M1 Project - Quantum Optimization

Martin PUJOL
martin.pujol@etu.sorbonne-universite.fr

Abstract

This bibliographic project aims to briefly introduce the foundational principles of quantum computing, address the concept of a qubit, ways to produce them, and the challenges associated with their usage. Subsequently, the project takes a more hands-on approach by formalizing a (simplistic) optimization problem and solving it using a quantum computer.

Keywords: Quantum Computing, Qubit, Optimization

1 What is a qubit?

1.1 Definition

A qubit, short for "quantum bit" is the basic unit of quantum computing. From a physicist's perspective, it can simply be considered as a two-level quantum system. A qubit is a quantum system that can exist in two basis states, denoted as $|0\rangle$ and $|1\rangle$. To define the state of this qubit, we need two complex numbers a and b such that $a^2 + b^2 = 1$. We can write the state of the qubit as follows:

$$|\psi\rangle = a|0\rangle + b|1\rangle \tag{1}$$

This is commonly represented as a column vector:

$$|\psi\rangle = \begin{pmatrix} a \\ b \end{pmatrix} \tag{2}$$

It is illustrated on the Bloch sphere:

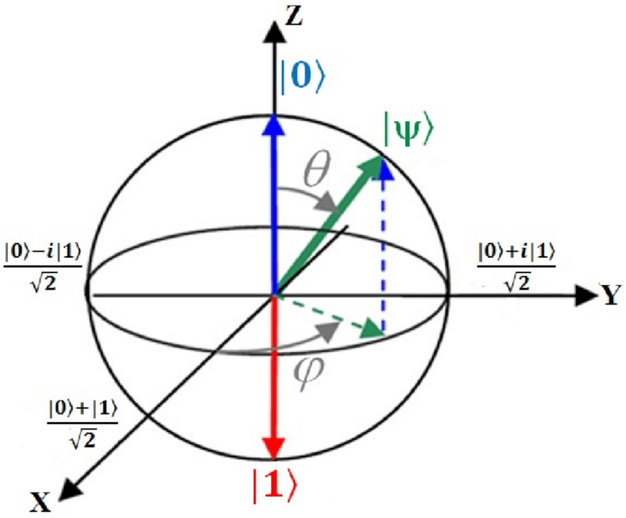

Figure 1: Bloch Sphere

Since a quantum state is defined up to a global phase factor, we can ensure that a is real. This property allows us to represent the state of a qubit in three dimensions. Adding the constraint on its norm enables us to visualize this state on the Bloch sphere (see Figure ??).

1.2 Measuring a qubit

Measuring a qubit is somewhat unique because it destroys the superposition. Measurement is an irreversible process that projects the state of the qubit onto one of its two basis states. We define two projection operators, $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$. The probability of measuring the state $|0\rangle$ is given as follows:

$$P(|0\rangle) = \langle\psi| |0\rangle \langle 0| |\psi\rangle = a^2 \tag{3}$$

And the probability of measuring the state $|1\rangle$ is given as follows:

$$P(|1\rangle) = \langle\psi| |1\rangle \langle 1| |\psi\rangle = b^2 \tag{4}$$

It becomes evident that we cannot measure the state of our qubit during computation. The challenge lies in exploiting the quantum properties of the qubit without measuring it. Only at the end of these manipulations can we measure the qubit to obtain the result using, among others, the projection operators defined earlier.

1.3 Superposition and entanglement of qubits

Introducing the notion of a qubit may not immediately reveal how computational power can be harnessed from such a system. By analogy with classical bits, we can consider what happens when using N qubits. This results in a quantum system with 2^N basis states, allowing the system to exist in a superposition of 2^N different states. For instance, the state of a 3-qubit system can be written as:

$$\begin{aligned}|\psi\rangle =& a\,|000\rangle + b\,|001\rangle + c\,|010\rangle \\ &+ d\,|011\rangle + e\,|100\rangle + f\,|101\rangle \\ &+ g\,|110\rangle + h\,|111\rangle \end{aligned} \tag{5}$$

Represented as a column vector:

$$|\psi\rangle = \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{pmatrix} \tag{6}$$

Thus, three qubits carry information of size $2^3 = 8$ continuous variables.

