Overview

Quantum Annealer: Special-purpose quantum computer;

- It is different from General Purpose Quantum Computers which can be freely programmed from computers
- Instead, it is designed & built to host a single quantum algorithm, **Quantum Annealing**

Quantum Annealing: heuristic optimization technique used to find the global minimum of a problem's energy landscape. It uses quantum mechanics, particularly quantum tunneling and superposition, to explore solutions.

- Simply put, quantum algorithm capable of solving optimization problems
- It is particularly useful for solving combinatorial optimization problems and challenges in discrete mathematics, such as factoring large numbers, solving SAT problems, or optimizing logistics.

History of Quantum Annealing

- **Simulated Annealing (1983)**
  - Classical simulated annealing was introduced by Scott Kirkpatrick, C. Daniel Gelatt, and Mario P. Vecchi.
  - technique involves gradually lowering the "temperature" of a system to find a global minimum in an optimization problem
  - inspired by *metallurgical annealing processes*, i.e. thermodynamics process
- **1998 (A. B. Finnila et al.):** Quantum annealing as a concept was introduced; proposes quantum fluctuations instead of thermal fluctuations
- **2001 (T. Kadowaki and H. Nishimori)** presented the formal theoretical work; explains how adiabatic quantum computing connects!
  - Link to Research Paper: https://arxiv.org/abs/cond-mat/9804280
- **D-Wave Systems (2000s):** The practical realization of quantum annealing began with D-Wave Systems, a Canadian company founded in 1999.
  - Introduced the world's first commercial quantum computer, **D-Wave One**

Fun fact:
The D-Wave Advantage is the largest quantum annealer in the world, with a quantum processing unit (QPU) that has over **5,000 qubits**. It's considered the **most powerful and connected quantum computer** available, with more than double the qubits and 2.5 times the connections of its predecessor, the D-Wave 2000Q.

Simulated Annealing, Classical Algorithm

Suppose we have an optimization problem; for example, a minimization problem where there is a function and we want to find the *lowest point* or *global minimum*.

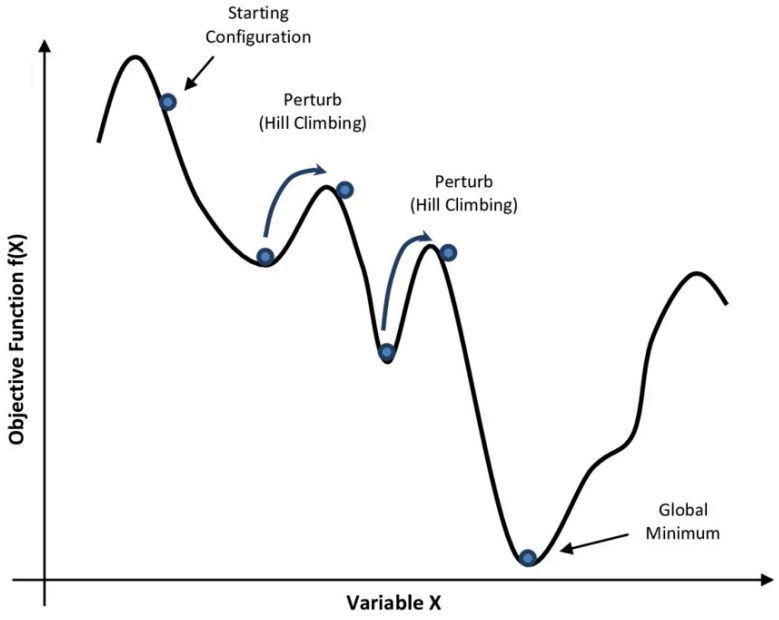
- *Objective function* - the function to be minimized; it is known and computable using a finite set of variables

Best way to solve a minimization problem is the **brute-force approach**

- calculate all the values for all possible inputs and consider the smallest
- this isn't a great approach if you have a lot of inputs (for N inputs, that is 2^n combinations)

Another way to solve minimization problems are **Simulated Annealing**: probabilistic strategy used to solve optimization problems

- Think of this analogy: a ball rolls along the graph of a function, falling into the holes defined by the minima
- Every time a ball lands in a hole it receives a certain amount of energy, enough to make it jump over another piece of the graph, looking for deeper minima points.



