

Quantum Annealing and Experience with D-Wave Systems ¹

Dhruvil Gandhi, Akshay More

Seidenberg School of CSIS, Pace University, Pleasantville, New York 10570

Email: {dgandhi, amore}@pace.edu

Abstract—There are different forms of quantum computing. D-Wave Systems offers adiabatic quantum computing through quantum annealing. We explore the principle of quantum annealing, how adiabatic quantum computing approach works, understand application of quantum annealing and how to implement example algorithms on their system using their SDK.

Index Terms—Quantum Computer, Quantum Annealing, Optimization Algos, D-Wave, Hamiltonian

I. INTRODUCTION

Quantum Computing has evolved significantly this decade. With a lot of services, infrastructures and tool kits commercially available to the public, there has been a great advancement in the field of quantum computing. A quantum computer takes advantage of the property of qubit to exist in more than one state at any time. Compared to classical computing, based on binary bits 0 and 1, quantum computers are based on 'qubits', which can store a lot more information than classical bit due to its property to be able to exist in more than one state. [1]

Various companies have made their quantum computers available to the public, IBM with Q-Experience, D-Wave Systems with D-Wave 2000Q, Rigetti, Alibaba, Microsoft, Google, to name a few. All these quantum computers work on different approaches of quantum computing. [2]

IBM created the first universal quantum computer. D-Wave created adiabatic quantum

computer. [3] There are a lot of resources available to learn and experiment with these systems. Through the paper, we briefly understand what a universal quantum computer is, different types of quantum computing, advantages and disadvantages. We explore D-Wave Quantum Computer, the principle it works on, implement optimization algorithm and show the results.

II. BACKGROUND

A. Universal Quantum Computer

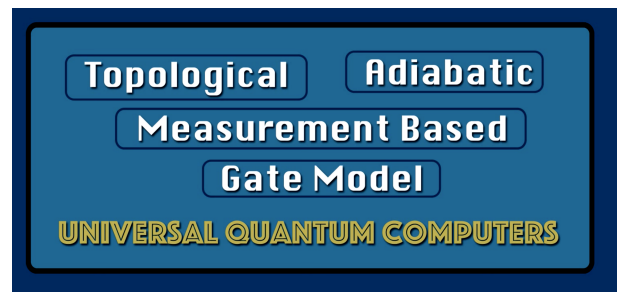


Fig. 1: Universal Quantum Computers

The first quantum computer was built using the gate model approach where we control and manipulate the state of the Quantum bits or qubits to perform operations and run algorithms. The Universal Quantum Computer is one where the gate model approach is able to map to other forms of quantum computing in polynomial time and polynomial resources. Various approaches of quantum computing are [4]:

- Circuit Based Quantum Computing
- Adiabatic Quantum Computing
- Topological Quantum Computing

¹This paper is for CS 837 at Pace University and the class is supported by IBM's grant.

- Measurement Based Quantum Computing

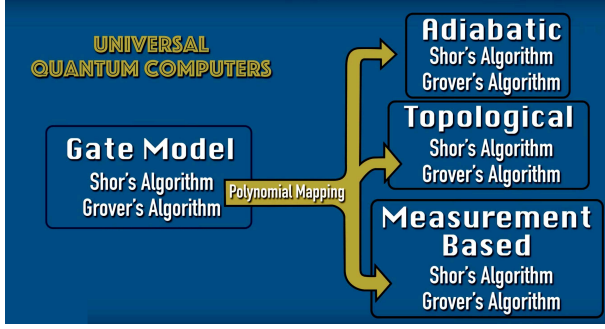


Fig. 2: Polynomial mapping

To be a universal quantum computer, a quantum computer must be able to operate all the above approaches of quantum computing.

