Quantum Computing: A Brief Background

Amr Magdy[1] Moustafa Youssef[2] Ibrahim Sabek[3]

[1]University of California, Riverside [2]The American University in Cairo
[3]University of Southern California

Quantum computing has shown significant potential for addressing complex optimization and computationally intensive problems in various domains. Quantum computing currently has two major models, the quantum circuit model and quantum annealing. This article gives a very brief background about the two models. The article is under development and will be updated.

1 Fundamental Concepts

This section introduces basic definitions of fundamental quantum computing concepts.

Qubit. A *quantum bit* (qubit) is the basic unit of information and is analogous to the classical bit. The classical bit, i.e., binary digit, can be in only one of two states at any time, taking the value 0 or 1, but not both. Contrary to classical bits, a qubit can exist in a *superposition* of the two states (zero and one states simultaneously). This superposition allows quantum computations to work on both states at the same time, a phenomenon often referred to as *quantum parallelism*. Qubits can have various physical implementations, e.g., the polarization of photons or the spin of electrons.

Quantum superposition is the odd-but-powerful natural quantum rule that allows a single particle, e.g., electron or a light photon, to exist in several possible states at the same time, like a coin that is both heads and tails until you look at it. In quantum computing, the quantum states are binary, either 0 or 1, to allow quantum computers to implement binary qubits. The power of superposition allows a small collection of qubits to explore many calculation paths in parallel and achieve promising speed-ups on certain problems once the result is finally *"observed"* and collapses to a single answer.

Mathematically, the superposition is modeled as a *wave function* of possibilities that represent the probabilities of the qubit to exist in each distinct state. Typically, a qubit in a superposition is represented by the Dirac notation (or bra-ket notation) as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are called the *probability amplitudes* of the classical states $|0\rangle$ and $|1\rangle$, respectively. The probability amplitudes represent probabilities of α^2 and β^2 for a qubit to exist in states $|0\rangle$ and $|1\rangle$, respectively, where $\alpha^2 + \beta^2 = 1$.

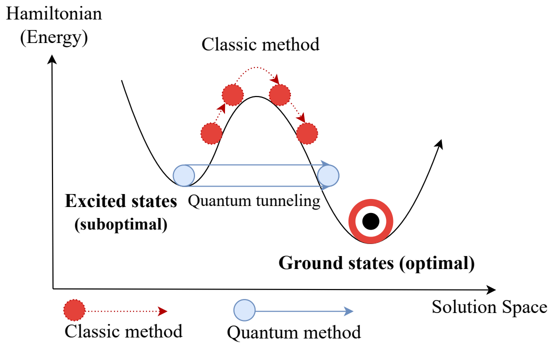

Figure 1: Quantum Annealing Process VS Classic Method

Quantum measurement. When an *"observer"* device measures (or looks at) at a qubit, this act of measurement counts as a physical interaction that forces the qubit's spread-out wave of possibilities to snap into one specific outcome, a sudden change known as wave-function collapse. Until that instant the particle lives in a superposition, meaning it simultaneously embodies all the states, e.g., places or spins, it could occupy. However, the act of checking (i.e., measurement) wipes out that blur and leaves the particle with a single, definite result every time.

2 Quantum Annealing

2.1 Overview

Quantum annealing (QA) has been proposed as a novel heuristic for combinatorial optimization problems [4], and it is similar to Simulated Annealing (SA). In SA, the configuration space is explored by randomly proposing changes to the state configuration: if the new value of the cause function is decreased, the proposed change is accepted; if the cost function increases, the change is accepted with some probability. In QA, these changes are applied on a quantum superposition of the configurations, allowing for a massively parallel search of the configuration space. This allows phenomena like quantum tunneling to facilitate the exploration, potentially avoiding getting stuck in local minima.

In QA, a quadratic cost function over binary variables is encoded in the interactions between two-level quantum mechanical systems, i.e., qubits, in a way that the solution to the problem of interest is the configuration that minimizes the energy of this interacting system. We can initialize the physical system in a state that is a quantum superposition of all possible configurations, and then we proceed to change the interaction strengths between the qubits slowly. This annealing process (if performed slowly enough) transforms the initial superposition into the state that minimizes the system's energy, which, in turn, encodes the solution to the combinatorial optimization problem.

The D-Wave quantum annealer [3] implements this model using superconducting technology, where small superconducting loops represent individual qubits: current circulating around these loops in both directions encode the 0 and 1 states of the qubit. For two qubits that are close to each other, a programmable interaction can be implemented between them, as well as local biases on the individual qubits. With these elements, an interacting system can be constructed whose energy takes the form

$$E(\vec{s}) = \sum_{ij} J_{ij}\, s_i\, s_j + \sum_i h_i\, s_i \qquad (1)$$

where $\vec{s} = (s_1, \ldots, s_n)$ is a binary vector representing the system configuration, and (h_i, J_{ij}) are parameters representing the local biases on each qubit, and the interaction strengths between them.

