Introduction to variational quantum algorithms[*]

Hanna Linn

October 4, 2023

Contents

For notes, please visit: https://github.com/aqpl-mc2-chalmers/Intro-to-QAOA

1 Quantum state

Quantum computing is just linear algebra in disguise, with some quantum sprinkles on top. This document focuses on the algebra part. For more quantum stuff, see the Appendix.

[*] Mostly QAOA

1.1 Notation: Bra-ket

Quantum physics has its own way of writing linear algebra. The notation used to describe a quantum state is the *bra-ket* notation, also called *Dirac* notation.

- A **vector** describing a quantum state is denoted by the ket-vector: $|\psi\rangle$.
- The **conjugate** to the ket-vector is the bra-vector: $\langle\psi|$.
- The name bra-ket comes from **the scalar product** of the two vectors: $\langle\psi|\phi\rangle$.

1.2 Qubit

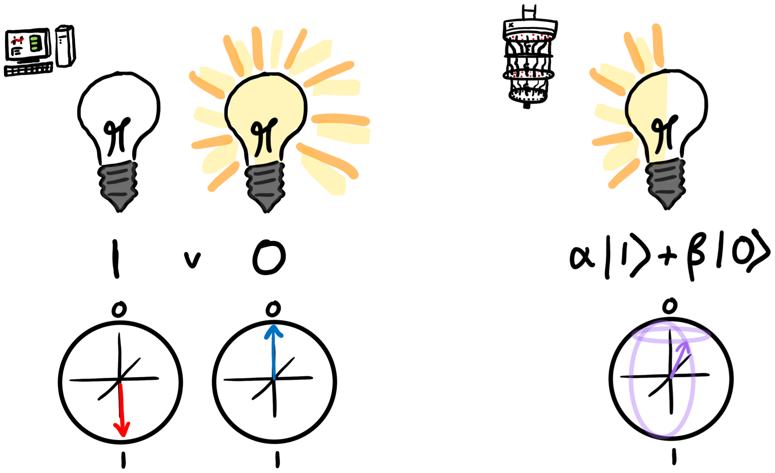


Figure 1: **Left:** The classical bit can either be on, 1, or off, 0. **Right:** The quantum bit, qubit, is a superposition of on, 1, and off, 0. **Bottom of the figure:** the Bloch sphere, a way to visualise the states of the bits.

- A quantum state of a two-level system is called a *qubit*.
- A qubit is the quantum counterpart of a classical bit.

Unlike the classical bit, which can only be in two states, a quantum bit can be in infinitely many states: all superpositions of $|1\rangle$ and $|0\rangle$.

$$|\psi\rangle = \alpha\,|1\rangle + \beta\,|0\rangle = \begin{bmatrix}\alpha\\ \beta\end{bmatrix} \tag{1}$$

where α, β are complex coefficients satisfying $|\alpha|^2 + |\beta|^2 = 1$. The probability of measuring the qubit to be 1 is $|\alpha|^2$ and $|\beta|^2$ for 0.

1.2.1 Bloch sphere

The **Bloch sphere** is a tool to visualise one qubit. A state of the qubit is represented as a point on the surface of the sphere. The poles on the sphere are the states 1 and 0, and all other points on the surface represents superpositions of 1 and 0, described with spherical coordinates. On the equator, in opposite ends of the x-axis, we find the phase states:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{2}$$

1.3 Qubits

If we have more than one qubit, then the total state of that system can be in a superposition of 2^N different states, where N is the number of qubits. For three qubits, the total set of all bit strings:

$$S = \{|x_0\rangle, |x_1\rangle, ..., |x_7\rangle\} = \{|000\rangle, |001\rangle, ..., |111\rangle\} \tag{3}$$

has the size $|S| = 2^3 = 8$.

We can describe the whole registry as:

$$|\Psi\rangle = \sum_i c_i |x_i\rangle, \tag{4}$$

where c_i are complex coefficients satisfying $|c_0|^2 + ... + |c_{N-1}|^2 = 1$, and $|c_i|^2$ is the probability of measuring the particle in state x_i.

The classical registry can of course also be in the same states, but only one at a time. **Storing the information about the amplitudes is the big thing here.** It would take 2^N classical bits to keep track of all the 2^N amplitudes.