1) *Gates based:* In gate based quantum computing, the gates are built on the principle of qubits: *superposition* and *entanglement*.

a) *Superposition:* Qubit can exist in both 0 and 1 state. It can be represented by the equation:

$$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle$$

where a_0 and a_1 are configuration that specifies all the possibilities of state being in 0 and 1. [1]

b) *Entanglement:* In quantum computing, in a multi-particle system, entanglement is property of qubits where the state of one qubit cannot be described independently. To express the state of system with two Particles A and B, the quantum state can be described as tensor product of two state space described by the equation: $|\Phi\rangle = |A\rangle \otimes |B\rangle$ or, it can be also written as $|\Phi\rangle = |AB\rangle$

2) *Adiabatic Quantum Computer:* Adiabatic quantum computing is based on Hamiltonian framework for quantum mechanics. Quantum systems evolves by unitary operators. We find the initial easily implementable Hamiltonian H_0 , find a path H_t and find the final Hamiltonian H_1 keeping the system in ground state which depends on eigen gap. The basic principle of the adiabatic quantum computing is: as long as the path is traversed slowly from H_0 to H_1 , the system remains in

ground state and the required solution is obtained. If traversed faster, the system jumps state and it results in error.

a) *Hamiltonian Formula:* The Hamiltonian formula can be simplified as follows:

$$\mathcal{H}_s(s) = \frac{-1}{2} \sum_i \Delta(s) \sigma_i^x + \epsilon(s) \left(- \sum_i h_i \sigma_i^z + \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z \right)$$

which can be broken down as:

- $\mathcal{H}_s(s)$ is the required Hamiltonian energy.
- $[\frac{-1}{2} \sum_i \Delta(s) \sigma_i^x]$ is the initial Hamiltonian, where all the qubits are in superposition state.
- $\epsilon(s) \left(- \sum_i h_i \sigma_i^z + \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z \right)$ is the final Hamiltonian, the answer to the problem, which contains the biases and couplers.

One thing to note is that the initial states are Quantum states and the final states are classical states.

b) *Eigen Value and Anti-crossing:* In adiabatic system, the systems stays in ground state through out the cycle of getting solution. Staying in ground state depends on speed of traverseing, if too fast or the thermal energy from physical systems. The system initially is at 0 energy level and at the end, it is at zero energy level. During the process, the systems's state might jump and escape from ground state to higher energy state based on the energy gap between two states, which is called anti-crossing and Eigen value of states.

III. QUANTUM ANNEALING FOR PROBLEMS

[4] Quantum Annealing is done using the adiabatic quantum computing approach.

A qubit can exist in two state due to superposition initially, 0 or \downarrow spin and 1 or \uparrow . The energy state at superposition is lowest.

Once quantum annealing is applied, each qubit goes into either 0 or 1 state, ie classical

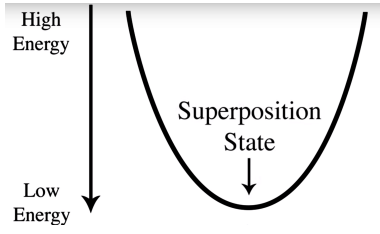


Fig. 3: Superposition qubit energy

state, the energy is represented by a double well potential diagram.

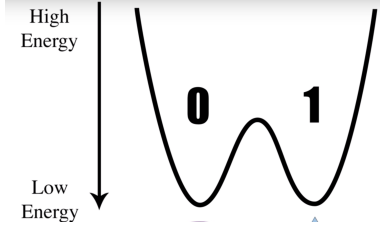


Fig. 4: Double well potential of a qubit

To get the desired answer, the energy of this double well potential is controlled by influence of a magnetic field, known as *bias*.

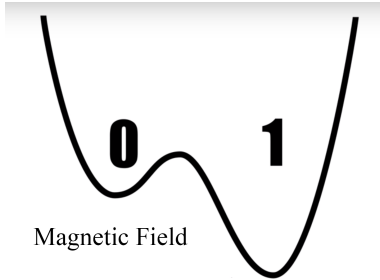


Fig. 5: Biased qubit double well potential

To influence all the qubits in system, qubits are linked by the entanglement property of qubits. This linking is called *coupling*. There are two types of coupling: 0-0 or 1-1 coupling and 1-0 or 0-1 coupling.

