Quantum Computing

Tyler Chang
May 7, 2019

Fundamentals

- A *qubit* is a quantum bit, i.e., a unit with quantum properties and two logical states, usually denoted by upward and downward spin.

- It is often not possible to know the state of a qubit with certainty until it is sampled. When a qubit has non-zero probability of being in more than one state, it is said to be in a *superposition* of states.

- When a qubit is not in a superposition of states, i.e., it is known to be in a particular state with certainty, then it is said to be in a *pure* state.

- For a system of *n* qubits, there are $2^n$ pure states. If 0 denotes a downward spin and 1 denotes an upward spin, then the pure states are denoted by $|0, 0, \ldots, 0, 0\rangle$, $|0, 0, \ldots, 0, 1\rangle$, $|1, 0, \ldots, 0, 1\rangle$, $\ldots$, $|1, 1, \ldots, 1, 1\rangle$.

- To observe the state of a qubit, a sample must be taken. Observing the state of a quantum system causes *collapse*, i.e., all the probabilities in the system are destroyed.

- Two quantum particles are said to be *entangled* when their probabilities are dependent. I.e., this can be thought of as an interaction between qubits.

- Qubits are capable of jumping across an energy barrier to a lower energy state through a process called *quantum tunneling*. The probability of tunneling across a barrier decays with the width of the barrier.

- A *quantum computer* is a system of entangled qubits. Generally, to maintain the entangled states, the qubits must be stored at nearly absolute zero temperature and are extremely sensitive to vibrations.

Quantum Annealing

A quantum annealer is a special purpose quantum computer that relies on quantum tunneling to minimize the energy of a quantum system. The energy in a system of *n* qubits $\sigma = [\sigma_1, \ldots, \sigma_n]^T$ can be expressed using the *Ising-Hamiltonian* model

$$H(\sigma) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} J_{i,j} \sigma_i^2 \sigma_j^2 + \sum_{i=1}^{n} h_i \sigma_i^2$$
where $\sigma^2 \in \{-1, 1\}$ represents the spin of $\sigma_i$ (1 indicates an upward spin, -1 indicates a downward spin), $h_i$ indicates the energy associated with $\sigma_i$, and $J_{i,j}$ indicates the interaction energy (via entanglement) between $\sigma_i$ and $\sigma_j$.

The Ising model is isomorphic to the quadratic unconstrained binary optimization (QUBO) model

$$C(x) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} b_{i,j} x_i x_j + \sum_{i=1}^{n} a_i x_i$$

via the transformation $\sigma_i^2 = 2x_i^2 - 1$, $h_i = \frac{a_i}{2} + \frac{\sum_{j=1}^{n} b_{i,j}}{4}$, $J_{i,j} = \frac{b_{i,j}}{4}$. The QUBO model can also be written as $C(x) = xAx$ where $A$ is a symmetric $n \times n$ matrix, with $A_{i,i} = a_i$ and $A_{i,j} = A_{j,i} = b_{i,j}/2$.

In order to encode a QUBO or Hamiltonian, it suffices to construct a truth table for all input and output bits, where the valid states for the circuit should result in minimum energy (or ground) states in the Hamiltonian’s energy landscape. Meanwhile, invalid states for the circuit should result in excited, or higher energy states.

For example, consider the following list QUBO of constraints for encoding an AND gate ($x_1 \land x_2 = x_3$).

$$a_3 > 0$$
$$a_2 = 0$$
$$a_2 + a_3 + b_{2,3} > 0$$
$$a_1 = 0$$
$$a_1 + a_3 + b_{1,3} > 0$$
$$a_1 + a_2 + b_{1,2} > 0$$
$$a_1 + a_2 + a_3 + b_{1,2} + b_{1,3} + b_{2,3} = 0.$$ 

To satisfy these constraints, we construct the following QUBO cost function

$$C(x) = 3x_3 + x_1 x_2 - 2x_1 x_3 - 2x_2 x_3.$$ 

I.e., $a_1 = 0$, $a_2 = 0$, $a_3 = 3$, $b_{1,2} = 1$, $b_{1,3} = -2$, $b_{2,3} = -2$. This results in the following table of energies.

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$C(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Sometimes it is not possible to satisfy all the constraints to encode a function. In these cases, an ancillary bit (or ancilla bit) is introduced, whose value is assigned specifically to make the problem solvable. For example, the XOR gate \((x_1 \oplus x_2 = x_3)\) requires one ancilla bit \((x_4)\). This lends the following cost function:

\[
C(x) = x_1 + x_2 + x_3 + 4x_4 + 2x_1x_2 - 2x_1x_3 - 4x_1x_4 - 2x_2x_3 - 4x_2x_4 + 4x_3x_4.
\]

which produces the valid states \((x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 0), (x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 0), (x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 0), \) and \((x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1)\). Note, this is equivalent to a half-adder.

