CPEN 400Q Lecture 16
Solving combinatorial optimization problems with the Quantum Approximate Optimization Algorithm (part 2)

Friday 10 March 2023

Announcements

- Quiz 7 beginning of class Monday
- Assignment 2 due Monday March at 23:59
- Updated class schedule:
  - Monday March 13: in person
  - Friday March 17: pre-recorded "infotainment" lecture about compilation

Canvas

We started exploring how optimization problems can be formulated as energy minimization problems:

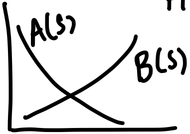
$$\min_{\vec{x}} \ \text{cost}(\vec{x}) \quad \text{subject to constraints}(\vec{x})$$

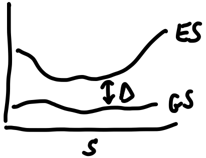
Optimization	**Physical system**
$\vec{x}$	State of the system
$\text{cost}(\vec{x})$	Hamiltonian
Optimum $\vec{x^*}$	Ground state
$\text{cost}(\vec{x^*})$	Ground state energy

General adiabatic quantum computing:

1. Design a Hamiltonian $\hat{H}_c$ whose ground state represents the solution to our optimization problem
2. Prepare a system in a easy-to-prepare ground state of a mixer Hamiltonian $\hat{H}_m$
3. Perform adiabatic evolution to transform the system from the ground state of the "easy" Hamiltonian to the ground state of the problem Hamiltonian

$$\hat{H}(s) = A(s)\hat{H}_m + B(s)\hat{H}_c$$





$\text{speed} \propto \frac{1}{\Delta^2}$

Last time

What we will do:

1. Design a Hamiltonian $\hat{H}_c$ whose ground state represents the solution to our optimization problem
2. Prepare a system in a easy-to-prepare ground state of a mixer Hamiltonian $\hat{H}_m$
3. Run the quantum approximate optimization algorithm

Last time

QAOA is a gate-model algorithm that can obtain approximate solutions to combinatorial optimization problems.

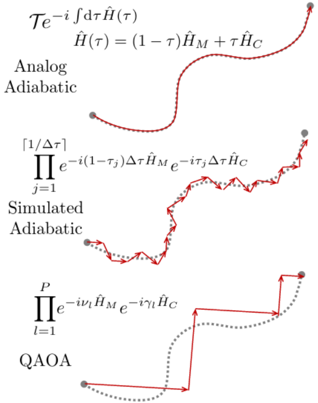
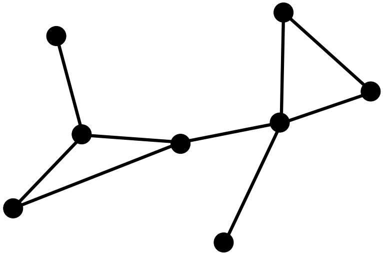
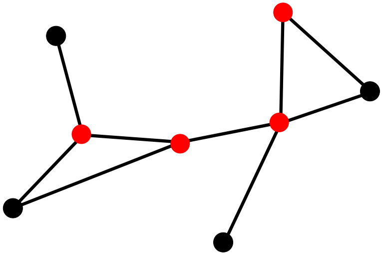


Image credit: G. Verdon, M. Broughton, J. Biamonte. *A quantum algorithm to train neural networks using low-depth circuits.* `https://arxiv.org/abs/1712.05304`

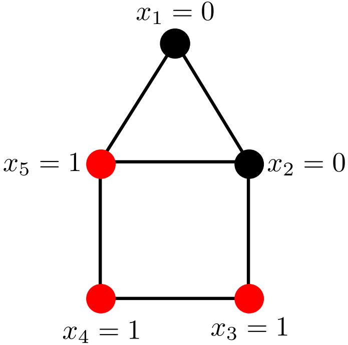
Minimum vertex cover: Given a graph $G = (V, E)$, what is the *smallest number of vertices* you can colour such that every edge in the graph is attached to at least one coloured vertex?



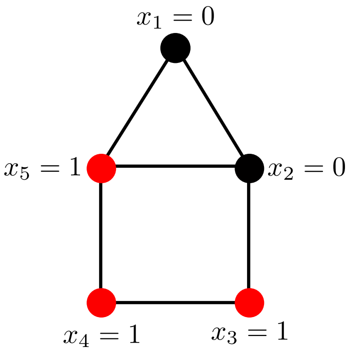


Last time

We turned this into an optimization problem over binary variables.