The superposition collapses to one state basis vector when a measurement is performed. **To obtain the information contained in the superposition, it is necessary to perform several measurements of the same experiment.**

1.4 Operators

1.4.1 Classical gates

To change our state in a classical computer, we use a gate. For example, the NOT-gate

$$\neg 1 = 0. \tag{5}$$

We can have one-bit gates, as the NOT-gate, or two-bit gates, like the OR-gate and AND-gate.

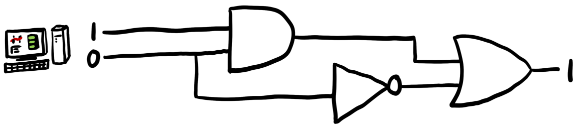


Figure 2: A classical gate circuit: We can use many gates to build a computation, starting in a state and then operate on our registry until we check our answer.

1.4.2 Quantum gates

In a quantum computer, we need quantum gates as the information is quantum. Again, we can have the NOT-gate:

$$\text{NOT}\,|1\rangle = |0\rangle. \tag{6}$$

- Quantum gates needs to be unitary to keep the total probability to 1, i.e., $|c_0|^2 + ... + |c_{N-1}|^2 = 1$. Thereby, general one-qubit gates are denoted U.

1.4.3 Back to the Bloch sphere

To be able to get to all possible quantum states, we have to reach all parts of the Bloch sphere. The different states on the Bloch sphere are reached by rotations around the x-, y- and z-axes. The rotation around axis i with angle θ comes from multiplication with a rotation matrix

$$R_i(\theta) = e^{-i\theta\frac{\sigma_i}{2}}, \tag{7}$$

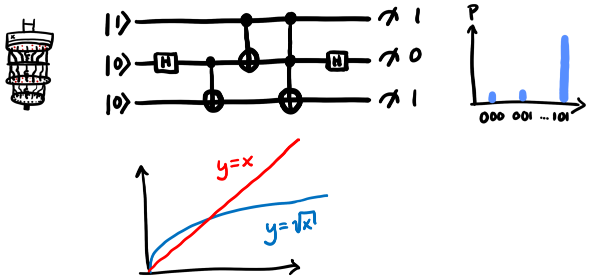


Figure 3: Quantum gate circuit: We can use many quantum gates to build a circuit, starting in a state and then operate on our registry until we measure each qubit in the end. Running the same circuit many times gives us the probability distribution of measuring each bit string. We can get as much as a quadratic speed-up with the quantum circuit compared to the classical circuit.

where σ_i is one of the *Pauli matrices*:

$$\sigma_x = X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \sigma_y = Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \sigma_z = Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{8}$$

which are all Hermitian. We can write the rotation around the x-axis:

$$R_x(\theta) = e^{-i\theta\frac{\sigma_x}{2}} = \cos\left(\frac{\theta}{2}\right) I - i\sin\left(\frac{\theta}{2}\right) X = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}, \tag{9}$$

where I is the identity matrix. Here we have used the power series to define the matrix exponential: $e^A = \sum_{n=0}^{\infty} \frac{A^n}{n!} = I + A + ...$, please see Mathworld.

A rotation of $\theta = \pi$ around the x-axis will be a NOT-gate.

$$R_x(\pi) = \begin{bmatrix} \cos\left(\frac{\pi}{2}\right) & -i\sin\left(\frac{\pi}{2}\right) \\ -i\sin\left(\frac{\pi}{2}\right) & \cos\left(\frac{\pi}{2}\right) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{10}$$

We can now write Equation 6 with matrices instead:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{11}$$

The **Hadamard gate** gets us to the phase states, $|+\rangle$ and $|-\rangle$:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{X+Z}{\sqrt{2}}. \tag{12}$$

The phase shift gate is also important:

$$P(\phi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix}. \tag{13}$$

1.4.4 Multi-qubit gates

Again, we can have two-qubit gates, as the CNOT-gate (controlled NOT), or even three-qubit gates, like the TOFFOLI-gate (controlled controlled NOT). The controlling gate entangles the qubits it acts on, making the acted on qubit dependant on the controlled one(s), see Figure 4.

