Benchmarking Chain Strength: An Optimal Approach for Quantum Annealing

Thinh V. Le*, Manh V. Nguyen*, Tu N. Nguyen*, Thang N. Dinh†, Ivan Djordjevic‡, and Zhi-Li Zhang§
*Department of Computer Science, Kennesaw State University, Marietta, GA 30060, USA.
†Department of Computer Science, Virginia Commonwealth University, VA 23284 USA.
‡Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721, USA.
§Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN 55455, USA.

***Abstract*—Quantum annealing (QA) is a promising optimization technique used to find global optimal solution of a combinatorial optimization problem by leveraging quantum fluctuations. In QA, the problem being solved is mapped onto the quantum processing unit (QPU) composed of qubits through a procedure called *minor-embedding*. The qubits are connected by a network of couplers, which determine the strength of the interactions between the qubits. The strength of the couplers that connect qubits within a *chain* is often referred to as the *chain strength*. The appropriate balance of chain strength is equally imperative in enabling the qubits to interact with one another in a way that is strong enough to obtain the optimal solution, but not excessively strong so as not to bias the original problem terms. To this end, we address the problem of identifying the *optimal chain strength* through the utilization of Path Integral Monte Carlo (PIMC) quantum simulation algorithm. The results indicate that our judicious choice of chain strength parameter facilitates enhancements in quantum annealer performance and solution quality, thereby paving the way for QA to compete with, or potentially outperform, classical optimization algorithms.**

***Index Terms*—Quantum annealing, chain strength, quantum simulation.**

I. Introduction

Quantum annealing (QA). Quantum mechanics is a branch of physics that characterizes the behavior of particles (e.g electron, photon, etc.). This field has paved the way for numerous cutting-edges quantum computing techniques, including cryptography [1], logistics optimisation [2], and more. In the world of quantum physics, a core principle is that systems tends to find minimum energy states. Quantum annealing is a potent optimization method that utilizes quantum physics, specifically the adiabatic theorem [3], to search for low-energy states of a combinatorial problem, thereby determining the global optimal or sub-optimal solutions. QA encompasses a wide range of real-world applications, especially solving optimization problems in healthcare [4], financing [5], drug discovery [6] and others. There are many enterprises that provide quantum annealing services, among which D-Wave standing out as a prominent company. Their most recent annealer generation contains over 5000 qubits [7], with an emphasis on the development of next-generation quantum processing units (QPU) with over 7000 qubits. As the companies continue to advance their technology, the potential for QA to revolutionize optimization grows, which bringing us closer to the quantum era.

We thank the anonymous reviewers for their suggestions and feedback. This research was in part supported by US NSF under Grants: AMPS-2229073, AMPS-2229075, CNS-2103405, CNS-1814322, and CNS-1901103. Corresponding author: Tu N. Nguyen. We release the source code at: https://github.com/NextCNS/piqmc-chainstrength.

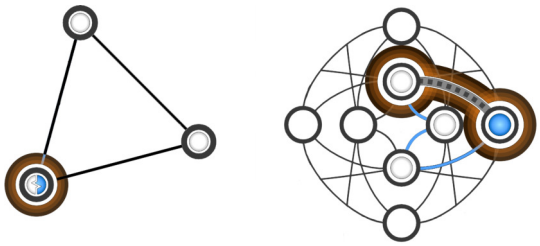


Fig. 1: A logical representation of an Ising Hamiltonian with three variables (left) is mapped onto D-Wave 2000Q-6 Chimera topology (right). A chain of qubits is connected by grey dash line. Solid blue lines represent original couplings.

Minor embedding. In order to use quantum annealer, the combinatorial objective function has to be formulated to the form of binary quadratic model (BQM) [8]. Problems in this class include quadratic unconstrained binary optimization (QUBO) and Ising model. These binary models can be represented by graphs (referred to as *logical graph*) and this graphic representation can be mapped onto physical QPU topology (i.e *physical graph*). The process of mapping logical graph onto physical graph of QPU is called *minor-embedding*, which is a critical step in solving optimization problems with QA and being a NP-hard problem itself [9]. Nodes in the logical graph represent variables (also known as *logical qubits*) with their linear coefficients are mapped to physical qubits and their external magnetic field. Meanwhile, edges that represent quadratic coefficients in BQM are mapped to couplers that connect corresponding physical qubits. Fig. 1(left) illustrates a logical graph of 3 logical variables and its mapping on D-Wave 2000Q Chimera topology [10] in Fig. 1(right). In Fig. 1(left), one of logical qubits is represented by 2 physical qubits in the QPU topology in Fig. 1(right). Since qubits within a chain represent a logical qubit, they are expected to be in the same state after annealing process. If the qubits that represent the same logical qubit do not have the identical state after annealing, the chain is classified as *broken*.