Last time

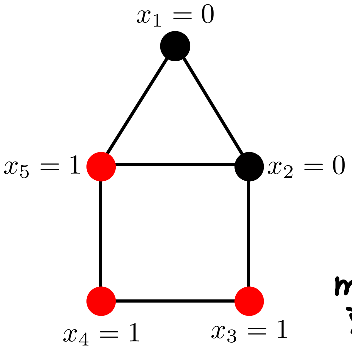


We defined a cost function term over a pair of vertices that is minimized for valid colourings:

$$f(x_i, x_j) = (1 - x_i)(1 - x_j)$$

Then for the whole graph,

$$\min_{\bar{x}} \sum_{ij \in E} (1 - x_i)(1 - x_j)$$



We finished by adding a penalty term to minimize the number of coloured vertices:

$$\sum_{i \in V} x_i$$

The full cost function is:

$$\min_{\vec{x}} \left(\sum_{ij \in E} (1 - x_i)(1 - x_j) + \sum_{i \in V} x_i \right)$$

Learning outcomes

- Convert cost functions of simple graph theory problems to Hamiltonians
- Distinguish between cost and mixer Hamiltonians and state the key requirements for the latter type
- Solve combinatorial optimization problems with QAOA in PennyLane

Game plan

1. Design a Hamiltonian $\hat{H}_c$ whose ground state represents the solution to our optimization problem
2. Prepare a system in a easy-to-prepare ground state of a mixer Hamiltonian $\hat{H}_m$
3. Run the quantum approximate optimization algorithm

Hamiltonian translation

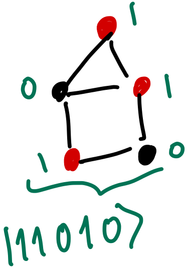
Our full cost function over binary variables is

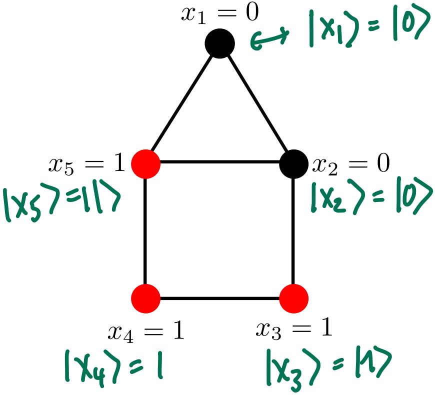
$$\min_{\vec{x}} \left(\sum_{ij \in E} (1 - x_i)(1 - x_j) + \sum_{i \in V} x_i \right)$$

Next steps:

1. Turn this into a Hamiltonian over qubits
2. Find its minimum energy

How should we do a mapping from this to qubits?

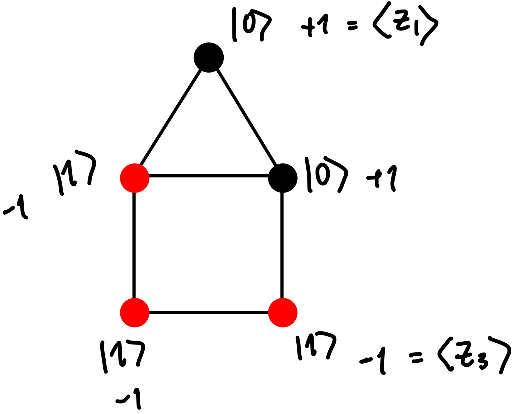




Hamiltonian translation

$$Z|0\rangle = +1\,|0\rangle$$
$$Z|1\rangle = -1\,|1\rangle$$

What should we use to represent cost?



$$x_i \in \{0, 1\} \rightarrow \{1, -1\}$$

Mathematically, we can make the mapping

$$x_i \rightarrow \frac{1}{2}(1 - z_i), \quad z_i \in \{1, -1\}$$

This associates

- $x_i = 0$ to $z_i = 1$ (corresponds to $|0\rangle$)
- $x_i = 1$ to $z_i = -1$ (corresponds to $|1\rangle$)

A complete derivation of the cost function is provided as an appendix at the end of the lecture slides.