Superposition is particularly interesting because it allows us to "explore" various superposed states to find the solution to an optimization problem. This extensive superposition space enables compact information encoding and computation by carefully preparing and measuring states.

Entanglement introduces a useful concept that allows the creation of correlations between states, enabling the transmission of information from one state to another. Quantum states can be prepared in such a way that they cannot be described as a superposition of two independent quantum states. For example, Bell states are:ntanglement introduces a useful concept that allows the creation of correlations between states, enabling the transmission of information from one state to another. Quantum states can be prepared in such a way that they cannot be described as a superposition of two independent quantum states. For example, Bell states are:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \tag{7}$$

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle - |11\rangle\right) \tag{8}$$

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right) \tag{9}$$

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right) \tag{10}$$

For these states, measuring one qubit immediately reveals the state of the other qubit.

At this stage, the idea of quantum computation might still seem abstract, especially for the author! Fortunately, after discussing how to produce qubits, we will encode an optimization problem and solve it using a quantum computer.

2 How to produce a qubit and manipulate its state?

Since a qubit is a two-level quantum system, it can be realized using a variety of physical systems. These include atoms, ions, photons, spin defects, quantum dots, molecules, superconductors, and more. It is challenging to be exhaustive here, but both industry and research focus on a few promising technologies. These technologies, which we will concentrate on, include superconducting qubits, trapped-ion qubits, and photonic qubits.

These technologies are promising for several reasons: a qubit needs to be highly stable (maintaining its state as long as possible), scalable (easy to produce and manipulate in large numbers), and enable certain quantum properties such as entanglement.

2.1 Trapped-ion qubits

Trapped-ion qubits consist of hydrogen-like ions trapped in an electromagnetic field. The quantum system to consider here is the energy level of the ion's electron. The ions are first cooled to temperatures near absolute zero and then trapped in a quadrupole electromagnetic field, as shown below:

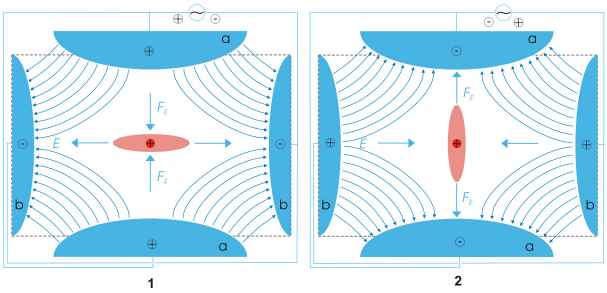

Figure 2: Paul Trap

Lasers are then used to manipulate the energy states of the electrons. These qubits are promising because of their stability, making them relatively easy to manipulate. However, they are challenging to produce and operate at scale. Companies such as Pasqal and QuEra ([7], [2]) have developed quantum computers based on this technology, particularly using Rydberg atoms. These hydrogen-like atoms have electrons in highly excited states, making them highly sensitive to electromagnetic fields and conducive to entanglement.

The Hamiltonian of a Rydberg atom is expressed as:

$$\hat{H}_0 = -\frac{\hbar^2}{2m}\nabla^2 - \frac{e^2}{4\pi\epsilon_0 r} \tag{11}$$

where m is the electron's mass, $\hbar$ is the reduced Planck constant, e is the electron charge, ϵ_0 is the permittivity of free space, and r is the electron's distance from the nucleus.

Focusing on two energy levels, denoted $|g\rangle$ (ground) and $|e\rangle$ (excited), the reduced Hamiltonian matrix becomes:

$$\hat{H} = \begin{pmatrix} E_g & \Omega/2 \\ \Omega/2 & E_e \end{pmatrix} \tag{12}$$

where E_g and E_e are the energy levels, and Ω represents the Rabi frequency, proportional to the applied electromagnetic field.