Chain strength. The QPU is composed of qubits-basic units of quantum information. The qubits are connected by a network of couplers, which determine strength of the interactions between the qubits. The strength of the couplers within a chain of qubits is often referred to as the *chain strength*. The chain strength plays an important role in quantum annealing because it determines the correlation between the qubits in the chain.

Why balancing chain strength. Determining the optimal chain strength with polynomial-time algorithms remains an open question in the literature with the following obstacles:

- **Weak chain strength.** If the chain strength is too weak, the qubits will not be able to interact strongly enough to find the optimal solution [11]. In particular, a weak chain strength will cause the chains to "break" (i.e qubits within chain are not in the same state) after annealing.
- **Excessively strong chain strength.** On the other hand, if the chain strength is too strong, the qubits within a chain are overly coupled, making negatively impact on the performance of the annealer such as slow annealing times and poor solution quality [11].

As illustrated in Fig. 1 (right), a pair of qubits within the chain have different states, which implies that chain strength is inadequate. In either case, we lost the opportunity to study the original problem and obtain the optimal solutions. Choi [12] and Fang et.al [13] put forth heuristic methodologies for determining ferromagnetic coupling within an individual chain. However, a more systematic approach is necessitated. To this end, we address the problem of identifying the *optimal chain strength* through the utilization of Path Integral Monte Carlo (PIMC) quantum simulation algorithm. We summarize key **innovations** and **contributions** of this work as follows:

- We deeply investigate the minor embedding process that encodes the chain strength and associated constraints within a chain of physical qubits. We also consider to address the challenge of the small number of qubits and sparse connectivity in near-term QPU.
- We then put forth the *first-of-its-kind* and comprehensive model for the benchmarking of chain strength in quantum annealing by employing the Sherrington-Kirkpatrick (SK) Ising model, alternatively known as a fully-connected Ising model, derived from an arbitrary Ising Hamiltonian logical graph without loss of generality and PIMC.
- Given the chain strength formulated in the prior steps, we advocate for a novel genetic-based algorithm to obtain the optimal chain strength in a pre-defined interval.
- Furthermore, this work also encompasses practical evaluations based on existing real-world D-Wave quantum computers [14] that allows a *close-to-metal* benchmarking of the chain strength in quantum annealing. The experimental results reveal that the chain strength obtained by the proposed algorithm performs better than the values determined by the existing algorithms in terms of both *total post processing time* and *ground state probability*.

Organization. In the rest of paper, we introduce preliminaries in §II. The formal definition of minor-embedding and alignment constraint is outlined in §III. §IV introduces the SK graph construction and PIMC quantum simulation, which lay a foundation for the GAC algorithm. Following that, we present the experiments results and analysis in §V. Lastly, §VI summarizes our contributions and discusses future works.

II. Preliminaries

In this section, we first present the Ising Hamiltonian and QUBO model, which encode combinatorial problems. We then discuss the QA and minor-embedding process, which are utilized to search for global minimum energy states of the Ising Hamiltonian.

A. Ising Hamiltonian and QUBO

The Ising model is originally proposed as the theoretical description of ferromagnetism, a physical phenomenon that has been mimicked to solve many optimization problems. The model describes the specific kind of magnetism that where materials, such as lodestone or iron, are able to inherently exert without the support of any electrical charge. This physical phenomenon is explained by the atomic spins take place within the material, where each constituent atom acts as an elementary electromagnet, when their associated moments aligned, a macroscopic magnetic field arises from the material. The Ising model binds the spin state of each individual atoms of the ferromagnetic object to the its total energy. In the following, we briefly describe the Ising model and its relevance to the aforementioned optimization scheme.

Consider a physical system of n atoms, the spin s_i of an atom i falls into one of two states: either spin up ($\uparrow$) or spin down ($\downarrow$), which are represented by either $+1$ or -1, respectively. The total energy of the system is modeled by the following Hamiltonian function:

$$H(s) = \sum_{i=1}^{n} h_i s_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} J_{ij} s_i s_j \tag{1}$$

where h_i models the magnetic field strength of atom i, and J_{ij} models the exchange energy between i and j. Via the principle of minimum energy, which states: *objects tends to arrange itself in order to seek the lowest energy state*, a new scheme of optimization emerges. That is, by replicating an instance of ferromagnetism such that h and J are controllable, the solution to an optimization problem can be derived via the settled lowest energy state of the system. The process of conversion is as follows. First, the variables are mapped to the spin state $\{+1, -1\}$ of the Ising Hamiltonian. Second, the coefficients, which expresses the correlation between variables and the objective, are mapped to h and J. Finally, the objective function is mapped to the total energy of system, and the optimized solution by sampling the settled spin state.

