Quantum Approximate Optimisation Algorithm

Trace QAOA's cost phases and mixer through a completely solvable one-edge example.

Undergraduate | 22 min lesson | Dr. Charith Jayasekara

By the end, you'll be able to

- State the cost and mixer sign conventions.
- Calculate one-layer amplitudes for a single edge.
- Distinguish expected score from optimum probability.

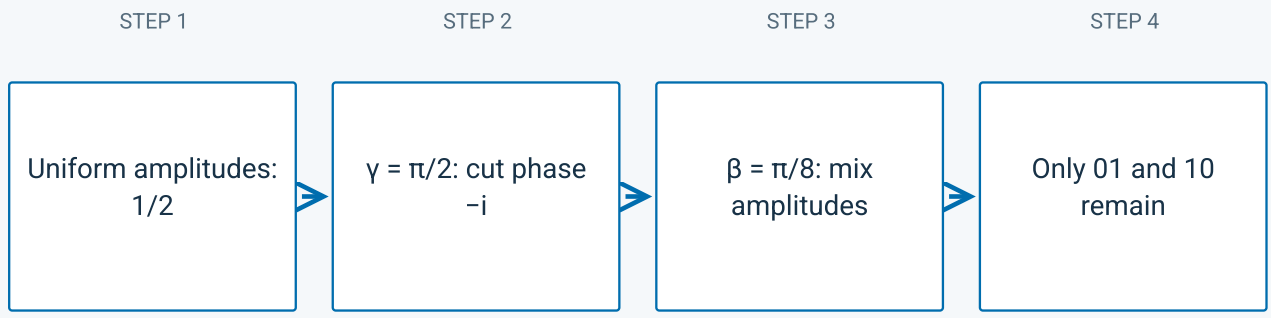
Alternate a score-dependent phase with mixing

QAOA constructs a parameterised circuit whose measurement outcomes are candidate solutions to a discrete optimisation problem. A cost operation gives different phases to candidates with different objective values. A mixer then combines amplitudes. Repeating these operations can make the final sampling distribution depend usefully on the objective. A classical optimiser chooses the angles from measured scores.

Prerequisites: phase and interference, controlled gates and a binary optimisation model. The original algorithm supplies specific approximation analyses for selected graph families; it does not prove practical superiority for every QUBO. Farhi, Goldstone and Gutmann.

WORKED DIAGRAM · A18

One-edge QAOA



At the stated angles, uncut amplitudes cancel and cut amplitudes become $-i/\sqrt{2}$, giving cut score 1 with certainty.

The algorithm setup

We use a maximisation objective C diagonal in the computational basis, a uniform initial state and a mixer B equal to the sum of X on each qubit. One layer applies the cost evolution with angle gamma, then the mixer evolution with angle

beta. With p layers there are two p angles. Measurement returns a bitstring, whose score is evaluated classically.

The classical objective can be the expected score or a specified alternative such as a tail statistic. State the choice. Optimising expectation and maximising the probability of an exact optimum are related but different goals. The training budget, shot allocation and final sampling budget are resources of the algorithm, not incidental interface details.

Worked example: a single edge

Two vertices connected by one edge have cut score zero for 00 and 11, and one for 01 and 10. The cost operator is one half times the identity minus the Z product. Start with equal amplitudes one half on all four strings. A cost angle of pi over two gives phases one on the uncut strings and negative i on the cut strings.

Choose beta equal to pi over eight. Each wire's mixer is the identity times cosine beta minus i X times sine beta. Combining both wires, the final amplitudes on 00 and 11 cancel to zero, while those on 01 and 10 become negative i divided by the square root of two. The cut score is therefore one with certainty in this tiny ideal example.

More generally, for this single edge and the stated signs, the expected cut is

$$\langle C\rangle = \frac{1+\sin(4\beta)\sin\gamma}{2}.$$

At beta zero, the distribution stays uniform because the cost phases alone do not change probabilities. At gamma zero, the uniform initial state is an eigenstate of the mixer, again leaving the distribution uniform. Both transformations are needed for the interference effect. These limits are useful tests of an implementation.