The above Hamiltonians/QUBOs represent logical models, which assume a fully connected graphy topology. However, in real world annealing systems, the qubits are typically connected according to some sparse topology, where only pairs of cross entangled qubits can share an interaction term. To get a physical model for a specific system from the logical model, one must create chains of connected qubits (by encoding an equality constraint between them in the Hamiltonian) until a sufficiently dense graph topology is synthesized. This physical model can be annealed on the quantum annealer hardware, which tunnels toward the ground energy state during the sampling time. In real world systems, the sampling time \(s\) needed to converge is given by \(s = \mathcal{O}(\Delta^{-2})\), where \(\Delta\) is the minimum energy gap between stability points (i.e., pure spin states). Therefore, it is typical to use relatively short annealing times and many independent runs to generate a distribution of results, with the highest probability solution(s) being returned in a post-processing step.

It should be noted that in this formulation, there is no distinction between running a circuit forward and backward (i.e., inverting). For example, for the AND gate, one could “pin” the values \(x_1\) and \(x_2\) to some predetermined values on input in order to solve for \(x_3\). Alternatively, one could pin \(x_3\), and find all the values of \(x_1\) and \(x_2\) that would satisfy \(x_1 \land x_2 = x_3\). Most notably, this implies that multiplication and factoring are equivalent operations on a quantum annealer.

Also note that Hamiltonians/QUBOs are additive, in that for two Hamiltonians/QUBOs \(H_1\) and \(H_2\) with solution sets \(X_1\) and \(X_2\), \(H_3 = H_1 + H_2\) will have the solution set \(X_3 = X_1 \cap X_2\). This allows for simple Hamiltonians to be combined to engineer complex circuits.

### Quantum Gate Model

Typically, when using the term “quantum computer,” people refer to the quantum gate model. This model is more similar to the classical computing model, and can solve a different class of problems from the quantum annealer. A general purpose quantum computer is capable of applying quantum logic gates in sequence to a systems of qubits, where these qubits need not be confined to pure states, and are allowed to exist in a superposition of states.

A system of \(n\) qubits is generally expressed as a complex vector with \(2^n\) entries, using bra-ket notation:

- \(|a\rangle\) denotes a column vector;
• $\langle a \mid$ denotes the row vector which is the Hermitian conjugate of $|a\rangle$;

• $\langle a \mid b \rangle$ denotes the inner product of $|a\rangle$ and $|b\rangle$; and

• $|a\rangle \langle b |$ denotes the outer product of $|a\rangle$ and $|b\rangle$.

The basis vectors for this complex vector space are the $2^n$ pure states described previously. The square of the modulus of each term $|a_i\rangle$ corresponds to the probability that $|a\rangle$ is in the $i$th pure state. For example, if $|a\rangle = [a_1, a_2, \ldots, a_{2^n}]^T$, then we can infer that the probability that $P(|a\rangle = |0, 0, \ldots, 0\rangle) = |a_1|^2$, $P(|a\rangle = |0, 0, \ldots, 0, 1\rangle) = |a_2|^2$, and so on. It follows that every valid state vector must have norm one, since the probabilities must sum to one. The unit sphere in $\mathbb{C}^m$ upon which all quantum states reside is called the Bloch sphere. If we agree to renormalize after addition and scalar multiplication, then the Bloch sphere can be considered as a vector space.

Since quantum operations must preserve information and probabilities, every quantum gate can be represented as a unitary matrix applied to the state vector. Since many classical gates are not reversible, it is typical to introduce ancillary qubits in a quantum circuit to preserve information that would be lost through a classical circuit. Since a sequence of unitary transformations is itself a unitary transformation, every quantum program can be represented as a single unitary transformation. Some common quantum gates are:

- Pauli spin matrices (X, Y, and Z);
- The Fredkin (CSWAP) matrix;
- The Hadamard (H) matrix;
- The Toffoli (CCNOT) and CX (CNOT) matrix.

A universal set of quantum gates, is a set of unitary matrices that can be applied in some sequence to approximate any unitary matrix (i.e., any quantum circuit) to arbitrary precision. After a quantum circuit has been run, a pure state solution is observed with probabilities determined by its posterior state vector. Therefore, a quantum circuit generally must be run many times to build a solution distribution. Since the quantum system and all its probabilities collapse after each observation, this involves a complete re-run of the quantum circuit. By starting inputs in superpositions of states, this allows for inherent parallelism.

A fault-tolerant quantum computer is one that is capable of returning correct solutions with high certainty. Theoretically, a fault-tolerant quantum computer is capable of performing any operation that a classical computer can. However, it is yet unclear whether quantum polynomial time (QP) is a superset of classical polynomial time (P). Two famous quantum algorithms that certainly improve over classical algorithms are

- Grover’s algorithm: performs a search of $n$ unordered items in a list in $O(\sqrt{n})$ time;
- Shor’s algorithm: performs integer factorization using a quantum Fourier transform.