While Ising is a quadratic model that can be used to directly convert a problem into the physical configuration, the variable setting $\{+1, -1\}$ is not a natural formulation in for many of the computer science optimization problems. Instead, a much

more preferable and equivalent model is QUBO, which is written as follows:

$$f(x) = \sum_i Q_{i,i} x_i + \sum_{i,j} Q_{i,j} x_i x_j \tag{2}$$

where the variables $x_i, x_j \in \{0, 1\}$ are binary, with $Q_{i,i}$ as the linear coefficient and $Q_{i,j}$ as the quadratic coefficient in relationship to the variables. The process of embedding a QUBO model onto the physical system is straight forward by converting it into the Ising model simply with $s_i = 2x_i - 1$.

The simple reformulation techniques offered by QUBO makes it embraceable by a large variety of combinatorial problems such as the traveling salesman problem and its variants [15], portfolio optimization, integer factoring, protein folding, etc. Classically, simulated annealing is used to replicate the ferromagnetic phenomenon. However, thanks to recent advancement in the field of quantum computing, specifically to the technology of quantum annealing, the scheme of optimization via Ising model has become even more popular. In the next section we discuss the quantum annealer and its key specification with regards to optimization.

B. Quantum Annealing

Quantum annealing has been made prominent by D-Wave, who is the first company to introduce the first commercial quantum annealer on the market. The ferromagnetic phenomenon is mimicked by the D-Wave annealer. Each elementary magnet is made up of a superconducting loop whose circulating current provide the encoding of information. Such components are dubbed *qubits*, since they also exerts quantum property. Each *qubits* can be initialized as a superposition of both spin up ($\uparrow$) and spin down ($\downarrow$) states, it can also be collapsed into one of two classical state, yielding the arrangement of spin states that matches the overall system energy that complies with the Ising equation.

D-Wave's annealing. To briefly describe the mechanism of D-Wave's quantum computer, let us assume an annealer of n qubits, the Hamiltonian of the system is represented as

$$\mathcal{H}(t) = A(t)\left(\sum_i \sigma_x^i\right) + B(t)\left(\sum_{i<j} J_{i,j}\sigma_z^i\sigma_z^j + \sum_i h_i\sigma_z^i\right) \tag{3}$$

where $\sigma^i_{\{x,z\}}$ denotes the Pauli matrices that operate on qubit i. The external magnetic field h_i is applied to qubit i to influence its probability of settling at certain classical state, the strength of this force is called a *bias*. Additionally, the exchange energy $J_{i,j}$ between the qubits is also controllable with quantum entanglement, and such operation is executed via a device called *coupler*. The annealing process starts out from the initial Hamiltonian at $t = 0$ where $A(t = 0) \gg B(t = 0)$ and all qubits are in the superposition state. Via the slow physical evolution process, at $t = 1$, we reach the lowest-energy state where $A(t = 1) \ll B(t = 1)$ and each qubit ends up at a classical state that potentially encodes the optimal solution to the combinatorial problem.

Quantum Boltzmann distribution. By controlling the energy landscape of the qubits, the annealing process slowly evolves the system through different ground states of the optimization problem. However, the high quality results can only be retrieve the sample distribution are close to the quantum Boltzmann distribution, put forth by the work of Aming et.al [16], which models the probabilities of the system ending up in certain qubits arrangement based on energy of the state and the temperature of the system. For a system of n qubits, we have 2^n states space. The probability that the system is in a state with a spin configuration s_i is described by Boltzmann distribution:

$$P(s) = \frac{e^{-\beta H(s_i)}}{Z} \tag{4}$$

where $\beta = 1/k_B T$ is the inverse temperature and the partition function

$$Z = \sum_{i=1}^{2^n} e^{-\beta H(s_i)} \tag{5}$$

Before performing annealing for Hamiltonian $H(s)$, one have to perform minor-embedding $H(s)$ onto physical graph, which is introduced in the following section.

C. Minor-embedding of Ising Hamiltonian

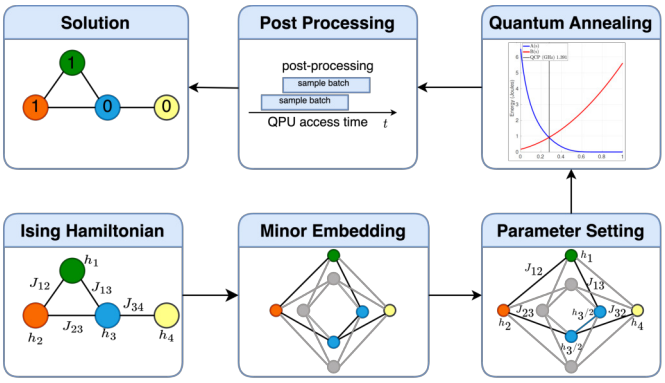


Fig. 2: Solving Ising Hamiltonian in quantum annealer.