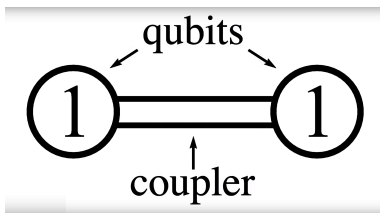


Fig. 6: Coupling between two qubit

Hence the potential of this coupled qubits depends on biases of each qubits and the coupling

between them. They can be present in either 00,01,10 or 11. The outcome is controlled by the problem we define, at the end the answer is the lowest potential state of the two qubits. Taking all the qubits into account, we get a energy landscape, where the minima on energy curve is the answer to the problem we are solving.

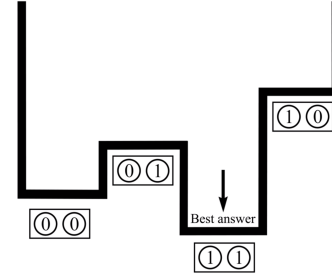


Fig. 7: Double well potential of a qubit

The combination (C) of states of a quantum computer for quantum annealing depends on number of qubits (n) defined as: $C = 2^n$.

To compare this with universal gate based quantum computer, the main difference lies in the way a problem is solved. In gate based quantum computer, the states of individual qubits is controlled and the solution is obtained. In quantum annealing, the problem is formulated, desired Hamiltonian is calculated, finally the system after going through annealing schedule, calculates the solution, which is the minima on energy curve of the ground state of quantum system. Quantum annealing is useful for optimization problem, where the goal is to get best possible solution with lowest cost. Quantum annealing is not good to solve search algorithms like Grover's search algorithm.

IV. D-WAVE SYSTEMS AND USING ITS SDK AND CLOUD

D-Wave Systems Inc has been developing quantum computers, which currently offers quantum computer with 2038 Qubits, which works on the principle of quantum annealing, as explained above. Quantum annealing can be used for areas including: [3]

- Optimization
- Machine learning
- Cybersecurity
- Image analysis
- Pattern recognition
- Financial analysis
- Bioinformatics/cancer research
- Traffic flow
- Manufacturing Process

The system is based on superconducting device at absolute zero temperature.

a) *Ocean SDK*: D-Wave offers Ocean SDK which has a set of tools to run, simulate and implement quantum algorithms. The D-Wave system provides a quantum processing unit (QPU) to solve a binary quadratic model (BQM). The are advanced features and tools for further improvements of the solution. There are numerous tools provided in the SDK including binary and quadratic sampler, annealing sampler, Ising problems micro-embedding. Some of the examples are available online on github [5] and on ocean's doc website.⁶

b) *LEAP*: Leap is a real time environment from D-Wave which provides tools and access to quantum computing in cloud to its uses. It has interactive examples which explains implementation of integer factoring problem and structural imbalance problem with cases, and comparison of time taken to solve it on D-Wave vs classical computer. Ocean's SDK can be implemented on LEAP.

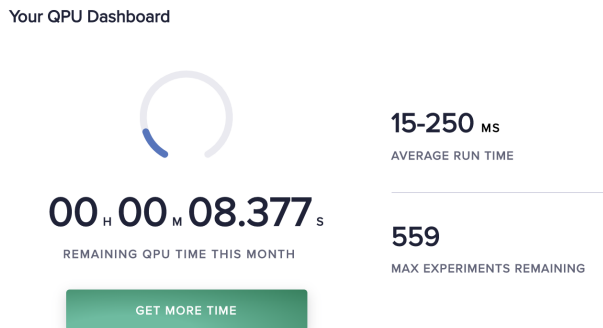


Fig. 8: Leap Dashboard 1

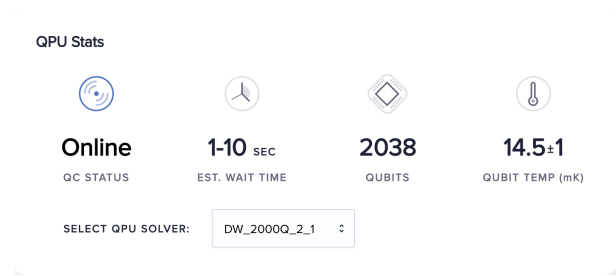


Fig. 9: Leap Dashboard

V. EXPERIMENTATION

To experiment with D-Wave system, we tried to use the LEAP system and run Integer Factoring and Structural Imbalance problem. Besides LEAP, we implemented demo examples from Ocean SDK, both of which can be found on the github [?] and in documents [6].

