Quantum Annealing

Explain ideal adiabatic following and the additional assumptions of physical annealing.

Undergraduate 21 min lesson Dr. Charith Jayasekara

By the end, you'll be able to

- Calculate a single-spin spectrum and minimum gap.
- Distinguish a schedule from its hardware implementation.
- Include repetition and embedding in a performance study.

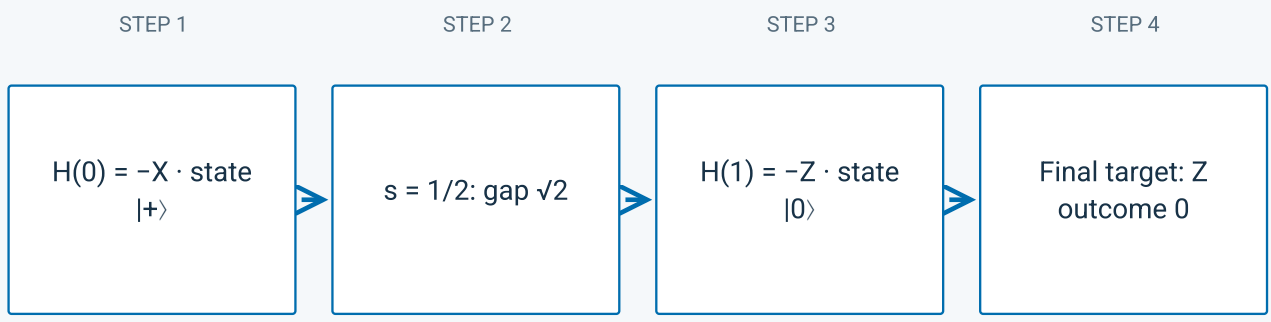
Change the energy landscape gradually

Quantum annealing uses a time-dependent physical Hamiltonian to seek low-energy configurations of a problem model. An ideal adiabatic algorithm starts in an easily prepared ground state and changes the Hamiltonian slowly enough to remain near the desired evolving eigenstate. Practical annealing devices operate with finite temperature, noise, control limits and restricted interactions, so their behaviour need not match that ideal closed-system picture.

Prerequisites: qubits, noise and hardware limits and the QUBO-to-Ising conversion. The foundational adiabatic proposal explicitly links runtime requirements to the minimum spectral gap rather than promising a fast solution to every optimisation instance. Farhi and colleagues.

WORKED DIAGRAM · A20

A solvable annealing path



For H(s) = −(1−s)X − sZ, the endpoint gaps are 2 and the minimum gap is √2 at s = 1/2.

The problem setup

The input is a supported energy model, an embedding into hardware interactions, an initialisation and an annealing schedule. The output is sampled classical configurations after readout. The application decodes those configurations and

evaluates their original objective. A low-energy sample is not necessarily a proven optimum, and an optimiser's success probability must be estimated over repeated runs.

A common interpolation combines a transverse-field driver with a diagonal Ising problem Hamiltonian. The driver has an easy product ground state. Early in the schedule its contribution dominates; later the problem contribution dominates. The physical schedule may include pauses, nonlinear ramps or reverse-annealing variants, but each choice changes the algorithm and should be recorded.

Worked example: a single spin

Consider the interpolation from negative X to negative Z:

$$H(s) = -(1-s)X - sZ, \qquad 0 \le s \le 1.$$

At the start, the ground state is plus. At the end, it is zero. The two energy levels are positive and negative the square root of the sum of the two squared coefficients. The gap is twice that square root. Its minimum occurs at the halfway point and equals square root of two, so this tiny path has no vanishing gap.

The ground-state Bloch vector points in the direction with x component proportional to one minus s and z component proportional to s. At the midpoint it lies halfway between positive x and positive z. A sufficiently controlled slow evolution can carry the state along that changing direction. An abrupt final change would instead leave the initial plus state, whose Z measurement is only half likely to give the desired zero.

This is a mechanism example, not a speedup demonstration. One spin can be solved by inspection. Its value is that the initial state, target, spectrum and gap can all be calculated, making the role of a schedule concrete before discussing a large interacting problem.