The workflow presented in Fig. 2 dictates how to solve an Ising Hamiltonian within a QA system. As previously discussed, an Ising model can be depicted using a logical graph. This logical graph is subsequently embedded onto the physical graph. Let us denote the logical graph as H and hardware graph is G. After identifying subgraph H' on G that represents graph H, in the *parameter setting* step, the original linear and quadratic terms of H is assigned to H'. Due to the specific architecture connectivity constraints, minor-embedding algorithms may not be able to find one-to-one mapping from logical graph to physical graph. Thus, it is imperative to employ chains of physical qubits that represent logical qubits to ensure that the interactions between logical variables are preserved. The magnitude connecting the qubits within a chain is referred to as the *chain strength* or *ferromagnetic* coupling, denoted by J_f. The challenge lies in

identifying an optimal value of J_f that is strong enough to avoid biasing the original problem interaction, yet not too weak to render the qubits in chains insufficiently coupled. In the following section, we present the formal definition of minor-embedding that encodes the chain strength constraint, which establishes the foundation for our proposed approach in addressing this novel question.

III. Minor-embedding formulation

In this section, we take a step towards addressing the novel challenge of finding optimal chain strength, as mentioned in II-C. In particular, we provide a formal definition of minor-embedding and introduce a constraint that favors chain alignment after an annealing process. This formulation lays a solid foundation for our approach in the subsequent section IV of this paper.

The quantum annealer is highly susceptible to temperature fluctuations [17], [18], which limits the programmable range of the external magnetic field h_i and coupling coefficients J_{ij}. As reported in [19], the effective range of the external magnetic field h_i is $[-2, 2]$, while the range for the coupling strength J_{ij} is $[-1, 1]$. If an Ising Hamiltonian contains values outside of these ranges, the terms in the Ising Hamiltonian must be scaled using a scale factor ε to satisfy $\varepsilon|h_i| \leq 2$ and $\varepsilon|J_{ij}| \leq 1$. The chain strength, denoted as J_f, acting as a ferromagnetic coupling ($J < 0$) that promotes chain-aligned states, is also scaled alongside the Ising Hamiltonian terms. If the chain strength is excessively large compared to the Ising terms, after scaling, it will cause the Ising terms to shrink near to zero ($\varepsilon|J_{ij}| \approx 0$). Conversely, if the chain strength is too small, the ferromagnetic couplings between qubits within a chain will shrink near to zero ($J_f \approx 0$). In either case, this increases the Ising Hamiltonian's susceptibility to flux qubit noise and analog errors in the quantum annealer. As a result, a moderate chain strength is generally preferred to ensure that the ground state is chain-aligned. Fig. 3 demonstrates that an optimal chain strength can enhance the probability of achieving the ground state in QA.

Given the Ising model described by a graph $H = (V, E)$, where $V = \{v_1, v_2, \cdots, v_N\}$ and $E = \{(v_i, v_j) : v_i, v_j \in V, v_i \neq v_j,\}$ stands for the vertex and edge set of H, respectively. Let denote G is the physical graph. The problem of minor-embedding can be formally defined as:

$$H_{emb}(s) = -\varepsilon \left[\sum_{c_i \in C} \left(\sum_{m \in c_i} \hat{h}_m s_m + \sum_{m,n \in c_i} J_f s_m s_n \right) + \left(\sum_{i \in c_i, j \in c_j} \hat{J}_{i,j} s_i s_j \right) \right] \tag{6}$$

subject to:

$$s_m s_n = 1 \text{ for all } s_m, s_n \in c_i, c_i \in C \tag{7}$$

where discrete variable $s_i \in \{-1, 1\}$, $C = \{c_1, c_2, \cdots, c_n\}$ is a set of chains that represents logical qubits of H on graph G. The rescaling term ε ensures all Hamiltonian terms fall within QPU programming ranges. Whereas $\hat{h}_m$ and $\hat{J}_{i,j}$ are, respectively, the external magnetic and coupling strength on physical graph G. If a logical qubit s_i, which has the external magnetic value h_i, is represented by m number of physical qubits $q_1, \cdots, q_m$. The value h_i can be shared across m physical qubits i.e., $h_i s_i \rightarrow (h_i/m)(q_1 + q_2 + \cdots + q_m)$, reducing h_i by a factor of m. Similarly, coupling $J_{i,j}$ can also be shared. One important parameter in this Hamiltonian is the coupling between qubits within a chain J_f (i.e. the chain strength). In the following section, we provide the method to benchmark chain strength J_f that favors chain-aligned ground state in quantum annealing. In order to investigate the behaviors of Hamiltonian described in Eq. 6 within a QA system, one can employ techniques such as Simulated Annealing (SA) [20], [21] and Markov Chain Monte Carlo (MCMC) simulation methods [22], [23]. In next section of this study, we utilize the PIMC - a MCMC based algorithm, to develop a methodology for obtaining optimal value of J_f.