The cost phase and mixer have distinct jobs, captured by the one-layer state.

$$C=\frac{I-Z_0Z_1}{2},\qquad B=X_0+X_1,\qquad |\gamma,\beta\rangle = e^{-i\beta B}e^{-i\gamma C}|++\rangle.$$

C awards one for different endpoint bits, while B mixes the two binary choices. The rightmost cost operation acts first; both angles are in radians.

For the stated angles, the uncut amplitudes cancel exactly.

$$\gamma=\frac{\pi}{2},\ \beta=\frac{\pi}{8}:\qquad |\gamma,\beta\rangle=-i\frac{|01\rangle+|10\rangle}{\sqrt{2}},\qquad \langle C\rangle=1.$$

The global factor $-i$ does not change the two equal cut probabilities. This exact tiny reference tests the signs and execution order in a QAOA implementation.

Step-by-step workflow

First build and test the classical objective. Map it to a diagonal cost operator, including constraint penalties if needed. Prepare the uniform state with H on every qubit. Apply alternating cost and mixer operations for chosen angles. Measure many fresh preparations, evaluate their scores and estimate the chosen training objective. Let a classical search method propose new angles, then repeat under a recorded budget.

After training, freeze the selected angles and collect a separate final sample. Decode and verify each candidate using the original problem constraints. Report the best score, its frequency, uncertainty and total cost. Separating training data from final evaluation prevents an unusually lucky optimisation sample from becoming an inflated performance claim.

Why deeper is not automatically better in practice

The ansatz family becomes more expressive as layers are added, but an optimiser may fail to find good parameters. Hardware errors, shot noise and limited budgets can make deeper circuits perform worse in an actual workflow. Statements about the best achievable value at a depth are not guarantees about a particular noisy training run.

Constraints also influence mixer design. A simple transverse-X mixer explores every bitstring, including infeasible ones. A feasibility-preserving mixer can stay inside a constraint subspace, but constructing it may require additional problem knowledge and gates. Penalty and constrained-mixer approaches solve related tasks with different assumptions and costs.

Cost terms that commute can often be implemented conveniently as phase rotations, but hardware routing may still add depth. Dense pair interactions require a connectivity strategy. Angle precision, compilation and control errors can matter when comparing an abstract p-layer circuit with a physical device. Count those costs rather than presenting the layer number as a complete runtime measure.

Expected score is not always optimum probability

On the single-edge problem, score is either zero or one, so expected score equals the probability of an optimal cut. That special agreement need not survive a larger objective with several score levels. Imagine one sampler always returns score one, while another returns score zero half the time and score two half the time. Both expectations are one, but only the second ever returns a score-two optimum.

If an application needs a best candidate, report the final score distribution or success probability at a declared target. If it needs average value, expectation may be the appropriate objective. Selecting training angles using one measure and reporting another is acceptable only if the relationship is explained and independently evaluated.

For the single edge, a success probability of 0.9 would give an expected 900 successful cuts in one thousand shots, with ordinary binomial fluctuation around that count. The circuit’s ideal distribution and the finite observed best score remain different levels of evidence.

What to compare classically

For one edge, exhaustive evaluation is immediate and uniform sampling already succeeds half the time. The example teaches mechanism, not advantage. On larger graphs, compare against relevant approximation algorithms, local search and exact methods where feasible. Match target quality and include the many circuit evaluations used to tune parameters. A single successful sample omits most of the work.

QAOA is a family of algorithmic designs, including variants of initial states, mixers and optimisation rules. Keep the chosen definition visible. A theorem for a particular graph family or depth does not apply automatically to every application labelled QAOA.