Hamiltonian translation

We will take the result:

$$\sum_{ij\in E}(1-x_i)(1-x_j)+\sum_{i\in V}x_i \longrightarrow \sum_{ij\in E}(z_i+z_j+z_iz_j)-2\sum_{i\in V}z_i$$

$\langle\hat{H}\rangle$

Remember what the z_i represent; how can we express this cost function as a Hamiltonian?

$$\hat{H}=\sum_{ij\in E}(Z_i+Z_j+Z_iZ_j)-2\sum_{i\in V}Z_i$$

$$I\otimes I\otimes\cdots\otimes \underset{i-1}{I}\otimes \underset{i}{Z}\otimes \underset{i+1}{I}\otimes\cdots$$

$$\hat{H} = \hat{A} + \hat{B} \qquad \langle \hat{H} \rangle = \langle \hat{A} \rangle + \langle \hat{B} \rangle$$

$$\boxed{\hat{H}_c = \sum_{ij \in E} (Z_i + Z_j + Z_i Z_j) - 2 \sum_{i \in V} Z_i}$$

This makes sense:

$$\langle \hat{H}_c \rangle = \sum_{ij \in E} \left(\langle Z_i \rangle + \langle Z_j \rangle + \langle Z_i Z_j \rangle \right) - 2 \sum_{i \in V} \langle Z_i \rangle$$

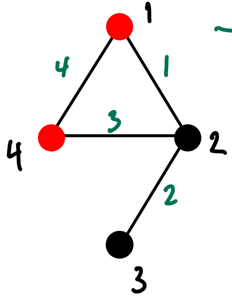
More generally, we have weight the two term differently:

$$\hat{H}_c = \gamma \sum_{ij \in E} (Z_i + Z_j + Z_i Z_j) - 2\lambda \sum_{i \in V} Z_i$$

$$\hat{H}_c = \sum_{ij \in E} (Z_i + Z_j + Z_i Z_j) - 2 \sum_{i \in V} Z_i$$

$$Z_i Z_j \downarrow Z_i \otimes Z_j$$

Try it: what is the energy of this *invalid* colouring?



$|1001\rangle$

Edge	Z_i	Z_j	Z_iZ_j	Σ
1	-1	1	-1	-1
2	1	1	1	3
3	1	-1	-1	-1
4	-1	-1	1	-1

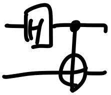
$=0$

V	Z_i
1	-1
2	1
3	1
4	-1

$=0$

$\langle \hat{H}_c \rangle = 0$

Hamiltonian translation



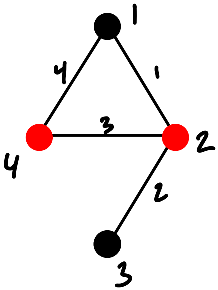
$\langle Z_1 \rangle$
$\langle Z_2 \rangle$
$\langle Z_1 \otimes Z_2 \rangle$

$$\hat{H}_c = \sum_{ij \in E} (Z_i + Z_j + Z_i Z_j) - 2 \sum_{i \in V} Z_i$$

$Z_i \otimes Z_j$

$\langle Z_i \otimes Z_j \rangle \neq \langle Z_i \rangle \langle Z_j \rangle$ in general

Try it: what is the energy of this *valid* colouring?



$\langle \hat{H}_c \rangle = -4$

$|0101\rangle$

$A + B \rightarrow \langle A + B \rangle = \langle A \rangle + \langle B \rangle$

Game plan

1. Design a Hamiltonian $\hat{H}_c$ whose ground state represents the solution to our optimization problem
2. Prepare a system in a easy-to-prepare ground state of a mixer Hamiltonian $\hat{H}_m$
3. Run the quantum approximate optimization algorithm

Mixer Hamiltonians

Did not derive in class; including here for completion

We also need a *mixer* Hamiltonian. The mixer must have a special property: it *cannot commute* with the cost Hamiltonian.

$$\hat{H}_c = \sum_{ij \in E} (Z_i + Z_j + Z_i Z_j) - 2 \sum_{i \in V} Z_i$$

$\hat{H}_c$ is just a diagonal matrix, and its eigenstates are the computational basis states.

$$\hat{H}_c |\vec{z}\rangle = E_z |\vec{z}\rangle, \quad \vec{z} \in \{0,1\}^n$$

Any state can be expressed in terms of the computational basis:

$$|\psi\rangle = \sum_{\vec{z}} \alpha_{\vec{z}} |\vec{z}\rangle$$

Evolve this under the cost Hamiltonian:

$$e^{-it\hat{H}_c}|\psi\rangle = e^{-it\hat{H}_c} \sum_{\vec{z}} \alpha_{\vec{z}} |\vec{z}\rangle$$
$$= \sum_{\vec{z}} \alpha_{\vec{z}} e^{-it\hat{H}_c} |\vec{z}\rangle$$
$$= \sum_{\vec{z}} \alpha_{\vec{z}} e^{-itE_{\vec{z}}} |\vec{z}\rangle$$

Have we actually changed anything?