Article: https://medium.com/analytics-vidhya/simulated-annealing-869e171e763c

This is a really nice algorithm if:

- guarantees a convergence upon running a sufficiently large number of iterations
- usually better than greedy algorithms when it comes to problems that have numerous locally optimum solution

However:

- For problems where the energy landscape is smooth, or there are few local minima, SA is overkill.
- Repeatedly annealing with a 1/log(k) schedule is very slow, especially if the cost function is expensive to compute.

Quantum Annealing

Key Steps in QA:

1. Representation of Problem
    - We previously introduced an objective function - this is really just a mathematical expression of the energy of a system
        - We want to find the minimum of this function; i.e. lowest energy of a system;
        - If the solver is a QPU, this is a function of the binary variables that represent its qubits; If the solver is a classical quantum hybrid solver, energy might be an abstract function
    - So, first, problems are mapped to a Hamiltonian (a mathematical object representing the total energy of a quantum system).
    - Before that, the goal is to convert all our problems into **minimization problems**.
        - Example: x + 1 = 2 can be rewritten as: $min[2-(x+1)]^2$
2. Problem Mapping to QUBO/Ising Model:
    - Quantum annealing solves problems expressed in either of these two forms: **QUBO or Ising model**
    - These are the native formats for quantum annealers; Why? It's because fundamental mechanism of quantum annealing relies on **minimizing the energy of the system**
    - Both models actually describe optimization problems as energy functions or Hamiltonians
        - In the **Ising model**, the ground state focuses on minimizing the spins' interaction energy.
        - In **QUBO**, the ground state corresponds to the binary configuration that **minimizes the quadratic objective**. Remember, objective is just another word for the function we want to minimize.
    - Note: Any QUBO problem can be easily mapped into an ISING problem through simple equivalence
    - Note: *PyQUBO is a python library, with a C ++ backend, written by DWAVE to use its quantum annealer.*
3. **Hamiltonian Dynamics**
    - At this point, assume we have the problem stated in terms of Hamiltonian (through Ising or QUBO)

- This is not where we start; The system begins in a simple initial Hamiltonian $H_{initial}$ where the ground state is easily achievable
    - This is designed to represent a state where all qubits (quantum bits) are in a uniform superposition: $H_{\text{initial}} = -\sum_i \sigma_x^i$
        - Pauli-X operator flips the state of a qubit; it has maximum uncertainty so we don't know what this final state will look like
    - The ground state will look like this: $[|\psi_{\text{initial}}\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle]$

- Over time, we want it to evolve into the problem Hamiltonian which is our Ising model or QUBO model;
- We want the problem Hamiltonian to be in this format: $[H_{\text{problem}} = \sum_i h_i \hat{s}_i + \sum_{i<j} J_{ij} \hat{s}_i \hat{s}_j]$
    - Note: s_i represents the quantum operator representing the i^{th} spin variable
    - H_{ij} is a coefficient representing the external field acting on the spins
    - J_{ij} is coefficient representing the coupling strength
- This is very similar to the Ising **model**; if we are using QUBO, we need to do some extra steps to get this form. See below.
- The system evolves smoothly from $H_{initial}$ and $H_{problem}$ governed by: $H(t) = (1-s(t))H_{initial} + s(t)H_{problem}$
- Note: s(t) is a time-dependent parameter that gradually increases from 0 to 1
- As we evolve, we rely on **adiabatic theorem** of quantum mechanics:
    - If the Hamiltonian changes slowly enough and there is a sufficient energy gap between the ground state and excited states, the quantum system remains in its ground state throughout the evolution.
- This means starting from the ground state of $H_{initial}$, the system will evolve to the ground state of $H_{problem}$
- The ground state of $H_{problem}$ happens to be the solution to our optimization problem
    - Remember, the quantum state of the system evolves according to the **time-dependent Schrödinger equation**: $i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle = H(t)|\psi(t)\rangle$
- At every time t, the Hamiltonian has a *unique ground state* and system's quantum state will follow this if evolution is slow
- Some quantum properties that are really helpful for this:
    - quantum entanglement captures correlations between variables, helping the system find the global minimum.
    - initial state is a superposition of all possible configurations; as we evolve, it collapses into the state corresponding to ground state
    - quantum tunneling allows the system to move through energy barriers that would trap classical systems in local minima.