Given a two-qubit registry spanning $\{|00\rangle, |10\rangle, |01\rangle, |11\rangle\}$, we may want to let the state of the first qubit to control whether a certain single-qubit gate, U, is applied to the second qubit. The block matrix is then:

$$\begin{bmatrix} I_2 & 0_2 \\ 0_2 & U \end{bmatrix}. \tag{14}$$

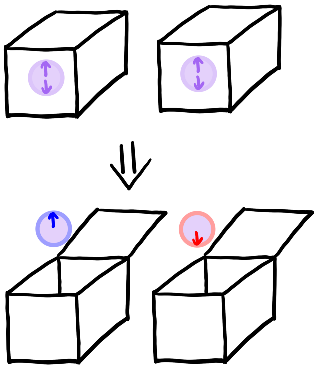


Figure 4: Entangled qubits. Operations on one of the entangled qubits will affect the other. When one is measured, the other chooses a state.

If $U = X$ we get the CNOT-gate:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{15}$$

which is able to do:

$$\text{CNOT}\,|10\rangle = |11\rangle\,. \tag{16}$$

1.4.5 Universal

There are more gates: T-gate, SWAP-gate, CZ, and so on. But you get everywhere with just a few specific gates. You can build all wanted gates from these gates, e.g., the Hadamard gate is just a combination of X and Z and the TOFFOLI-gate can be built with CNOTs and single qubit gates. This calls for compilers.

In classical computing, we have $\{\text{NAND}\}$ as a complete set of operators able to do whatever calculation you may want, called a **Universal set**.

In quantum computing, a universal set have to be able to access the whole Bloch sphere and to entangle bits. A widely used universal set is $\{R_x(\theta), R_y(\theta), R_z(\theta), P(\phi), \text{CNOT}\}$.

2 Quantum approximate optimisation algorithm (QAOA)

2.1 Why?

2.1.1 Future: Large fault-tolerant quantum computers

Shor's and Grover's algorithms showed that it is possible to prove an **exponential and polynomial speedup with respect to their classical counterparts** of prime-factorisation and searching in an unsorted list. The promise of a quantum speedup of classical algorithms sparked the run for building quantum computers. **Shor's and Grover's algorithm calls for fully error-corrected devices within the order of 10^5 qubits** [1], which we have yet to see today.

2.1.2 Today: Small noisy quantum computers:

Until we have large fault-tolerant quantum computers, we have to find algorithms working on the smaller noisy devices we have today, with tens of qubits, so-called Noisy Intermediate-Scale Quantum (NISQ) devices [2, 3, 4, 5], that still is superior to its fastest classical counterpart and could show the next step for helpful quantum computing.

One very popular NISQ algorithm is the **Quantum approximate optimisation algorithm (QAOA)** [6]. QAOA solves specifically combinatorial optimisation problems, and the hope is that it will be able to tackle NP-hard and NP-complete problems. Since its initial publication, the algorithm has had a better approximation ratio than some known polynomial time classical algorithms[**?**] but has been beaten by other classical algorithms since its initial publication. The speed-up of QAOA compared to its classical counterparts is an open research question.

2.2 What?

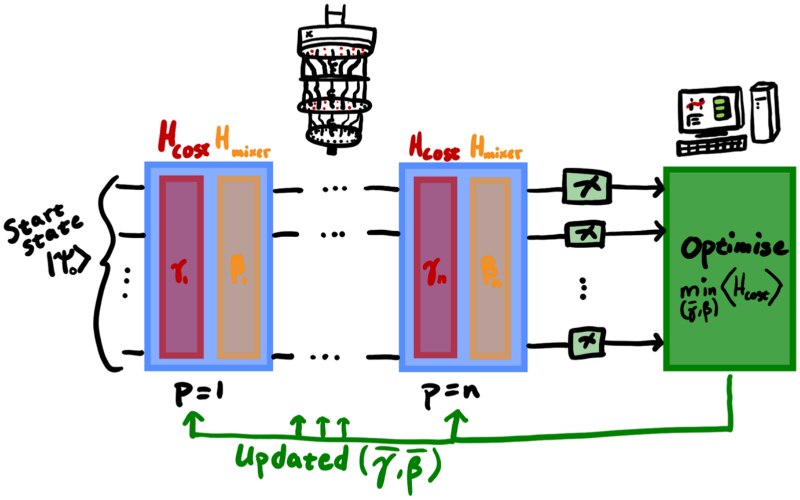
An optimisation problem can be defined as:

$$\begin{array}{ll} \text{minimise} & C_{\text{obj.}}(x) \\ \text{subject to} & x \in F \end{array} \tag{17}$$

where F is the feasible solution set, i.e., all bit strings fulfilling all constraints of the optimisation problem. The energy function of the model, $C_{\text{obj.}}$, can be used for finding the lowest energy solution, x_{best}, $x_{\text{best}} \in F$.