Original state:

$$|\psi\rangle = \sum_{\vec{z}} \alpha_{\vec{z}} |\vec{z}\rangle \to \Pr(\vec{z}) = \alpha_{\vec{z}} \alpha_{\vec{z}}^* = |\alpha_{\vec{z}}|^2$$

New state:

$$e^{-it\hat{H}_c}|\psi\rangle = \sum_{\vec{z}} \alpha_{\vec{z}} e^{-itE_{\vec{z}}} |\vec{z}\rangle \to \Pr(\vec{z}) = \alpha_{\vec{z}} e^{-itE_{\vec{z}}} \cdot \alpha_{\vec{z}}^* e^{itE_{\vec{z}}}$$

$$= |\alpha_{\vec{z}}|^2$$

Simply evolving under the cost Hamiltonian doesn't change the probability distribution of the state.

If $\hat{H}_m$ commutes with $\hat{H}_c$, then $\hat{H}_c$ and $\hat{H}_m$ have a shared set of eigenvectors so evolving under $\hat{H}_m$ doesn't affect the state either.

Need a mixer which *does not commute* with $\hat{H}_c$. Something like

$ZX - XZ \neq 0$

$$\hat{H}_m = \sum_i X_i$$

$|++\cdots+\rangle$

Uniform superposition is an "easy to prepare" eigenstate of $\hat{H}_m$.

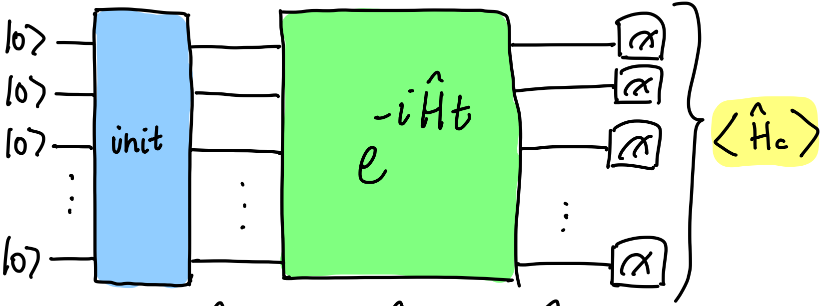
X: $|+\rangle$ $+1$, $|-\rangle$ -1

Apply Hadamards

Game plan

1. Design a Hamiltonian $\hat{H}_c$ whose ground state represents the solution to our optimization problem
2. Prepare a system in a easy-to-prepare ground state of a mixer Hamiltonian $\hat{H}_m$
3. Run the quantum approximate optimization algorithm

Initial idea: apply the unitary that evolves the Hamiltonian?

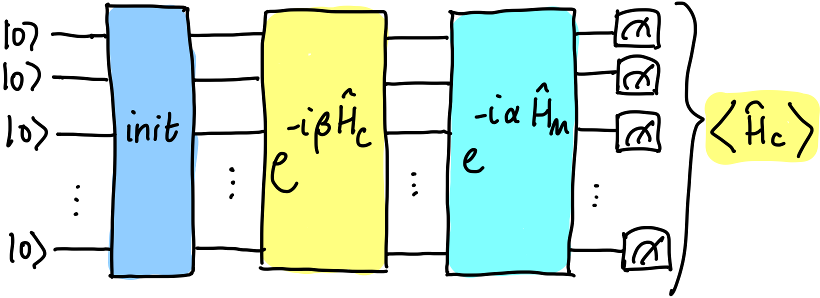


$$\hat{H} = \alpha \hat{H}_m + \beta \hat{H}_c$$

Hamiltonian simulation

How do we implement this circuit?

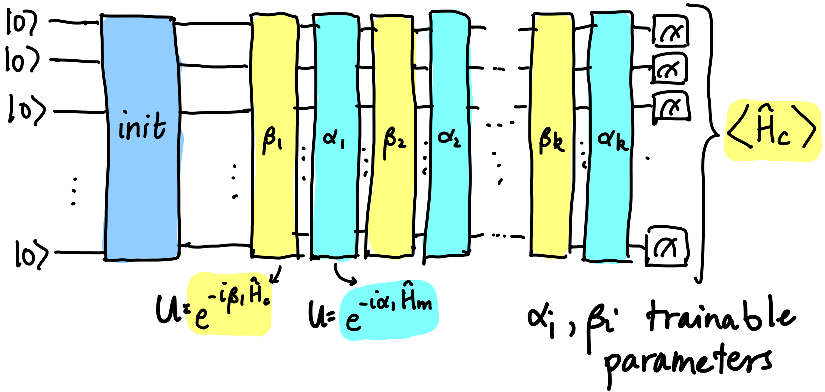
You might think that since $\hat{H}$ is a sum of terms...