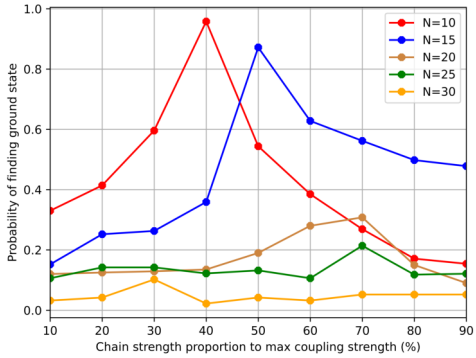


Fig. 3: Median probability of reaching to ground state with different number of instances using D-Wave 2000Q quantum annealer with annealing time = 1000μs.

IV. MCMC based quantum simulation

In this section, we commence by outlining the construction of the Sherrington-Kirkpatrick (SK) Ising model, alternatively known as a fully-connected Ising model, derived from an arbitrary Ising Hamiltonian logical graph without loss of generality, as described in IV-A. Following this, in section IV-B, we encode chain strength J_f into the ferromagnetic Ising-like term of PIMC algorithm. We then develop a Genetic Algorithm (GA)-based algorithm, named GAC, to assess chain strength within an effective coupling range utilizing PIMC as fitness function for a specific chain strength and return chain strength that yields the best performance for the original Ising Hamiltonian in section IV-C.

A. Sherrington-Kirkpatrick Ising Hamiltonian

In the SK model, each node interacts with every others as illustrated in Fig. 4(a). Similar to the SK graph that is proposed in our prior work [24], we construct an undirected weighted SK graph denoted by $G_{SK} = (V, A)$ where the set of vertices

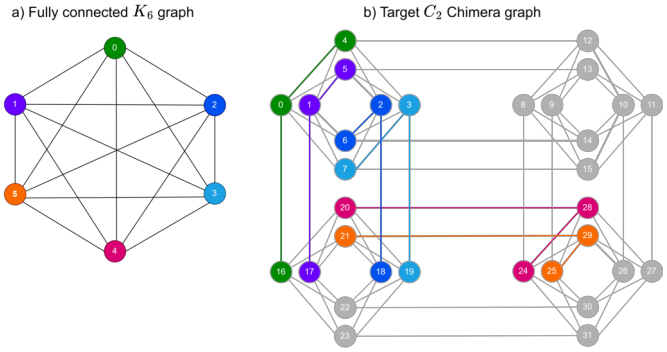


Fig. 4: (Left) An example of $N = 6$ variables embedding on a $C2$ subgraph of D-Wave 2000Q Chimera topology. Each logical qubits is encoded using a *chain* of physical qubits represented by different color (right).

$V = \{v_1, v_2, \cdots, v_{n+1}\}$ in which nodes $\{v_1, v_2, \cdots, v_n\}$ correspond to logical variables $s_1, s_2, \cdots, s_n$ in the Ising Hamiltonian. The addition vertex v_{n+1} represents the external fields h_i of each logical variables s_i. The set of arcs $A = \{(v_i, v_j) : v_i, v_j \in V, v_i \neq v_j, \}$ represents the interaction between variables s_i and s_j. A new coupling strength J'_{ij} is associated with arc $(v_i, v_j) \in A$ where $J'_{ij} = J_{ij} + J_{ji}$ if s_i, s_j interact in the original Ising Hamiltonian. For non-interact logical qubits, we add an edge with a very small weight between them (e.g. $J'_{ij} = 10^{-8} \approx 0$), which will not affect result of the simulated quantum annealing algorithms. Finally, the arc (v_i, v_{n+1}) has the coupling $J'_{i,n+1} = h_i$ represents the external field h_i. We can define the SK Ising Hamiltonian as:

$$H_{SK}(s) = -\sum_{i,j \in G_{SK}} J'_{i,j} s_i, s_j \tag{8}$$

The equivalence between finding the ground state of the Ising Hamiltonian $H(s)$ and the SK graph $H_{SK}(s)$ has been rigorously established in our previous work [24]. Inherently, due to its construction, the $H_{SK}(s)$ does not have external fields. The Hamiltonican when minor-embed G_{SK} onto the physical topology G can be expressed as:

$$H^{SK}_{emb}(s) = -\varepsilon \left[\left(\sum_{c_i \in C} \sum_{m,n \in c_i} J_f s_m s_n \right) + \left(\sum_{i \in c_i, j \in c_j} \hat{J}_{i,j} s_i s_j \right) \right] \tag{9}$$

subject to:

$$s_m s_n = 1 \text{ for all } s_m, s_n \in c_i, c_i \in C \tag{10}$$

where notations is similar to Eq. 6. By leveraging the Lagrangian penalty method [25], the constraint Eq. 10 can be integrated into the Hamiltonian in Eq. 9 as a penalty term:

$$\bar{H}^{SK}_{emb}(s) = -\varepsilon \left[\left(\sum_{c_i \in C} \sum_{m,n \in c_i} J_f (\frac{1}{2} + s_m s_n) \right) + \left(\sum_{i \in c_i, j \in c_j} \hat{J}_{i,j} s_i s_j \right) \right] \tag{11}$$

The first term of the Hamiltonian $\bar{H}^{SK}_{emb}$ corresponds to the ferromagnetic interaction among spins within the same chain, which encourages the alignment of physical qubits in an annealing process. The chain strength term J_f acts as a *scalar penalty* applied to the Hamiltonian $\bar{H}^{SK}_{emb}$, penalizing spin configurations with misaligned spins within a chain. The second term represents the interaction between spins in different chains. Altogether, Eq. 11 represents an unconstrained Hamiltonian that integrates both the alignment of physical qubits inside a chain and the interaction between separate chains according to the G_{SK} graph. The optimization objective is to identify a spin configuration that minimizes this Hamiltonian, which is equivalent to discovering the ground state of the initial problem.