This simplified form omits the complexity of interactions between multiple electrons and coupling with electromagnetic fields. Nonetheless, it highlights the physical basis for using Rydberg atoms as qubits.

2.2 Superconducting Qubits

Superconducting qubits are electrical circuits cooled to extremely low temperatures to render them superconducting. At such temperatures, these circuits lose their electrical resistance, enabled by the formation of Cooper pairs, consisting of two electrons behaving as bosons. In these materials, all electrons can potentially form Cooper pairs and share the same quantum state, allowing the creation of qubits without the need for individual atoms.

We consider an LC circuit, where L is the inductance and C is the capacitance.

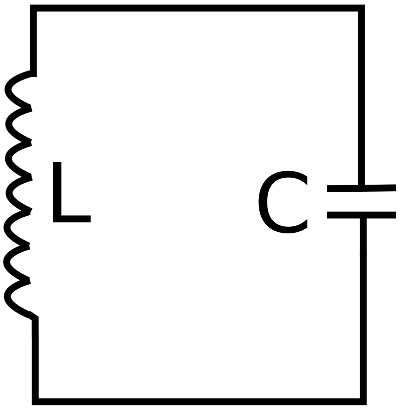

Figure 3: LC Circuit

The Hamiltonian for this circuit can be expressed as follows:

$$H = \frac{\hat{Q}^2}{2C} + \frac{\hat{\Phi}^2}{2L} \tag{13}$$

The quantized Hamiltonian above is often written in a more convenient form using the reduced charge $\hat{n} = \hat{Q}/2e$ and the phase $\hat{\phi} = 2\pi\hat{\Phi}/\Phi_0$, where $\Phi_0 = h/2e$ is the flux quantum, corresponding to the operators for the number of Cooper pairs and the phase across the Josephson junction, respectively. The quantized Hamiltonian then becomes:

$$H = 4E_c\hat{n}^2 - E_L\frac{d^2}{d\hat{\Phi}^2} \tag{14}$$

A qubit is then represented by a portion of material where all electrons share the same quantum state. Since electrons behave approximately as harmonic oscillators within the material, it is possible to focus on two energy levels of these oscillators for quantum manipulation.

An interesting aspect is the Josephson effect, which arises from the macroscopic and continuous nature of the wavefunction of Cooper pairs. This means that bringing two superconducting boxes close to each other can create a coupling between their wavefunctions, facilitating quantum entanglement.

These circuits behave as quantum oscillators that can exist in two base states. Transistors can be used to act on the state of these oscillators, enabling the production of qubits.

Superconducting quantum computers, such as those produced by IBM and Google, use a specific circuit called a transmon.

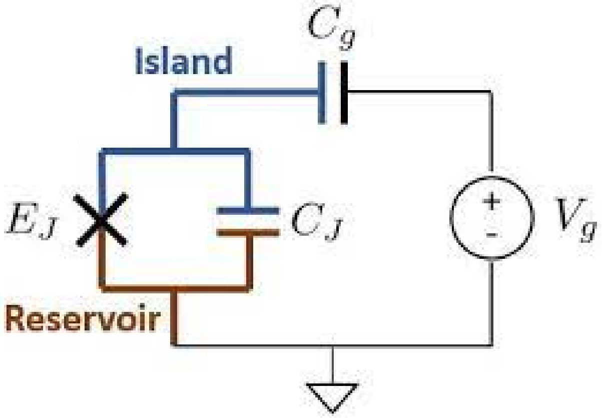

Figure 4: Transmon Circuit

Without delving into details, this circuit is analogous to an LC circuit (as shown in **??**), where the Hamiltonian is given by:

$$H = 4E_c\hat{n}^2 - E_J\cos(\hat{\phi}) \tag{15}$$

where E_c is the charging energy, E_J is the Josephson energy, and $\hat{n}$ and $\hat{\phi}$ are the operators for the number of Cooper pairs and the phase across the Josephson junction, respectively.