The two energy levels give a nonzero gap throughout this simple interpolation.

$$E_{\pm}(s) = \pm\sqrt{(1-s)^2 + s^2}, \qquad \Delta(s) = E_{+}(s) - E_{-}(s) = 2\sqrt{(1-s)^2 + s^2}, \qquad \min_{s} \Delta(s) = \Delta(1/2) = \sqrt{2}.$$

s runs from zero to one along the existing path $H(s) = -(1-s)X - sZ$. The gap is an energy difference; the schedule determines how quickly the path is traversed.

Why the gap matters

Nearby energy levels are harder to separate dynamically. Adiabatic error bounds depend on the spectral gap, the rate and smoothness of change, and relevant matrix elements; a bare "slow enough" instruction is incomplete. Hard instances may have very small avoided-crossing gaps. Increasing the system size does not guarantee that a practical schedule remains efficient.

Tunnelling is often discussed as a possible advantage of quantum dynamics, but it is not a universal escape route through every optimisation barrier. Barrier width, structure, temperature, dissipation and alternative classical moves all matter. An analogy with a ball passing through a hill cannot replace an instance-dependent runtime analysis.

From a model to hardware

The problem graph may require interactions unavailable directly on the device. An embedding can represent one logical variable with a chain of physical qubits. Chain-strength selection, broken-chain decoding and coefficient scaling become part of the workflow. Count the physical resources used after embedding, not just the original number of binary variables.

Finite precision can also change the realised energy ordering. If two candidate solutions differ by a tiny objective value after rescaling, calibration errors may compete with that difference. Temperature and environment can induce transitions

during the schedule. A useful experiment characterises these effects instead of interpreting every failure as an abstract algorithmic limitation or every success as proof of adiabatic following.

Design a complete sampling study

Choose a family of instances and a target quality. Estimate the probability of success per run, including uncertainty. If repeated runs are approximately independent and each succeeds with probability p, the probability of at least one success after R runs is one minus one minus p raised to R. Select R to meet a specified confidence target, then include programming, annealing, readout and postprocessing costs consistently.

Compare against relevant classical optimisation methods, including strong heuristics rather than only random guessing. Separate an empirical speed difference on finite instances from an asymptotic advantage. A primary benchmark-methodology paper explains why the chosen comparison and scaling question can change the meaning of a speedup claim. Rønnow and colleagues.

A changing schedule is part of the experiment

At the midpoint of the single-spin path, the ground-state direction has equal positive x and z components. Its Z-zero probability is one half times one plus one over the square root of two, approximately 0.853553. This is neither the initial half probability nor the final certainty. The intermediate state can therefore be checked through a calculated observable, not only through an end-of-run success flag.

A pause at that point is a different schedule from a uniform ramp. In an ideal isolated system already in its instantaneous eigenstate, holding the Hamiltonian only adds an eigenphase. In an open physical system, a pause can permit relaxation or other environmental processes. The same visual schedule change therefore has different interpretations under different models.

When comparing schedules, keep total duration, temperature assumptions and readout procedure explicit. A better final sample rate may reflect dynamics, noise or selection of a different runtime budget. A controlled study changes one factor at a time before assigning a mechanism.

Exercises and worked answers

Exercise 1. Calculate the single-spin gap at s equal to zero, one half and one.

> Check the worked answer
>
> The endpoint gaps are two. At the midpoint the coefficient norm is one over square root of two, so the gap is square root of two. This is the minimum along this path.

Exercise 2. A device succeeds with probability 0.2 per independent run. What is the chance of at least one success in ten runs?

> Check the worked answer
>
> It is one minus 0.8 to the tenth power, approximately 0.8926. Reporting only a successful best sample would hide that repetition cost. Correlations between runs would require revisiting the independent-run model.

Check your understanding

PAUSE & THINK

What is the minimum energy gap along this single-spin path?

- ☐ 0
- ☐ 1
- ☐ √2

Answer: √2. The gap is $2\sqrt{((1-s)^2+s^2)}$, minimised at $s = 1/2$, giving √2.

Sources

- Farhi and colleagues
- Rønnow and colleagues