In the D-Wave 2000Q-6 Chimera topology [26], each unit cell consists of a $K_{t,t}$ complete bipartite graph, with two sets of t qubits ($t = 4$ in Chimera topology), known as "shores", fully interconnected within the unit cell as illustrated in Fig. 4(b). To embed a G_{SK} graph onto the Chimera topology, we need a chain of $\lceil \frac{N}{t} \rceil + 1$ physical qubits, where t represents the shore size within each unit cell, and $\lceil . \rceil$ denotes the ceiling function. Fig 4(a) displays a fully connected K_6 source graph minor-embedded onto a C_2 Chimera graph, where each logical qubit is represented by a chain of logical qubits with a corresponding color. For example, logical qubit 0 (green) is represented by physical qubits $\{0, 4, 16\}$ with the same color. Converting a standard Ising Hamiltonian into the G_{SK} form offers the following advantages:

1) This conversion yields greater *flexibility* due to the fully connected nature of the SK graph. As a result, it can represent any Ising model, irrespective of the initial graph structure.
2) By unifying diverse problem representations into a single SK graph representation, we can effectively determine the number of *Trotter slices* [27] required for PIMC and reduce the complexity of PIMC algorithm by removing the external field terms.

Given these advantages of SK model, in the subsequent section, we employ PIMC with a modified ferromagnetic Ising-like term to simulate $\bar{H}^{SK}_{emb}$. This approach will enable us to understand how a quantum annealing system responds to a particular chain strength.

B. Path Integral Monte Carlo

In the prior section, we construct G_{SK}, which enables us to determine the number of Trotter slices for PIMC discussed in this section. Here, we reformulate the Hamiltonian 11 that is introduced in IV-A to the form of PIMC objective function,

which help us to propose an algorithm to assess chain strength in section IV-C using classical computer.

PIMC, also known as Simulated Quantum Annealing (SQA), is an classical Markov Chain Monte Carlo (MCMC) based algorithm for sampling Boltzman state of a QA system [27]. Consider a quantum Hamiltonian:

$$\hat{H} = -\sum_{i,j} J_{i,j}\sigma_i^z\sigma_j^z - \Gamma\sum_i \sigma_i^x \tag{12}$$

Here, σ_i^z, σ_i^x are Pauli matrices. The transverse field Γ controls the transition between $\uparrow$ and $\downarrow$ of each spin. To find the Boltzmann state of $\hat{H}$, one has to evaluate canonical partition function:

$$Z_\beta = tr(e^{-\hat{H}/T}) \tag{13}$$

Nonetheless, the task of evaluating the partition function through exponentiation of $e^{-\hat{H}/T}$ proves to be considerably challenging. The main idea behind PIMC is the utilization of path integral formulation, which was invented by Richard Feynman [28], and the Trotter formula [29]–[31] to approximate canonical partition function Z_β. To approximate Z_β, R. Martonak et.al [27] add an imaginary-time dimension and map quantum Hamiltonian $\hat{H}$ to a $(d+1)$ dimensional *anisotropic* classical Ising system. The approximation of Z_β is

$$Z_\beta \approx \sum_{s^1}\cdots\sum_{s^P} e^{-H_{pimc}/PT} \tag{14}$$

where

$$H_{pimc} = -\sum_{k=1}^{P}\left(\sum_{i,j\in G_{SK}} J_{i,j}s_i^k s_j^k + J^\perp \sum_{i\in G_{SK}} s_i^k s_i^{k+1}\right) \tag{15}$$

Binary variables s_i^k takes value in $\{-1,+1\}$. Parameter P, k are the number of Trotter slices and $k = \{1,2,\cdots,P\}$ being the index for extra imaginary-time dimension. $J_{i,j}$ is the original coupling strength in Eq. 9. The $J^\perp$ term can interpreted as ferromagnetic Ising-like coupling strength between adjacent Trotter replicas of the same spin along time imaginary-time dimension:

$$J^\perp = -\frac{PT}{2}\ln(\tanh\frac{\Gamma}{PT}) \tag{16}$$

A common method for performing this sampling process is to use the Metropolis algorithm, which combines both local and global moves. To be specific, for each local move, we attempt to independently flip spins at all sites in all Trotter slices. The probability of accepting new state after a local move is specified by the Metropolis acceptance rule. After the local moves, the global move is performed by flipping simultaneously all the replicas of the same site in all Trotter slices [21], [27]. Each complete updating all spins locally and globally constitutes a *sweep*. PIMC and its Metropolis implementations on classical computing systems allows the acquisition of insights on QA quantum phenomena. These insights may contribute to a deeper comprehension and analysis of QA such as the evaluation of D-Wave machines quantum annealer performance.