A. Map Coloring

This example is implemented using D-Wave's Ocean SDK.

Map coloring problem is a constraint satisfaction problem. The primary requirement of a constraint satisfaction problem is to find a result from finite set of domain, values are assigned to a problem, staisfying all constraint.

In map coloring example, all regions are to be colored and the constraint is that for any neighbouring regions, the color should not be same. In this case, the regions are states of Canada.

To implement this using Ocean SDK, we followed and used the example given on D-Wave's documentation [6]. The steps briefly to solve the problem are:

- Formulate problem as graphic
- Create binary constraint satisfaction problem
- Convert to binary quadratic model (BQM)
- Sample using D-Wave SAPI solver
- Plot a valid solution.

B. Integer Factoring

Integer factoring is a P and NP-hard problem for classical computers. For quantum computing it runs in polynomial time when universal gate model approach is used, currently for which, technology does not allow the possibility for large numbers.

Using annealing, we can represent integer factoring as constraint satisfaction problem. For integer factoring, the constraints are to whole number factors a and b and their multiplication be equal to factored number P .

1) *On D-Wave:* D-Wave systems solve binary quadratic models (BQM), the Ising model traditionally used in statistical mechanics and its computer-science equivalent, the quadratic unconstrained binary optimization (QUBO) problem. Given N variables x_1, \dots, x_N , where each variable x_i can have binary values 0 or 1, the system finds assignments of values that minimize [?]

$$\sum_i^N q_i x_i + \sum_{i < j}^N q_{i,j} x_i x_j,$$

where q_i and $q_{i,j}$ are configurable (linear and quadratic) coefficients. To formulate a problem for the D-Wave system is to program q_i and $q_{i,j}$ so that assignments of x_1, \dots, x_N also represent solutions to the problem.

There are different ways to formulate the factoring problem as a BQM. We program D-Wave system using following steps:

- Express $P = ab$ as a CSP with a binary multiplication circuit.
- Convert to a BQM.
- Program the quantum computer with the BQM's coefficients.

C. Optimisation - Structural Imbalance

Structural imbalance can be defined as a measure for a network which can not be cleanly divided, is the minimum number of edges which violates the triadic relation theory. To solve

structural imbalance, finding a division that minimizes this violation is a NP hard problem. It is an example of optimization, other include traffic flow, protein-folding etc.

The problem is similar to Ising problem. The problem is formulated using Ocean's penalty model. The micro-embedding feature is used to embed 200 nodes in the machine.

The basis for solving the given Militants mapping project is done using penalty model where undesirable outcomes are given a higher potential to be eliminated from the solution.

VI. EXPERIENCE AND RESULT

Quantum annealing proves to be useful in general to solve problems and find solution in terms of time required as compared to a quantum computer. D-Wave's Ocean SDK and LEAP is useful tool to learn quantum annealing and how to practically use it and implement to solve problems. It provides an insight to how just using adiabatic quantum computing approach, number of qubits can be increased in a quantum computer as compared to a universal gate based quantum computer and the advantages and disadvantages of adiabatic quantum computing.

Setting up Ocean SDK gives the connection to D-Wave's QPU (Quantum Processing Unit).

```
Using endpoint: https://cloud.dwavesys.com/sapi
Using solver: DW_2000Q_2.1
Submitted problem ID: cbaa6d72-1706-41be-8a65-cd0094f6af30

Wall clock time:
* Solver definition fetch: 953.228 ms
* Problem submit and results fetch: 657.055 ms
* Total: 1610.284 ms

QPU timing:
* total_real_time = 7712 us
* qpu_access_overhead_time = 489 us
* anneal_time_per_run = 20 us
* post_processing_overhead_time = 369 us
* qpu_sampling_time = 164 us
* readout_time_per_run = 123 us
* qpu_delay_time_per_sample = 21 us
* qpu_anneal_time_per_sample = 20 us
* total_post_processing_time = 369 us
* qpu_programming_time = 7549 us
* run_time_chip = 164 us
* qpu_access_time = 7712 us
* qpu_readout_time_per_sample = 123 us
```

Fig. 10: Ocean SDK ping status

The result for the algorithms implemented above are shown below.