Example 1

https://github.com/dwave-examples/factoring-notebook

Problem Statement: Factor the integer N=15 into its prime components.

- If N = pq where p and q are prime numbers then factor N into finding integers p and q
- We express this problem as an optimization problem by converting it into a QUBO model.

Let p and q be represented as binary variables. Assume it's n = 2 bits long and let $p = p_0 + 2p_1$ and $q = q_0 + 2q_1$where $p_0, q_0, p_1, q_1 \in 0, 1$

So, if pq = N, then using the above format:

$$(p_0 + 2p_1) \cdot (q_0 + 2q_1) = 15$$
$$p_0q_0 + 2p_0q_1 + 2p_1q_0 + 4p_1q_1 = 15$$
$$4p_1q_1 + 2p_1q_0 + 2p_0q_1 + p_0q_0 - 15 = 0$$

We transform this equation into a cost function:

$$C(p_0, p_1, q_0, q_1) = (4p_1q_1 + 2p_1q_0 + 2p_0q_1 + p_0q_0 - 15)^2$$

See this: Link.

After simplifying out the coefficients and all, the QUBO matrix will look like this:

$$[Q = \begin{bmatrix} 1 & 0 & 2 & 2 \\ 0 & 3 & 2 & 4 \\ 2 & 2 & 1 & 0 \\ 2 & 4 & 0 & 1 \end{bmatrix}]$$

So, the problem hamiltonian will be (in Hamiltonian format):

$$H_{\text{problem}} = (p_0q_0 + 2p_0q_1 + 2p_1q_0 + 4p_1q_1 - 15)^2$$
$$H_{\text{problem}} = p_0q_0 + 2p_0q_1 + 2p_1q_0 + 4p_1q_1 - 30p_0q_0 - 60p_0q_1 - 60p_1q_0 - 120p_1q_1 + 225$$
$$H_{\text{problem}} = 1p_0^2 + 3p_1^2 + 1q_0^2 + 1q_1^2 + 2p_0q_0 + 4p_0q_1 + 4p_1q_0 + 8p_1q_1 - 30p_0 - 60p_1 - 60q_0 - 120q_1 + 225$$

while our initial Hamiltonian is still: $H_{\text{initial}} = -\sum_i \sigma_x^i$

We iteratively apply $H(t) = (1 - s(t))H_{\text{initial}} + s(t)H_{\text{problem}}$ until s(t) = 1. Now, the time step is dependent on the energy gap between ground and excited states. The smaller Δt, the slower the evolution and too many updates to calculate by time. So, let's **cry**.

Okay, but let's do the first iteration;

$$H_{\text{initial}} = -(\sigma_x p_0 + \sigma_x p_1 + \sigma_x q_0 + \sigma_x q_1)$$

So, let's set up our evolution parameters:

$$s(t) = \frac{T}{t}, \quad \text{where } t_1 = \Delta t,\ T = \text{total annealing time.}$$