Examples of optimisation problems are MAX-CUT, SAT, EXACT-COVER, and KNAPSACK.

2.3 How?



H_{cost} H_{mixer}
H_{cost} H_{mixer}
Start state $|\psi_0\rangle$
γ_1
β_1
γ_n
β_n
P=1
P=n
Optimise
$\min_{(\vec{\gamma},\vec{\beta})} \langle H_{cost} \rangle$
Updated $(\vec{\gamma},\vec{\beta})$

Figure 5: QAOA-algorithm circuit: We build a circuit in layers to search and constrain each layer. The algorithm consists of a classical optimiser and quantum cirquit. The classical optimiser quarries the quantum circuit in each step and then updates the variables.

With a circuit that is flexible and uses a classical part to accommodate the problem. In QAOA, the circuit is done in a sequence of depth p:

$$\begin{aligned} |\psi_p(\gamma,\beta)\rangle =& e^{-i\beta_p H_{mixer}} e^{-i\gamma_p H_{\text{cost}}} \\ & ...e^{-i\beta_1 H_{mixer}} e^{-i\gamma_1 H_{\text{cost}}} |\psi_{\text{init.}}\rangle , \end{aligned} \tag{18}$$

where γ and β are vectors of parameters to optimise by a classical computer so that the probability of measuring the optimal state is high.

The function to optimise is often the expectation value

$$C_{\text{obj.}} = \langle \psi_p(\gamma,\beta)| H_{cost} |\psi_p(\gamma,\beta)\rangle . \tag{19}$$

If we get the lowest expectation value, we should get the answer with the lowest energy. But we could use another function to optimise over.

2.3.1 Cost Hamiltonian

The cost Hamiltonian encodes the optimisation problem is defined as:

$$H_{\text{cost}} = \sum_i h_i \sigma_i^z + \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z, \tag{20}$$

which represents a number of gates. Encodes the problem (constraints) and thereby guides the initial state of the registry to the final optimal state, together with the mixer Hamiltonian in QAOA.

2.4 Example of encoding: Packing problem

At each position in the amino acid sequence of length N, choose side-chain conformation that gives the lowest energy, given a specific backbone fold, that will stay fixed. Conformations allowed at each position are rotamers of either a given amino acid (no mutation) or a set of rotamers for possible amino acids (mutation allowed). The number of possible rotamers at position n is D_n. The energy for a given choice of rotamers at each position, one instance of the problem, can be calculated with each rotamers own potential (one-body energy O_i) and with the pairwise interaction potentials (two-body interaction T_{kj}). The energies can be contained in a matrix of size the total number of rotamers at all positions squared $(\sum_{i=1}^{N} D_n)^2$.

Is NP-complete and NP-hard as an optimisation problem [7]. Reducible from SAT, also Graph colouring (positions are the nodes and rotamers are the colours). Decision form (yes/no question) of above problem:

Is there a rotamer at each position such that $E_{total} \leq K$ for a specified constant K?

2.4.1 Cost Hamiltonian of rotamer-based model

$$H_{cost}(Z) = \underbrace{\sum_{i=1}^{N} O_i Z_i}_{\text{one-body}} + \underbrace{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} T_{ij} Z_i Z_j}_{\text{two-body}} \tag{21}$$

T and O come from the Rosetta energy function [8].