But this is only true when $\hat{H}_c$ and $\hat{H}_m$ *commute*.

QAOA

QAOA does something similar to this but instead of applying each block for a fixed “time”, “time” is a trainable parameter.



Let’s implement this, and find parameters that minimize the cost.

Next time

Content:

- Density matrices and mixed states
- Noise in quantum computing

Action items:

1. Explore PennyLane's built-in QAOA module
2. Assignment 2
3. Work on final project

Recommended reading:

- Original QAOA paper `https://arxiv.org/abs/1411.4028`
- PennyLane Intro to QAOA tutorial `https://pennylane.ai/qml/demos/tutorial_qaoa_intro.html`
- Qiskit QAOA tutorial `https://qiskit.org/textbook/ch-applications/qaoa.html`

Full derivation: Hamiltonian translation

In order

We will make the mapping

$$x_i \to \frac{1}{2}(1 - z_i), \quad , z_i \in \{-1, 1\}$$

This associates $x_i = 0$ to $z_i = 1$ (corresponds to $|0\rangle$), and $x_i = 1$ to $z_i = -1$ (corresponds to $|1\rangle$).

Full derivation: Hamiltonian translation

Let's expand our cost function and make this substitution.

$$\sum_{ij \in E} (1 - x_i)(1 - x_j) + \sum_{i \in V} x_i$$

$$\sum_{ij \in E} (1 - x_i - x_j + x_i x_j) + \sum_{i \in V} x_i$$

Full derivation: Hamiltonian translation

$$\sum_{ij\in E}(1 - x_i - x_j + x_i x_j) + \sum_{i\in V} x_i$$

Substitute:

$$\sum_{ij\in E}\left(1 - \frac{1}{2}(1 - z_i) - \frac{1}{2}(1 - z_j) + \frac{1}{4}(1 - z_i)(1 - z_j)\right) + \sum_{i\in V}\frac{1}{2}(1 - z_i)$$

Expand:

$$\sum_{ij\in E}\left(1 - \frac{1}{2} + \frac{1}{2}z_i - \frac{1}{2} + \frac{1}{2}z_j + \frac{1}{4} - \frac{1}{4}z_i - \frac{1}{4}z_j + \frac{1}{4}z_i z_j\right) + \sum_{i\in V}\frac{1}{2}(1 - z_i)$$

Collect:

$$\sum_{ij\in E}\left(\frac{1}{4} + \frac{1}{4}z_i + \frac{1}{4}z_j + \frac{1}{4}z_i z_j\right) + \sum_{i\in V}\frac{1}{2}(1 - z_i)$$

$$\sum_{ij \in E} \left(\frac{1}{4} + \frac{1}{4} z_i + \frac{1}{4} z_j + \frac{1}{4} z_i z_j \right) + \sum_{i \in V} \frac{1}{2} (1 - z_i)$$

Consider now that: the total number of edges and vertices are constant - they will provide only an "offset" to the cost, and the values of the variables don't matter.

$$\sum_{ij \in E} \left(\frac{1}{4} z_i + \frac{1}{4} z_j + \frac{1}{4} z_i z_j \right) - \sum_{i \in V} \frac{1}{2} z_i$$

And finally, the absolute value doesn't matter, so we can rescale:

$$\sum_{ij \in E} (z_i + z_j + z_i z_j) - 2 \sum_{i \in V} z_i$$

Can also weight the terms differently depending on which constraint is more important (i.e., if you care more about just getting a valid colouring, weight the first one more).

$$\gamma \sum_{ij \in E} (z_i + z_j + z_i z_j) - 2\lambda \sum_{i \in V} z_i$$

To turn this into a Hamiltonian, recall that

- Each z_i represents an expectation value of Z_i
- Computing expectation values is linear

Full derivation: Hamiltonian translation

$$\gamma \sum_{ij \in E} (z_i + z_j + z_i z_j) - 2\lambda \sum_{i \in V} z_i$$

$$\hat{H} = \gamma \sum_{ij \in E} (Z_i + Z_j + Z_i Z_j) - 2\lambda \sum_{i \in V} Z_i$$

Next time: we will look at the actual QAOA that can find the optimal configuration / minimum energy.