Assume:

$$T = 1\,\text{ms},$$

$$\text{Number of steps } n = 1000,$$

$$\Delta t = \frac{T}{n} = 1\,\mu\text{s},$$

$$s(t_1) = \frac{\Delta t}{T} = 0.001.$$

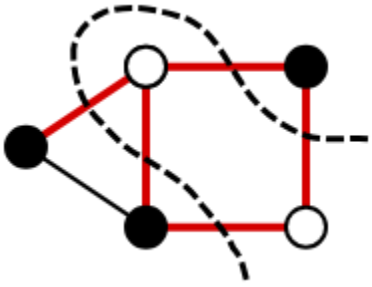
Then, for the first evolution we use this timestep of 0.001 to get:

$$H(t_1) = (1 - 0.001)H_{\text{initial}} + 0.001 H_{\text{problem}}$$

Also, substitute the hamiltonians into the problem and combine terms to get:

$$H(t_1) = -0.999(\sigma_x^{p_0} + \sigma_x^{p_1} + \sigma_x^{q_0} + \sigma_x^{q_1}) + 0.001\Big(p_0q_0 + 4p_0q_1 + 4p_1q_0 + 16p_1q_1 - 30p_0q_0 - 60p_0q_1 - 60p_1q_0 - 120p_1q_1 + 225\Big)$$

Example 2



Problem Statement: The Max-Cut Problem involves dividing the vertices of a graph into two groups such that the sum of the weights of the edges connecting the groups is maximized.

Each vertex is represented by a spin; +1 and -1; and edges are encoded as coupling terms. We remove the need for "biases"

The initial hamiltonian is $H_{\text{initial}} = -\sum_i \sigma_x^i$ and the problem hamiltonian is simply:

$$H_{\text{problem}} = \sum_{i<j} J_{ij} s_i s_j$$

5. Example

Graph:

- Vertices: $V = \{1, 2, 3, 4\}$
- Edges and weights:
  - $J_{12} = 3, J_{13} = 2, J_{23} = 4, J_{24} = 1, J_{34} = 5$

Solution:

1. Construct the Ising Hamiltonian:

$$H = 3s_1s_2 + 2s_1s_3 + 4s_2s_3 + s_2s_4 + 5s_3s_4$$

2. Use quantum annealing to minimize H.
3. Result:
   - Spins: $s = [+1, -1, +1, -1]$
   - Groups: $S_1 = \{1, 3\}, S_2 = \{2, 4\}$
   - Maximum cut: $J_{12} + J_{24} + J_{34} = 3 + 1 + 5 = 9$.

Appendix?

Related: Ising model

- It is a well-known model in statistical mechanics.
- an Ising Hamiltonian has as variables +1 and -1 which relate to spin variables (+1 is spin up, -1 is spin down)
- The relationships between the spins, represented by the coupling values of the Hamiltonian, represent the correlations or anti-correlations
  - essentially, how individual qubits (or spins) interact with each other
- Mathematically, the energy of the system is $H = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j$
  - coefficients h represent the bias values associated with the qubits
  - coefficients J represent the strength of the coupling bonds

An easier way to understand the Ising Model is through the **switch analogy**.

- **Switch States Represent Spin**:

- In the Ising model, each "spin" can take one of two values: +1 or -1.
- You can think of this as a switch with two positions:
    - +1 is ON
    - -1 is OFF

- **Switch Network Represents Interactions:**
    - **The switches are connected to each other in a network.**
    - Each connection has a "preference" for how the two connected switches should align, represented by the coupling constant J_{ij}
        - If this constant is >0, then connection prefers the switches to be in the same position (both ON or both OFF).
        - If this constant is <0, the connection prefers the switches to be in **opposite positions** (one ON and one OFF)
    - Each switch is also influenced by an external "force" or "field," represented by h_i
        - If $h_i > 0$ then external field encourages the switch to stay ON
        - If $h_i < 0$ then external field encourages the switch to stay OFF
    - The energy of the entire switch system is calculated using the Ising Hamiltonian $H = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j$
        - First term captures the energy contribution from the external fields acting on each switch
        - second term captures the energy contribution from the interactions between pairs of switches
    - The "optimal configuration" of switches is the one that minimizes the total energy H
    - This corresponds to finding the positions (ON/OFF states) for all switches that best satisfy the competing influences of the external fields and pairwise interactions.