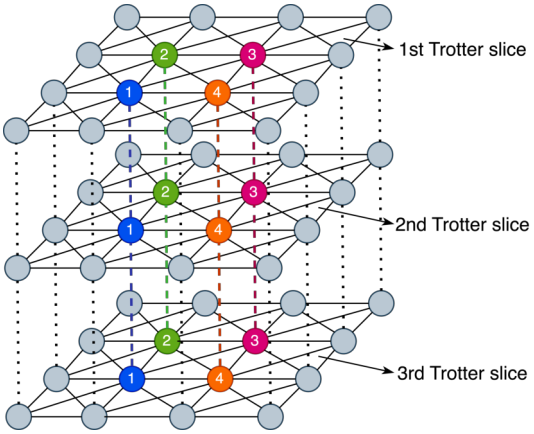


Fig. 5: A demonstration of 3 Trotter slice, same spins in all trotter slices are connect via a $J^\perp$ ferromagnetic Ising-like represent a chain of physical qubits as in Fig. 4(b)

Utilizing PIMC, we formulate Hamiltonian $\bar{H}_{emb}^{SK}$ to the form of H_{pimc} to benchmark a given chain strength as follows:

$$\bar{H}_{pimc} = -\sum_{k=1}^{P}\left(\sum_{i,j\in G_{SK}} \hat{J}_{i,j}s_i s_j + J^\perp \sum_{i\in G_{SK}} (\frac{1}{2} + s_i^k s_i^{k+1})\right) \tag{17}$$

where notation is the same with Eq. 15 and

$$J^\perp = -\frac{PT}{2}\ln(\tanh\frac{\Gamma}{J_f PT}) \tag{18}$$

Fig. 5 depicts a classical *anisotropic* Ising system with 3 Trotter slices. In this system, replicas of a spin in all Trotter slices form a *"chain"*, which corresponds to chains of physical qubits in Fig. 4(b). These replicas are coupled together using ferromagnetic Ising-like $J^\perp$. By transforming original Ising Hamiltonian to SK Ising model, we can determine the number of Trotter slices, which is equal to the number of physical qubits that needed to map a vertex in G_{SK}, (i.e $P = \lceil\frac{N}{t}\rceil + 1$). The second term in $\bar{H}_{pimc}$ imposes a penalty on misalignment within chain along the imaginary-time dimension. Furthermore, we also incorporate chain strength parameter J_f into the $J^\perp$ term, with J_f being proportional to $J^\perp$.

The PIMC algorithm starts with random initialization in all Trotter slices and independent among slices to obtain an initial spin configuration s^0. As mentioned, we need to update spins locally and globally. For a local update of each spin, we then derive the energy different for flipping this spin ($s_i^k = -s_i^k$) as follow:

$$\Delta_{local}E = 2\sum_{i,j\in G_{SK}} \hat{J}_{ij}s_i^k s_j^k + 2J^\perp(s^{k-1}s_i^k + s_i^k s_i^{k+1}) \tag{19}$$

The local move accept new state with the probability $min(1, e^{-\Delta_{local}E/PT})$ and the energy different of global move can be calculated as:

$$\Delta_{global}E = 2\sum_{k=1}^{P}\sum_{i,j\in G_{SK}} \hat{J}_{ij}s_i^k s_j^k \tag{20}$$

with the new state acceptance ratio is $min(1, e^{-\Delta_{global}E/PT})$.

If J_f is too small, the Trotter slices become practically decoupled (i.e. independent of each other). Even if they are initialized with the same configuration, they will eventually become uncorrelated after evolution in the decoupled regime for too long. On the other hand, an excessively large J_f value can also be detrimental as it prevents proper correlations the slices between slices. Therefore, in the next section, we propose a GA algorithm to benchmark chain strengths within programmable range of QPU leveraging PIMC algorithm with modification that is proposed in this section.

C. Benchmarking Chain Strength Algorithm

In this section, we describe the adoption of Genetic Algorithm (GA) to find an "optimal" chain strength, named *GAC*. In our proposed GAC algorithm, the population is updated based on *fitness score*. There are three main stages in GAC algorithm: 1) Population initialization 2) Fitness calculation and 3) Population update. The pseudocode of our proposed GA algorithm in outlined in algorithm 1.