Transmons also resemble quantum oscillators, as shown in the figure representing their energy levels (from IBM documentation [4]):

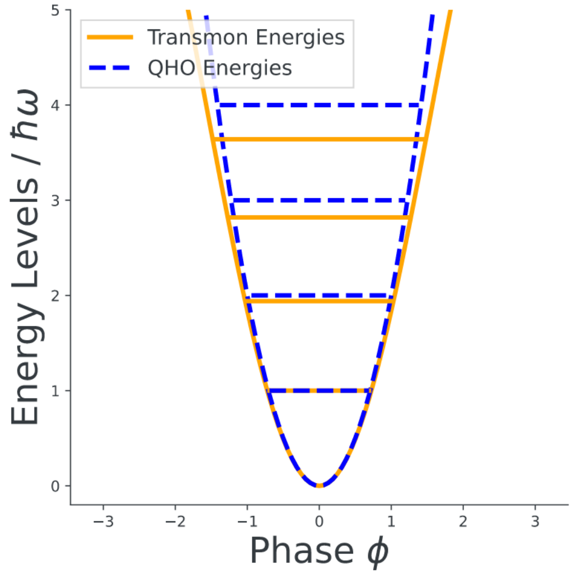

Figure 5: Energy levels of a Transmon (IBM)

2.3 Photonic qubits

A straightforward and intuitive way to create qubits is using photons. Photon polarization can be decomposed into two basis states, denoted $|0\rangle$ and $|1\rangle$. Optical fibers and devices such as polarizers are used to manipulate the state of photonic qubits.

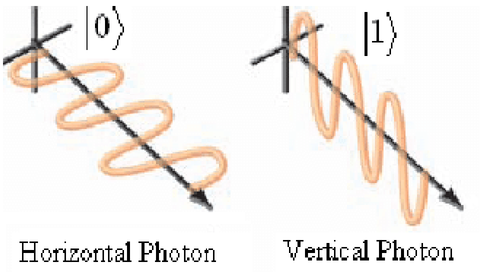

Figure 6: Photonic Qubit

Unlike other technologies, the energy Hamiltonian is not of primary interest for photonic qubits. Instead, operators like $\hat{\sigma}_z$ measure polarization states and are implemented via polarizing filters.

2.4 Which technology to choose?

It is challenging to determine which technology is the most promising. Each has its advantages and disadvantages. IBM, with its superconducting qubits, and Pasqal, with its trapped-ion qubits, appear highly scalable, allowing for a large number of qubits. However, applying logical operations to these qubits can modify the information, such as flipping a qubit from the state $|0\rangle$ to $|1\rangle$. Thus, gate fidelity is a critical issue, though companies rarely disclose such details. To mitigate errors, multiple qubits can be used for error correction—a process called quantum error correction.

Pasqal, for instance, still does not allow logical gates on its qubits, and there are other limitations. For example, trapped-ion superconducting qubits allow for entanglement only between neighboring qubits (spatially), making it hard to entangle distant qubits. Other companies take different approaches, such as Alice Bob, which focus on cat qubits immune to specific types of errors.

For quantum gate-based computers, performance can be gauged by two metrics: the fidelity of 2-qubit gates and the number of physical qubits. Achieving the NISQ (Noisy Intermediate-Scale Quantum) stage—a key milestone—requires over 40 physical qubits and 2-qubit gate fidelities exceeding 99.9%. The figure below (**??**) from [3] visualizes the performance of various technologies based on the number of qubits and gate fidelities.

