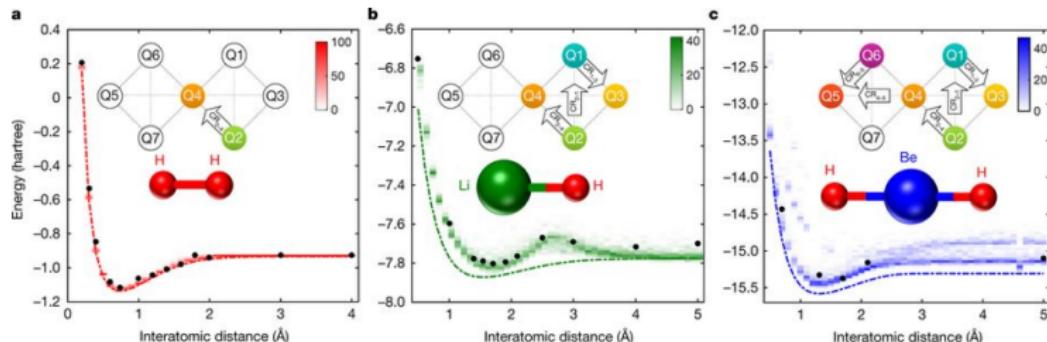
- Por tanto, podemos medir la energía en $Z_1 Z_3 Z_4$ de $(H \otimes I \otimes HS^\dagger \otimes I) | \psi \rangle$ porque

$$\left((H \otimes I \otimes HS^\dagger \otimes I) | \psi \rangle \right)^\dagger = \langle \psi | (H \otimes I \otimes SH \otimes I)$$

- Esto es equivalente a una medida en una base diferente

Estudiando moléculas con VQE

- VQE ha sido utilizado para estudiar los estados de energía mínima de varias moléculas
- El hamiltoniano químico (fermiónico) tiene que traducirse en un hamiltoniano de qubits (Jordan-Wigner, Bravyi-Kitaev...)
- Se usa información del problema para:
 - El estado inicial (estado del vacío $|0\rangle$, Hartree-Fock...)
 - La forma variacional (Unitary Coupled-Cluster Single and Double excitations...)



Encontrando estados excitados

- También podemos usar VQE para encontrar estados excitados (vectores propios que no son de mínima energía)
- Una vez que tenemos el estado de mínima energía $|\psi_0\rangle = U(\theta_0) |\psi\rangle$, consideramos el hamiltoniano

$$H'_f = H_f + C |\psi_0\rangle \langle \psi_0|$$

- Así, tenemos que $\langle \varphi | H'_f | \varphi \rangle$ es

$$\langle \varphi | H_f | \varphi \rangle + C \langle \varphi | \psi_0 \rangle \langle \psi_0 | \varphi \rangle = \langle \varphi | H_f | \varphi \rangle + C |\langle \psi_0 | \varphi \rangle|^2$$

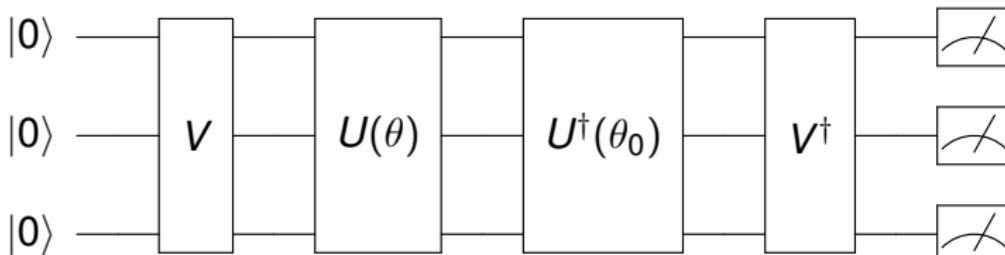
- Si C es mayor que la diferencia entre la energía mínima y el siguiente nivel de energía de H_f , entonces $|\psi_0\rangle$ ya no es estado de mínima energía de H'_f

Calculando productos escalares de estados parametrizados

- Para calcular el producto escalar en la nueva expresión de la energía podemos tener en cuenta que $|\psi_0\rangle = U(\theta_0) |\psi\rangle$ y que el nuevo estado será de la forma $|\varphi\rangle = U(\theta) |\psi\rangle$ para algún θ
- Entonces, es fácil estimar $|\langle\psi_0|\varphi\rangle|^2$ ejecutando el circuito de la figura y calculando la frecuencia relativa de $|0\rangle$ porque

$$|\langle\psi_0|\varphi\rangle|^2 = |\langle 0| V^\dagger U(\theta_0)^\dagger U(\theta) V |0\rangle|^2$$

donde V es una unitaria tal que $V|0\rangle = |\psi\rangle$



Ejercicios

- Usar VQEUCFactory para calcular el perfil de disociación de la molécula de dihidrógeno
- Probar VQEUCFactory en ejecución con ruido
- Utilizar VQE con un problema de optimización sencillo (un ejemplo de Max Cut con un grafo pequeño, por ejemplo)
- Ejecutar en:
 - Simulador
 - Simulador con ruido
 - Simulador con ruido y mitigaciones de errores
 - Ordenador cuántico
 - Ordenador cuántico con mitigación de errores
- Repetir con QAOA

Parte VI

Conclusiones

Conclusiones

- Los ordenadores cuánticos de los que disponemos hoy en día son o de propósito específico (quantum annealers) o NISQ
- Aún así, se están estudiando sus aplicaciones en:
 - Optimización
 - Simulación química y física
 - Quantum Machine Learning
- Las herramientas principales son:
 - Modelos QUBO/Ising
 - Circuitos variacionales
- Campo fascinante pero aún no entendido por completo

El futuro está por construir