2.4.2 Transformation into the Ising model

The Ising model uses spin up and spin down for on and off in each qubit, by convention that qubit (binary variable) $z = \{0, 1\}$ translates to spin $s = \{1, -1\}$ we then need the Eq. 21 transformation:

$$z(s) = \frac{1-s}{2}$$

such that

$$z(s=1) = \frac{1-1}{2} = 0$$

$$z(s=-1) = \frac{1+1}{2} = 1.$$

Transformation is then:

$$H_{cost}(S) = \sum_{i=1}^{N} O_i (\frac{1-s_i}{2}) + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} T_{ij} (\frac{1-s_i}{2})(\frac{1-s_j}{2})$$

$$H_{cost}(S) = \sum_{i=1}^{N} (\frac{O_i}{2} - \frac{O_i s_i}{2}) + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} T_{ij} \frac{1}{4}(1 - s_i - s_j + s_{ij})$$

$$H_{cost}(S) = \sum_{i=1}^{N} (\frac{O_i}{2} - \frac{O_i s_i}{2}) + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (\frac{T_{ij}}{4} - \frac{s_i T_{ij}}{4} - \frac{s_j T_{ij}}{4} + \frac{s_{ij} T_{ij}}{4})$$

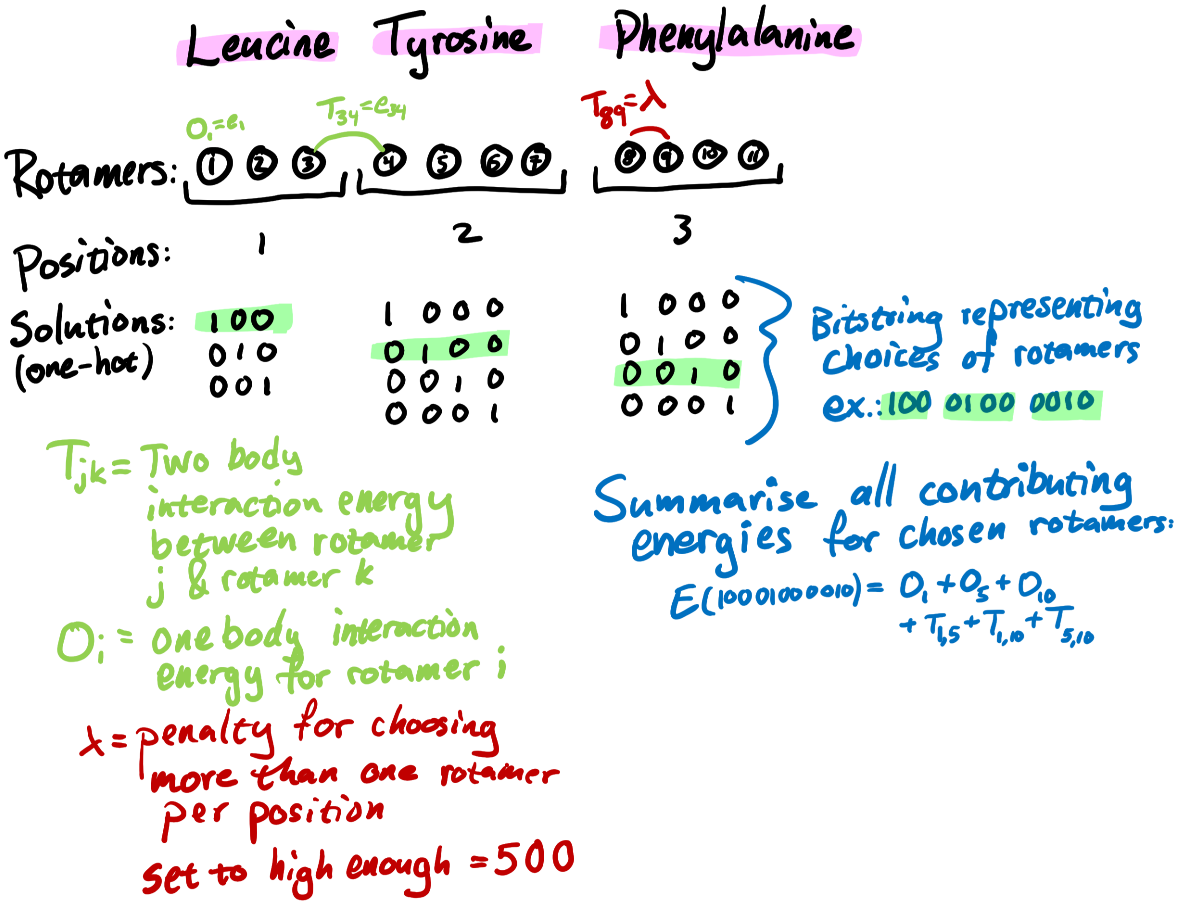

Leucine
Tyrosine
Phenylalanine
$O_1=e_1$
$T_{34}=e_{34}$
$T_{89}=\lambda$
Rotamers: 1 2 3 4 5 6 7 8 9 10 11
Positions: 1 2 3
Solutions: (one-hot)
100
010
001
1000
0100
0010
0001
1000
0100
0010
0001
Bitstring representing choices of rotamers
ex.: 100 0100 0010
Summarise all contributing energies for chosen rotamers:
$E(10001000010) = O_1 + O_5 + O_{10} + T_{1,5} + T_{1,10} + T_{5,10}$
T_{jk} = Two body interaction energy between rotamer j & rotamer k
O_i = One body interaction energy for rotamer i
λ = penalty for choosing more than one rotamer per position
set to high enough = 500

Figure 6: Example of encoding.

$$H_{cost}(S) = \underbrace{\frac{O_i}{2} + \frac{T_{ij}}{4}}_{const.} - \frac{O_i s_i}{2} - \frac{s_i T_{ij}}{4} - \frac{s_j T_{ij}}{4} + \frac{s_{ij} T_{ij}}{4}$$

$$H_{cost}(S) = \sum_{i=1}^{N} -\frac{O_i s_i}{2} + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (-\frac{s_i T_{ij}}{4} - \frac{s_j T_{ij}}{4} + \frac{s_{ij} T_{ij}}{4})$$