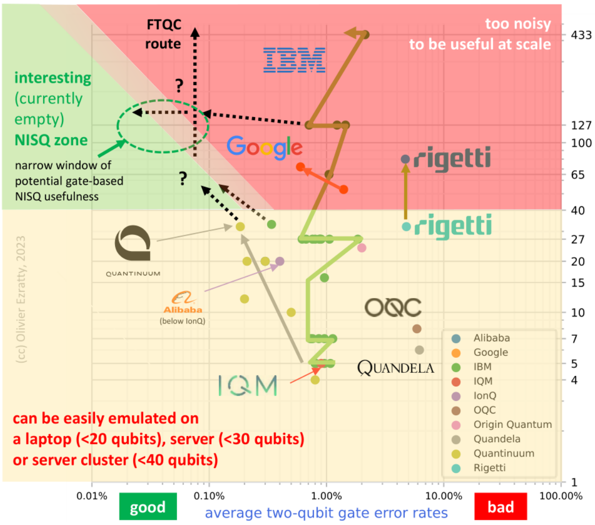

Figure 7: Performance of different technologies by qubit count and 2-qubit gate fidelity

To date, the most advanced technology is superconducting qubits, with IBM and Google producing quantum computers exceeding 50 qubits. While gate fidelities are not exceptionally high, IBM provides a variety of gates and highly effective error-correcting codes ([1]).

On the other hand, trapped-ion qubits offer the most promising scalability. Experiments have demonstrated the ability to maintain chains of thousands of these ions, and Quantinuum's efforts to deliver highly stable qubits are particularly encouraging.

Recent developments by Quantinuum and Microsoft have introduced new error-correcting codes that reduce errors by a factor of 800 ([8]), positioning Quantinuum as a potential leader in quantum computing.

It is worth noting that no two quantum computers are identical. Cat qubits are immune to certain errors, trapped-ion qubits exhibit unique physical properties like the Rydberg blockade, and photonic quantum computers operate very differently, offering specialized advantages for specific applications.

3 Algorithms and Quantum Optimization

To manipulate the state of a qubit, we use operators. These operators mathematically represent the effects of physical processes or measurements on a qubit's state. When focusing solely on the abstract mathematical description of the state (defined by the coefficients a and b in $|\psi\rangle = a|0\rangle + b|1\rangle$), the specific physical realization of the qubit becomes less relevant. Whether the qubit is implemented using a Rydberg atom, a superconducting circuit, or a photon, the same operators can be applied to manipulate the qubit state, such as flipping $|0\rangle$ to $|1\rangle$ and vice versa.

3.1 Quantum Gates

Quantum gates are the algebraic representations of operators acting on a qubit's state. These gates perform specific transformations, such as rotations or state flips, and are fundamental building blocks in quantum algorithms. Their physical implementation depends on the underlying quantum hardware.

3.1.1 Pauli gates

Pauli gates act on a qubit's state by rotating it about the x, y, or z-axes. They are defined as follows:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{16}$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{17}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{18}$$

3.1.2 Hadamard gate

The Hadamard gate rotates a qubit about both the x- and z-axes:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{19}$$

3.1.3 Phase gate

The phase gate rotates a qubit about the z-axis:

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{20}$$

3.1.4 CNOT gate

The CNOT gate acts on two qubits by flipping the second qubit based on the state of the first qubit:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{21}$$

These fundamental gates, when combined, allow for any logical operation to be performed on the states of qubits.

3.2 Quantum optimization

One significant application of quantum properties is solving optimization problems. If we can encode an optimization problem's cost into the Hamiltonian of a quantum system, finding the ground state of that system corresponds to solving the optimization problem.

3.2.1 Adiabatic theorem

To find the ground state of a quantum system, we rely on the adiabatic theorem. This theorem states that if a quantum system starts in the ground state of its Hamiltonian, and the Hamiltonian changes slowly, the system will remain in the ground state of the Hamiltonian.