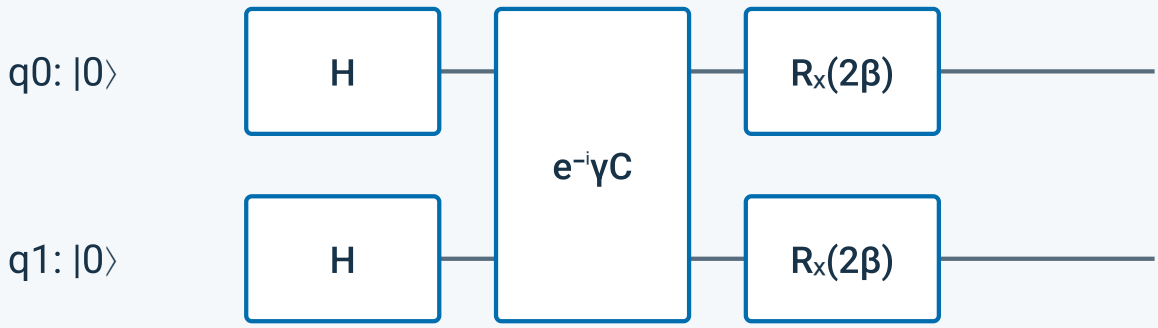
Explore the mechanism

Use Explore freely to compare amplitudes and measurement probabilities. Choose Predict & test for a quick prediction, then run measurements. Change one setting at a time and explain the change.

ALGORITHM LAB Ideal local model

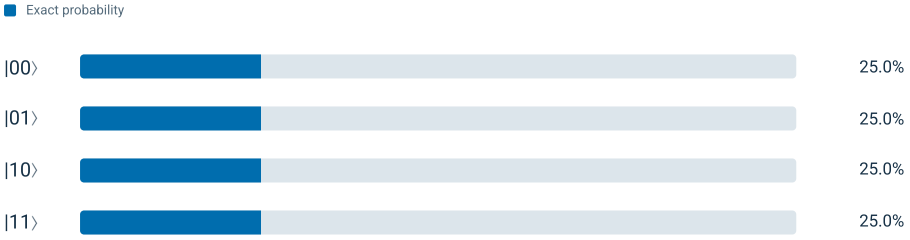
QAOA: phases and mixing on one edge

Two vertices and one edge. Cost phases use exp(−iγC); the mixer uses exp(−iβ(X_0+X_1)). Angles below are in degrees.



The statevector is an ideal simulation. A useful application also counts state preparation, oracle construction, compilation and measurement.

Initial model reference



Statevector before readout · ideal simulation

Basis	Amplitude a + bi	Magnitude	Phase (rad)	Probability
|00⟩	`0.5 + 0i`	0.500000	0.000000	25.0%
|01⟩	`0.5 + 0i`	0.500000	0.000000	25.0%
|10⟩	`0.5 + 0i`	0.500000	0.000000	25.0%
|11⟩	`0.5 + 0i`	0.500000	0.000000	25.0%

Signs and complex phases are part of the state. Phase is undefined for a zero amplitude. A shared global phase changes no probabilities; relative phases can affect a later operation. Values are rounded to six decimal places.

Exercises and worked answers

Exercise 1. For the single edge, set gamma to pi over two and beta to zero. What is the success probability, and why did the oracle-like phase not improve it?

Check the worked answer

It is one half. The magnitudes remain one half on every basis state. Without mixing, the phase pattern has not been converted into a population difference.

Exercise 2. Use the expectation formula at beta equal to pi over eight and gamma equal to negative pi over two. What happens?

Check the worked answer

The expected cut is zero, so only uncut outcomes occur. The sign convention changes constructive and destructive interference. A cost-sign mistake can drive the circuit toward the opposite objective even when the implementation is otherwise correct.

Check your understanding

PAUSE & THINK

In the one-edge example, what happens if β = 0?

- [ ] The cut succeeds with certainty
- [ ] The cut probability stays 1/2
- [ ] Every amplitude becomes zero

Answer: The cut probability stays 1/2. A cost phase alone changes no basis probabilities. Mixing is needed to convert the phase pattern into a population difference.

Sources

- Farhi, Goldstone and Gutmann