One of the main limitations of quantum annealers is the restriction imposed on the available interactions due to the geometrical constraints imposed by the actual physical layout of the qubits. Ideally, we would like all qubits to be able to interact with every other qubit. However, to make two qubits interact, they must be close together, and when we lay out the physical superconducting loops on a two-dimensional surface, we see that this requirement is impossible to fulfill. Hence, each qubit can only interact with a fixed number of other qubits, resulting in a sparse connectivity graph. For the D-Wave Advantage system [5] that we use (that uses the Pegasus architecture [1]), the maximum degree of this graph is 15. Even though there are techniques to get around this issue by mapping a single variable to a chain of physical qubits that has a larger effective connectivity degree, we pay a price in the size of problems that can be implemented (since the total number of qubits in the processor is fixed). Encoding other problem features like constraints further reduces the number of qubits available to encode variables. However, as we shall see next, there are other ways of exploiting the quantum mechanical features of the processor by combining them with classical approaches.

2.2 Hybrid Solvers and the Constrained Quadratic Model (CQM)

As discussed above, even though any NP-complete problem can be mapped into an instance of the Ising model, the limitations of physical quantum annealing devices restrict the type of problems that can run on them. The sparse connectivity of superconducting quantum processors requires using multiple qubits to represent a single problem variable, reducing the problem instance size that can be solved. This becomes even more problematic for constrained problems. Since the native problem solved by quantum annealers is a Quadratic Unconstrained Binary Optimization (QUBO) problem, solving constrained problems first requires transforming the constraints into penalty terms that need to be added to the objective function. Mathematically, this can be done in a way that guarantees that the optimal solution of the resulting QUBO can be used to extract the optimal feasible solution of the original problem. However, this brings new issues: (i) enforcing equality constraints as penalty terms results in an increasing connectivity requirement (usually full connectivity); (ii) enforcing inequality constraints requires using slack variables, which need to be represented by physical qubits, further reducing the effective size of the instances that can be solved; (iii) guaranteeing the faithful mapping of optimal solutions can require fine-tuning of the penalty terms coefficients; and (iv) the required strength of the penalty terms may be beyond the dynamic range of the programmable parameters of the quantum annealer, resulting in some of the smaller coefficients in the objective function being effectively neglected. Most of these issues, which can have a negative effect on the performance of the device, also apply to other platforms like a digital annealer.

One way to get around these obstacles is to abandon the goal of a faithful mapping of optimal solutions and exploit the potential computational power of the quantum annealer in a hybrid framework. The main idea of a hybrid solver is to pair the quantum processor with a classical heuristic in an iterative fashion, where this heuristic explores the search space and the output of quantum queries sent to the quantum module (QM) is used to suggest more promising areas of the solution space that the classical heuristic can further explore. A front-end takes the problem description as input and runs multiple threads of classical heuristics; the QM generates native QUBO problems that can be solved by the Quantum Processor Unit (QPU) using quantum annealing; the results are passed back to the front-end that outputs the best solution after a user-provided time limit. Figure 2 (from [2]) shows a schematic of the hybrid solver framework of D-Wave.

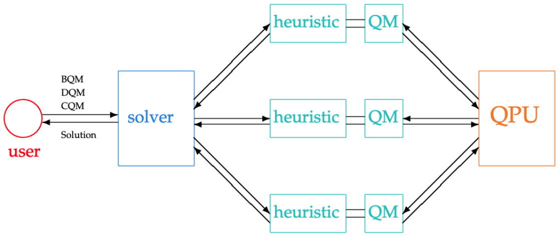

Figure 2: Schematic of the D-Wave hybrid solver framework.

D-Wave provides a Python framework to construct classical-quantum hybrid solvers. They also provide three user-ready hybrid solvers: the Binary Quadratic Model (BQM), the Discrete Quadratic Model (DQM), and the Constrained Quadratic Model (CQM). In this work, we used the CQM to solve the JO optimization problem. The main advantage of the CQM is its ease of use in programming complex constrained models. The inputs required are a description of the optimization variables (that could be binary or integer), the objective function, and the constraints. The constraints can be linear or quadratic, and the CQM provides simple tools to input any equality or inequality constraint. Algorithm 1 shows the steps required to solve a constrained quadratic problem. There are no steps required to map the problem to the underlying processor's connectivity, and there is no need to fine-tune parameters.