Mathematically, if $\hat{H}(t)$ is the system's Hamiltonian at time t, and it evolves as:

$$\hat{H}(t) = (1 - t/T)\hat{H}_0 + (t/T)\hat{H}_f \tag{22}$$

where $\hat{H}_0$ is the initial Hamiltonian and $\hat{H}_f$ is the final Hamiltonian at $t = T$, then for sufficiently large T, the ground state of $\hat{H}_0$ evolves into the ground state of $\hat{H}_f$.

This variation must satisfy the following condition:

$$\tau \gg \frac{\max_{0\leq s\leq 1} \left| \left\langle \psi_1(s) \left| \frac{\partial \hat{H}(s)}{\partial s} \right| \psi_0(s) \right\rangle \right|}{\min_{0\leq s\leq 1} \Delta^2(s)} \tag{23}$$

where τ is the evolution time, $\psi_0(s)$ and $\psi_1(s)$ are the ground and first excited states of the Hamiltonian $\hat{H}(s)$, and $\Delta(s)$ is the energy gap between these states.

3.2.2 Ising Hamiltonian

To encode our optimization problem, we use the Ising Hamiltonian:

$$\hat{H} = -\sum_{i,j} J_{ij}\hat{\sigma}_i^z\hat{\sigma}_j^z - \sum_i h_i\hat{\sigma}_i^z \tag{24}$$

Here, $\hat{\sigma}_i^z$ is the Pauli-z operator, J_{ij} are coupling weights between spins, and h_i are local magnetic fields.

A critical point is ensuring that the initial Hamiltonian $\hat{H}_0$ does not commute with the Ising Hamiltonian. Otherwise, the initial state would remain stationary for both Hamiltonians, preventing evolution.

3.3 Optimization problem: Max-Cut

The Max-Cut problem is a combinatorial optimization task that involves dividing a graph into two subsets to maximize the number of edges crossing between them. The input consists of the graph's nodes and edges. It is categorized as NP-Hard, meaning classical algorithms solve it in exponential time. However, a quantum computer can solve this problem more efficiently.

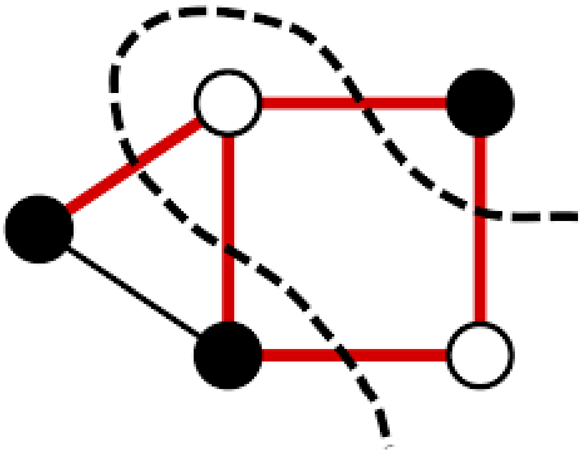

Figure 8: Max-Cut problem

For example, consider a graph with two nodes:

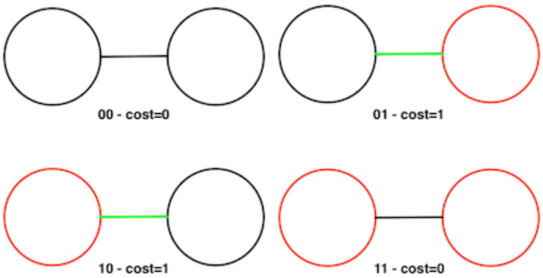

Figure 9: Max-Cut with 2 nodes

Here is another example with three nodes:

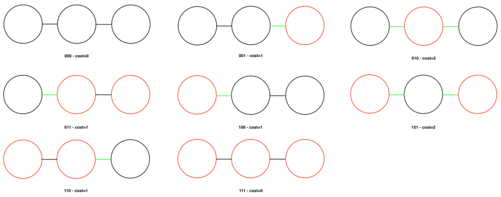

Figure 10: Max-Cut with 3 nodes

The cost function for the problem can be expressed as:

$$C_{total} = \sum C_{ij} = \sum \frac{1}{2} w_{ij}(1 - z_i z_j) \tag{25}$$

where $w_{ij} = 1$ if i and j are connected by an edge, 0 otherwise. The binary variables z_i and z_j define whether a node belongs to one of the subsets.