A. Map Coloring Problem

For map coloring problem, the following graph plot was obtained.

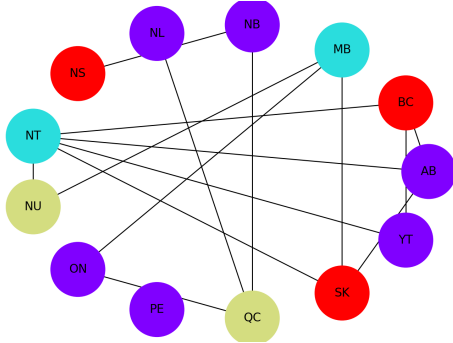


Fig. 11: Map Coloring Solution

B. Integer Factoring

For integer factoring, there is an interactive demo on LEAP. We implemented the available demo on LEAP using Jupyter notebook provided with the resources. For 21, 7 and 3 were obtained. The energy sample for solution is shown in the figure below.

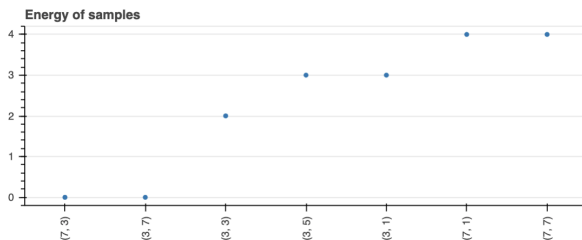


Fig. 12: Energy Sample Graph for 21

C. Structural Imbalance

Both interactive demo and the jupyter notebook were implemented. For structural imbalance, the experiment had to be run multiple times as the mapping of graph to the problem is done using heuristic minor-embedding which doesn't behave as expected and fails.

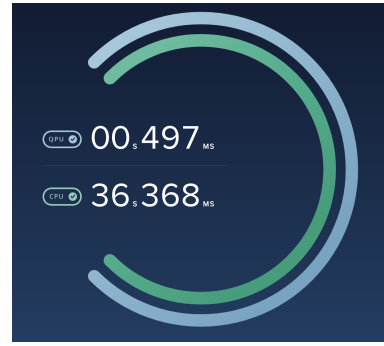


Fig. 13: Solution time: QPU vs CPU

VII. FUTURE STEPS

All the work that was carried out is in regards to Quantum Annealing. For next steps we want to try out all the available advanced features on D-Wave and implement advanced algorithms to solve problems.

VIII. CONCLUSION

D-Wave's quantum computer are powerful quantum computer for quantum annealing. The resources and example given can be easily understood and implemented. Quantum Annealing is useful to solve problem where the best possible solution is required. This feature can also be used in machine learning, authentication and validation and analysis. Using leap's interactive examples, we understand how solving these problems compare on a classical computer versus a quantum computer and the advantage it provides in terms of time.

REFERENCES

- [1] Nielsen, Michael A and Chuang, Isaac L *Quantum computation and quantum information*
- [2] Dong Finke, *Qubit Count* <https://quantumcomputingreport.com/scorecards/qubit-count/>
- [3] D-Wave 2000Q Announcement <https://www.dwavesys.com/press-releases/d-wave%20announces%20d-wave-2000q-quantum-computer-and-first-system-order>
- [4] D-Wave Resource Videos <https://www.dwavesys.com/resources/videos>
- [5] Ocean SDK Demos <https://github.com/dwavesystems/demos>
- [6] D-Wave documentation <https://docs.ocean.dwavesys.com/en/latest/index.html>
- [7] Paper Repo <https://github.com/dhruv857/Quantum-Annealing>