3 Quantum Circuit Model

Formally, Dirac notation is commonly used to describe the state of a qubit as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers called the *amplitudes* of the classical states $|0\rangle$ and $|1\rangle$, respectively. The state of the qubit is normalized, i.e. $|\alpha|^2 + |\beta|^2 = 1$. When the state $|\psi\rangle$ is measured, only one of $|0\rangle$ or $|1\rangle$ is observed, with probability $|\alpha|^2$ and $|\beta|^2$, respectively. The measurement process is *destructive*, in the sense that the state collapses to the value $|0\rangle$ or $|1\rangle$ that has been observed, losing the original amplitudes α and β.

Algorithm 1 Constrained Quadratic Model

```
Declare the optimization variables
Describe the objective function to be minimised $f(x)$
Describe the constraints in terms of the declared variables
Initialize the CQM Model
Run the hybrid CQM solver
Extract the best solution provided that satisfies the constraints
```

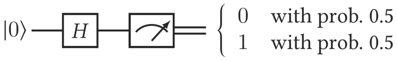

Figure 3: A simple quantum circuit. Single lines carry quantum information while double lines carry classical information.

Operations on qubits are usually represented by *gates*, similar to gates in a classical circuit. An example of a common quantum gate is the NOT gate (also called the Pauli-X gate) that is analogous to the classical NOT gate. Specifically, when we apply the NOT gate to the state $|\psi_0\rangle = \alpha|0\rangle + \beta|1\rangle$, we obtain the state $|\psi_1\rangle = \beta|0\rangle + \alpha|1\rangle$. Gates are usually represented by unitary matrices while states are represented by column vectors.[1] The matrix for the NOT gate is $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and the operation above can be written as $|\psi_1\rangle = X|\psi_0\rangle$.

Another important gate is the Walsh–Hadamard gate (H), which maps $|0\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, i.e. a superposition state with equal probability for $|0\rangle$ and $|1\rangle$, and maps $|1\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

Figure 3 shows a simple quantum circuit. Single lines carry quantum information while double lines carry classical information (typically after measurement). The circuit applies an H gate to state $|0\rangle$, producing the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Upon measurement, the result is either 0 or 1 with equal probability, and the state collapses to the observed classical bit value.

Quantum *interference* is at the core of quantum computing: by applying gates, one can cleverly bias the amplitudes of the qubits toward the desired state, thereby achieving a specific computational result.

The notion of a qubit can be extended to higher di-

[1] The ket notation $|*\rangle$ denotes column vectors, while the bra notation $\langle *|$ denotes row vectors.

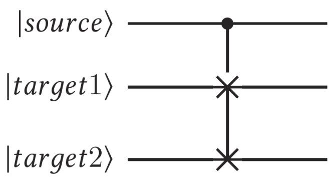

Figure 4: Example of controlled gates: the two target qubits are swapped iff the control qubit is $|1\rangle$.

mensions via a *quantum register*. A register $|\psi\rangle$ consisting of n qubits lives in a 2^n-dimensional complex Hilbert space $\mathcal{H}$. For example, $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$ with $\sum_i |\alpha_i|^2 = 1$. Basis state $|i\rangle$ denotes the n-bit binary encoding of the integer i.

We use the tensor product $\otimes$ to compose quantum systems. For instance, combining $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and $|\phi\rangle = \gamma|0\rangle + \delta|1\rangle$ yields $|\omega\rangle = |\psi\rangle \otimes |\phi\rangle = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$.

Gates can also act on multiple qubits. Figure 4 illustrates a common multi-qubit gate, the *controlled* gate. In such a gate, the operation (e.g. SWAP) is performed on the target wire(s) *iff* the control qubit is in state $|1\rangle$. Controlled operations are used to *entangle* qubits, meaning measurements on one qubit reveal information about the other, even when spatially separated.

A convenient way to describe a quantum algorithm is via a *quantum circuit*, a collection of gates applied to quantum registers. The circuit manipulates the state using superposition, entanglement, and interference to reach a desired output state that encodes the algorithm's answer. The final step is to measure the output state(s), revealing the required classical information.

Finally, the *no-cloning theorem* states that quantum bits cannot be copied. Consequently, quantum information cannot be duplicated as in classical systems, which has important implications for cryptography (preventing perfect eavesdropping) while making classical error-correction techniques unsuitable for quantum states.

References

[1] D-Wave documentation. `https://docs.dwavesys.com/docs/latest/c_gs_4.html`, 2024.

[2] Hybrid Solver for Constrained Quadratic Models [WhitePaper]. `https://www.dwavesys.com/media/rldh2ghw/14-1055a-a_hybrid_solver_for_constrained_quadratic_models.pdf`, 2021.

[3] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. S. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose. Quantum annealing with manufactured spins. *Nature*, 473(7346):194–198, 2011.

[4] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Phys. Rev. E*, 58:5355–5363, Nov 1998.

[5] The D-Wave Advantage System [WhitePaper]. `https://www.dwavesys.com/media/3xvdipcn/14-1058a-a_advantage_processor_overview.pdf`, 2022.