3.4 Quantum encoding of the problem: Which Hamiltonian?

The Max-Cut problem is well-suited for encoding into an Ising Hamiltonian. Each node of the graph corresponds to a qubit, and coupling weights are assigned between qubits if their respective nodes are connected by an edge.

The Ising Hamiltonian is then defined as:

$$\hat{H} = -\sum_{i,j} J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z - \sum_i h_i \hat{\sigma}_i^z \tag{26}$$

with coupling weights:

$$J_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are connected by an edge} \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

and local magnetic fields:

$$h_i = 0 \tag{28}$$

The state of each qubit ($|0\rangle$ or $|1\rangle$) determines its subset, avoiding the need for additional qubits to encode this information (as in classical bits).

3.5 Implementation

To solve the Max-Cut problem, we use the Leap library to encode the problem and execute it on a D-Wave quantum annealer. D-Wave devices specialize in quantum annealing, allowing us to define the Hamiltonian for our qubits without implementing logical gates.

The code below, based on prior work ([5], [6]), first formulates the problem as a graph:

```
G = nx.Graph()
G.add_edges_from([
        [0,3],[0,4],[1,3],[1,4],
        [2,3],[2,4],[3,5],[1,3],
        [6,7], [2,7], [5,7], [3,5]
 ])
nx.draw(G,
        pos=nx.bipartite_layout(
            G, [0,1,2]
        ))
```

This produces the following graph:

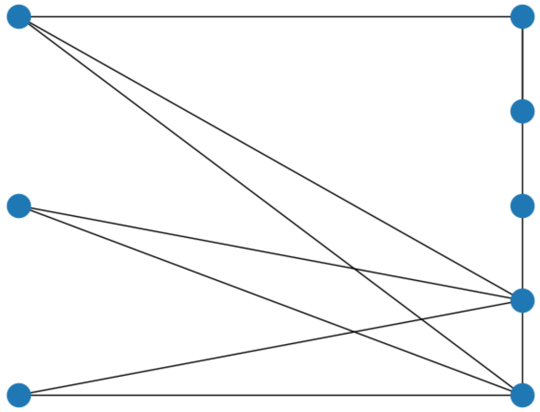

Figure 11: Graph for the problem

The graph is then converted into an Ising Hamiltonian:

```
# Initialize h vector and J matrix
h = defaultdict(int)
J = defaultdict(int)

# Update J matrix for each edge in the graph
for i, j in G.edges:
    J[(i,j)] += 1
```

A sampler instance, corresponding to the quantum computer, is created:

```
# Initialize quantum sampler
client = Client.from_config(
    config_file="dwave.conf",
    profile="bqm"
)
sampler = EmbeddingComposite(
    DWaveSampler()
)

# Define parameters
chainstrength = 2
numruns = 10
```

Finally, the Hamiltonian is submitted to the sampler, which returns the optimized solution:

```
# Solve the problem
response = sampler.sample_ising(h, J)
```

The solution is presented as follows:

0	1	2	3	4	5	6	7
-1	-1	-1	+1	+1	-1	-1	+1
+1	+1	+1	-1	-1	+1	+1	-1

Table 1: Maximum cut solution for the graph

The solution can be visualized as follows:

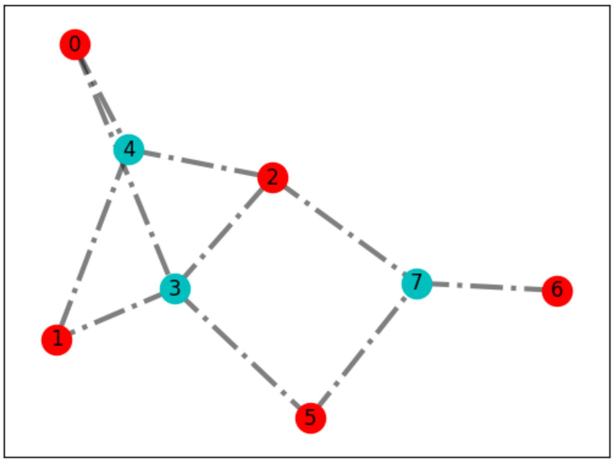

Figure 12: Visualization of the maximum cut solution

4 Conclusion

This bibliographic project introduced the fundamental principles of quantum computing, particularly the concept of qubits, methods for producing them, and associated challenges. It also demonstrated solving a classical optimization problem, the Max-Cut, using a quantum computer.

Qubits, with their unique properties like superposition and entanglement, enable computations far more efficient than classical bits. However, their manipulation remains delicate due to the irreversible nature of measurement.

We discussed several technologies for producing qubits, such as superconducting qubits, trapped-ion qubits, and photonic qubits, each with its strengths and weaknesses.

Finally, the project explored solving the Max-Cut problem using quantum annealing, showcasing how quantum computers can encode and optimize problems efficiently. As quantum computing continues to develop, it holds immense promise for addressing complex computational challenges.

References

[1] Sergey Bravyi et al. “High-threshold and low-overhead fault-tolerant quantum memory”. In: *Nature* 627.8005 (Mar. 2024). Publisher: Nature Publishing Group, pp. 778–782. ISSN: 1476-4687. DOI: 10.1038/s41586-024-07107-7. URL: https://www.nature.com/articles/s41586-024-07107-7 (visited on 05/06/2024).

[2] *Building Quantum Computers with Neutral Atoms — QuEra*. URL: https://www.quera.com/neutral-atom-platform (visited on 02/03/2024).

[3] Olivier Ezratty. *Where are we heading with NISQ?* June 12, 2023. DOI: 10.48550/arXiv.2305.09518. arXiv: 2305.09518[quant-ph]. URL: http://arxiv.org/abs/2305.09518 (visited on 05/04/2024).

[4] *Introduction to Transmon Physics*. URL: https://community.qiskit.org/textbook/ch-quantum-hardware/transmon-physics.html (visited on 02/04/2024).

[5] Siddharth Jain. "Solving the Traveling Salesman Problem on the D-Wave Quantum Computer". In: *Frontiers in Physics* 9 (2021). ISSN: 2296-424X. URL: https://www.frontiersin.org/articles/10.3389/fphy.2021.760783 (visited on 02/05/2024).

[6] Michał Stechły. *mstechly/quantum_tsp_tutorials*. original-date: 2018-08-13T05:57:03Z. Jan. 8, 2024. URL: https://github.com/mstechly/quantum_tsp_tutorials (visited on 02/05/2024).

[7] Karen Wintersperger et al. "Neutral atom quantum computing hardware: performance and end-user perspective". In: *EPJ Quantum Technology* 10.1 (Dec. 2023). Number: 1 Publisher: SpringerOpen, pp. 1–26. ISSN: 2196-0763. DOI: 10.1140/epjqt/s40507-023-00190-1. URL: https://epjquantumtechnology.springeropen.com/articles/10.1140/epjqt/s40507-023-00190-1 (visited on 01/31/2024).

[8] Jason Zander. *Advancing science: Microsoft and Quantinuum demonstrate the most reliable logical qubits on record with an error rate 800x better than physical qubits*. The Official Microsoft Blog. Apr. 3, 2024. URL: https://blogs.microsoft.com/blog/2024/04/03/advancing-science-microsoft-and-quantinuum-demonstrate-the-most-reliable-logical-qubits-on-record-with-an-error-rate-800x-better-than-physical-qubits/ (visited on 05/06/2024).