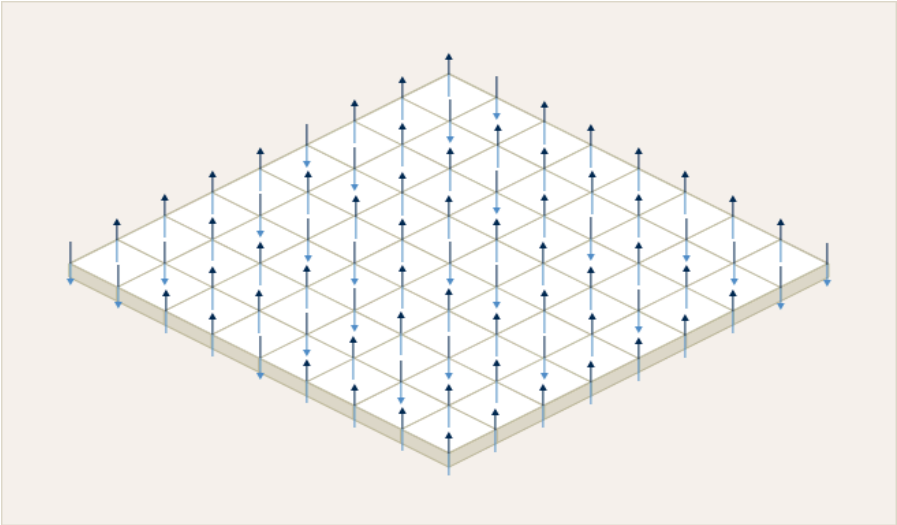


Illustration depicting an Ising Model

Related: QUBO model

- QUBO problem is defined by a matrix Q (upper triangular) and a vector of binary variables x.
- The mathematical form of a QUBO (Quadratic Unconstrained Binary Optimization) problem is given as: $f(x) = \sum_i a_i x_i + \sum_{i<j} b_{ij} x_i x_j$
  - $x \in 0, 1$ are binary variables
  - a_{ij} are linear coefficients, b_{ij} are quadratic coefficients
- For the graph:
  - Nodes A, B, C
  - Weights $w_{ab} = 3$, $w_{bc} = 2$, $w_{ac} = 4$
  - The QUBO matrix Q is: $[Q = \begin{bmatrix} 0 & -6 & -8 \\ 0 & 0 & -4 \\ 0 & 0 & 0 \end{bmatrix}]$

To convert from the **Ising model** to the **QUBO (Quadratic Unconstrained Binary Optimization)**:
https://dominicplein.medium.com/qubo-and-ising-equivalence-65275a4f4f39

Related: QUBO to Problem hamiltonian

Suppose the QUBO matrix is: Q = $\begin{bmatrix} 1 & -2 \\ 0 & 3 \end{bmatrix}$

The QUBO energy function is:

$$E(x) = x1(1) + x2(3) + x1x2(-2)E(x) = x_1(1) + x_2(3) + x_1x_2(-2)$$

The problem Hamiltonian becomes:

$$H_{\text{problem}} = Q_{11}x^1 + Q_{22}x^2 + Q_{12}x^1x^2$$
$$H_{\text{problem}} = Q_{11}\hat{x}_1 + Q_{22}\hat{x}_2 + Q_{12}\hat{x}_1\hat{x}_2$$

Related: Quantum Tunneling

- enables the quantum system to escape local minima and move toward the global minimum of the problem Hamiltonian.
  - must overcome energy barriers to escape local minima; In classical sense, this is energy we give them.

- Quantum tunneling: refers to a particle's ability to pass through a potential energy barrier that it classically does not have enough energy to overcome
    - Tunneling enables the system to transition between local minima and approach the global minimum, even if barriers separate the minima.