2.4.3 Mixer Hamiltonian

X-mixer Naive mixer.

$$H_{X-mixer} = \sum_i X_i \tag{22}$$

XY-mixer Quantum alternate optimisation anzats [9]. A smarter mixer that uses the clusters of the Hamming weight. It will keep the Hamming distance and not walk out of the feasible solution set, given that the algorithm starts in a feasible state. We can put some constraints in the mixer part of the QAOA algorithm, i.e., keep the Hamming distance of clumps of qubits.

$$H_{XY-mixer} = \frac{1}{2} \sum_{i,j \in E(G)} (X_i X_j + Y_i Y_j) \tag{23}$$

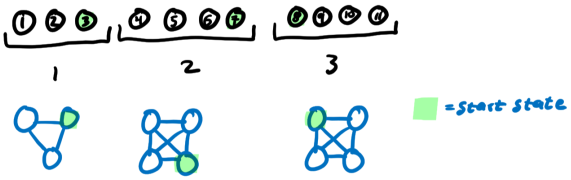


Figure 7: Graphs for positions used in the XY-mixer.

2.5 Encodings

How the problem is encoded affects the use of qubits and gates in the quantum computer.

Decimal	OH	BIN	Gray	BUBIN$_{g=3}$
0	10000	000	000	00 01
1	01000	001	001	00 10
2	00100	010	011	00 11
3	00010	011	010	01 00
4	00001	111	110	10 00

Table 1: Different encodings.

2.6 How did we do?

Metric 1: Ratio of approximation (RA):

$$r(\gamma, \beta) = \frac{\langle \gamma, \beta | \, H_{cost} \, | \gamma, \beta \rangle}{C_{min}} \tag{24}$$

where the numerator is just the cost function.

Metric 2: Success probability or Overlap with solution:

$$SP(\gamma, \beta) = | \, \langle x_{best} | \, | \gamma, \beta \rangle \, |^2 \tag{25}$$

which is the probability of measuring the solution given the parameters.

3 Other quantum algorithms

3.1 Quantum annealing

Quantum annealing (that runs on the D-wave) is the inspiration for the QAOA algorithm, but they do not work precisely the same. QAOA can be mapped into quantum annealing, but QAOA is an algorithm to be run on a gate-based quantum computer. See Tutorial 3 from the Advanced quantum algorithms for comparison.

You don't need to understand quantum annealing in order to understand QAOA, but it is good to know that there are many similarities between them.

4 Variational quantum eigensolver

Given a molecule, you may want to know the orbitals of the electrons. The variational quantum eigensolver (VQE) is also a NISQ algorithm and if quantum chemistry is your field, then this is the jam. Similar to QAOA, it's a hybrid algorithm with a quantum circuit and a classical optimiser. As of 2022, the variational quantum eigensolver can only simulate small molecules like the helium hydride ion or the beryllium hydride molecule. In 2020, a 12-qubit hydrogen chain (H12) simulation was demonstrated using Google's Sycamore quantum processor.

A Appendix: More quantum physics

A.1 Hilbert spaces

The whole state space is the Hilbert space

$$\mathcal{H} = \{\left|\psi_0\right\rangle, ..., \left|\psi_n\right\rangle\}, \tag{26}$$

where the $\left|\psi_i\right\rangle$ are the basis vectors of the state space. It is possible to describe any property of a quantum system, such as the energy level of an electron or its distance from the atom kernel. An *observable* a, a physical quantity, e.g., the position of the electron, is represented by the operator $\hat{A}$ acting on the Hilbert space. Measurement of a yields one of the operator's eigenvalues $\hat{A}$.

The **tensor product** $\otimes$ between two Hilbert spaces provides a way to describe a space with more than one property or degree of freedom, i.e., a system consisting of subsystems. For example, in the Hilbert space

$$\mathcal{H} = \mathcal{H}_{spin} \otimes \mathcal{H}_{space} \tag{27}$$

the tensor product separates two degrees of freedom, spin and space, each with their own basis vectors. Using tensor notation makes it possible to view resulting correlations between the two degrees of freedom.

Together with the identity matrix, the Pauli matrices constitute a unitary operator basis, called the *Pauli group* $\boldsymbol{P}_n$, to the Hilbert space spanning the qubit states $\mathcal{H}_{qubit} = \{\left|0\right\rangle, \left|1\right\rangle\}$.

A.2 Superposition

Two or more distinct states can be added together to form a *superposition state*. An example of a superposition is the rolling motion of a wheel. The motion can be expressed as a translational motion together with a rotational motion. A *pure* superposition state, a state that can be described by a single ket vector, is described as

$$\left|\Psi\right\rangle = c_0\left|\psi_0\right\rangle + c_1\left|\psi_1\right\rangle + ... + c_n\left|\psi_n\right\rangle, \quad c_{1,...,n} \in \mathbb{C}, \tag{28}$$

where $c_{1,...,n}$ are complex coefficients satisfying $|c_0|^2 + ... + |c_n|^2 = 1$, and $|c_i|^2$ is the probability of measuring the particle in state ψ_i. If $\left|\Psi\right\rangle$ is expanded in the eigenbasis of $\hat{A}$, operator for observable a, then $|c_i|^2$ is the probability of measuring a_i, e.g., a certain position. The superposition collapses to one state basis vector when a measurement is performed. To obtain the information contained in the superposition, it is necessary to perform several measurements of the same experiment.

A.3 Schrödinger equation

The *Schrödinger equation* describes a quantum system evolving with time

$$i\hbar \frac{d}{dt}|\Psi(t)\rangle = \hat{H}|\Psi(t)\rangle, \tag{29}$$

where $\hbar$ is the reduced Planck constant. $\hat{H}$ is the Hamiltonian operator, and it corresponds to the total energy of the system, both kinetic and potential energy. The Hamiltonian must be *Hermitian*, i.e., it has real eigenvalues and fulfils,

$$\hat{H} = \hat{H}^{\dagger}, \tag{30}$$

where $\hat{H}^{\dagger}$ is called the *Hermitian adjoint*. The Schrödinger equation predicts that when certain properties of a system are measured, the results may be quantised, e.g., the energy of an electron in an atom is discrete.

The *position-space wave function* $\Psi(\mathbf{r}, t)$ of the quantum system is the expansion of the state vector in terms of the position eigenvector $|\mathbf{r}\rangle$, such that

$$\Psi(\mathbf{r}, t) = \langle \mathbf{r}|\Psi(t)\rangle \, . \tag{31}$$

The wave function provides information about the state of the particle. According to Born's statistical interpretation, when measuring position, the probability of finding the particle at point $\mathbf{r}$ at time t is given by integrating $|\Psi(\mathbf{r}, t)|^2$ over space. To get the wave function, one has to solve the Schrödinger equation expressed in the position basis

$$i\hbar \frac{\partial}{\partial t}\Psi(\mathbf{r}, t) = \left[\frac{-\hbar^2}{2m}\nabla^2 + V(\mathbf{r}, t)\right]\Psi(\mathbf{r}, t), \tag{32}$$

where $V(\mathbf{r}, t)$ is the potential energy of the particle at position $\mathbf{r}$ at time t, and the terms within the square brackets describe the Hamiltonian. This version of the Schrödinger equation is analogous to Newton's second law.

The wave nature of particles, e.g., electrons, encoded into the wave function, give rise to the interesting phenomenon of *interference*. Interference of two waves in phase is when the overlapping of the waves either reinforces each other in some locations, called constructive interference, or cancels each other out in other locations, called destructive interference. An interference pattern for the particle's location appears at the distance of measurement, shown to be true for electrons by Young's famous double-slit experiment in 1801.

References

[1] N. C. Jones, R. Van Meter, A. G. Fowler, P. L. McMahon, J. Kim, T. D. Ladd, and Y. Yamamoto, "Layered Architecture for Quantum Computing," *Physical Review X*, vol. 2, p. 031007, July 2012. Publisher: American Physical Society.

[2] P. Chandarana, N. N. Hegade, I. Montalban, E. Solano, and X. Chen, "Digitized-Counterdiabatic Quantum Algorithm for Protein Folding," Dec. 2022. arXiv:2212.13511 [cond-mat, physics:quant-ph].

[3] L. DiCarlo, J. M. Chow, J. M. Gambetta, L. S. Bishop, B. R. Johnson, D. I. Schuster, J. Majer, A. Blais, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf, "Demonstration of two-qubit algorithms with a superconducting quantum processor," *Nature*, vol. 460, pp. 240–244, July 2009.

[4] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, "Demonstration of a small programmable quantum computer with atomic qubits," *Nature*, vol. 536, pp. 63–66, Aug. 2016.

[5] M. Reagor, C. B. Osborn, N. Tezak, A. Staley, G. Prawiroatmodjo, M. Scheer, N. Alidoust, E. A. Sete, N. Didier, M. P. da Silva, E. Acala, J. Angeles, A. Bestwick, M. Block, B. Bloom, A. Bradley, C. Bui, S. Caldwell, L. Capelluto, R. Chilcott, J. Cordova, G. Crossman, M. Curtis, S. Deshpande, T. El Bouayadi, D. Girshovich, S. Hong, A. Hudson, P. Karalekas, K. Kuang, M. Lenihan,

R. Manenti, T. Manning, J. Marshall, Y. Mohan, W. O'Brien, J. Otterbach, A. Papageorge, J.-P. Paquette, M. Pelstring, A. Polloreno, V. Rawat, C. A. Ryan, R. Renzas, N. Rubin, D. Russel, M. Rust, D. Scarabelli, M. Selvanayagam, R. Sinclair, R. Smith, M. Suska, T.-W. To, M. Vahidpour, N. Vodrahalli, T. Whyland, K. Yadav, W. Zeng, and C. T. Rigetti, "Demonstration of universal parametric entangling gates on a multi-qubit lattice," *Science Advances*, vol. 4, p. eaao3603, Feb. 2018. Publisher: American Association for the Advancement of Science.

[6] E. Farhi, J. Goldstone, and S. Gutmann, "A Quantum Approximate Optimization Algorithm," Nov. 2014. arXiv:1411.4028 [quant-ph].

[7] N. A. Pierce and E. Winfree, "Protein Design is NP-hard," *Protein Engineering, Design and Selection*, vol. 15, pp. 779–782, Oct. 2002.

[8] R. F. Alford, A. Leaver-Fay, J. R. Jeliazkov, M. J. O'Meara, F. P. DiMaio, H. Park, M. V. Shapovalov, P. D. Renfrew, V. K. Mulligan, K. Kappel, J. W. Labonte, M. S. Pacella, R. Bonneau, P. Bradley, R. L. Dunbrack, R. Das, D. Baker, B. Kuhlman, T. Kortemme, and J. J. Gray, "The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design," *Journal of Chemical Theory and Computation*, vol. 13, pp. 3031–3048, June 2017.

[9] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, "From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz," *Algorithms*, vol. 12, p. 34, Feb. 2019. arXiv:1709.03489 [quant-ph].