Algorithm 1 GAC Algorithm

Input: an interval $[0, \alpha]$, step γ, number of populations $\mathcal{D}$
Output: Chain strength J_f

1: Initialize population $\mathcal{P}$ uniformly in range $[0, \alpha]$, step γ.
2: Let $\mathcal{C} \leftarrow \emptyset$ be the set of chain strength candidates
3: Initial $p_a = 0$, $p_b = \alpha$
4: **while** *generation* $\leq \mathcal{D}$ or $p_b - p_a > \gamma$ **do**
5: Let F be a set of fitness score for generation $\mathcal{P}$
6: Let F_{min} be the lowest fitness score in F
7: Update p_a, p_b be the first chain strength and the last chain strength in $\mathcal{P}$ that has $F_a, F_b = F_{min}$
8: Append p_b to $\mathcal{C}$
9: Update chain strengths in next population $\mathcal{P}$ in range [p_a,p_b] with step γ.
10: **end while**
11: J_f = max($\mathcal{C}$)
12: return J_f

Initialization (Lines 1-3, Alg. 1): Initializing chain strength is an important step in our proposed algorithm. There are several strategies to initial population, but we can categorize them as: *random initialization* and *heuristic initialization*. In spire by [32], we initial chain strength population in the range $[0, \alpha]$ where:

$$\alpha = \sqrt{\delta^2} \tag{21}$$

with $\delta^2 = \frac{1}{(N-1)} \sum_{i<j} J_{i,j}^2$. The step between chain strength in each population is:

$$\gamma = \frac{\sqrt{\sum_{i<j} J_{i,j}^2}}{N} \tag{22}$$

The population initialization step serves as the foundation for subsequent iterative improvement. We can calculate population size denoted as $\mathcal{V} = \alpha/\gamma$. Following this, we initiate the set of chain strength candidates, denoted as $\mathcal{C}$, which is employed to retain the best chain strength observed after each generation.

Lemma 1. *The value of γ is always smaller than α with $N > 1$*

Proof. Square both α and γ:

$$\alpha^2 = \frac{1}{N-1} \sum_{i<j} J_{i,j}^2$$

$$\gamma^2 = \frac{1}{N^2} \sum_{i<j} J_{i,j}^2$$

Showing that $\gamma < \alpha$ is equivalent to $\gamma^2 < \alpha^2$. We have:

$$\frac{\alpha^2}{\gamma^2} = \frac{N^2}{N-1} > 1 \quad (\forall N > 1)$$

Thus, $\alpha > \gamma$, which complete the proof. □

Iterative fitness score calculation (Line 5, Alg. 1): In section IV-B, we propose a PIMC formulation which integrate chain strength into ferromagnetic Ising-like $J^{\perp}$ term. The fitness function takes chain strength J_f as the input and produces the suitability of J_f as output. For each chain strength in population $\mathcal{P}$, the fitness score is defined as:

$$\min_k \left(\sum_{i,j\ G_{SK}} J_{i,j} s_i s_j + J^{\perp} \sum_{i \in G_{SK}} (\frac{1}{2} + s_i^k s_i^{k+1}) \right) \tag{23}$$

where $k = \{1, 2, \cdots, P\}$. Put simply, the fitness score of a chain strength J_f is equal to minimum energy of Trotter slices obtained after performing PIMC simulation with that J_f.

Population update (Lines 6-9, Alg. 1): In order to prevent getting stuck at local minimum, after each generation, we refine the search space interval after each generation by utilizing the smallest and largest elements in $\mathcal{P}$, denoted as p_a and p_b respectively, which has the lowest fitness scores. The offspring in the next generation are updated based on the new interval and step γ. The algorithm stops once it reach the maximum of generations or $p_b - p_a < \gamma$. Lastly, the proposed chain strength is determined by the largest element within the candidate set $\mathcal{C}$.

D. Complexity Analysis

The overall complexity of GAC will depend on the number of generations $\mathcal{D}$, the complexity of the fitness function, and convergence of p_a, p_b. The time complexity of initializing the chain strength population is $\mathcal{O}(\mathcal{V})$, where $\mathcal{V}$ represents the number of chain strengths in each generation. The time complexity of evaluating the fitness of each chain strength depends on the complexity of the fitness function (i.e PIMC algorithm). Assuming the fitness function has a time complexity of $\mathcal{O}(\mathcal{F})$, the complexity of this step is $\mathcal{O}(\mathcal{V} * \mathcal{F})$. Assuming the fitness function has a time complexity of $\mathcal{O}(\mathcal{F})$, the complexity of this step is $\mathcal{O}(\mathcal{V} * \mathcal{F})$. The actual number of iterations will depend on the input parameters and the problem itself. To keep the analysis simple, let us assume the primary loop has an average number of iterations, denoted as $\mathcal{N}_{avg}$. The overall time complexity of GAC can be writen as $\mathcal{O}(\mathcal{N}_{avg} * \mathcal{V} * \mathcal{F})$.

V. Experiment

In the prior section, we present the application of PIMC simulation to assess the effectiveness of a chain strength in a QA system and GAC algorithm, a GA-based algorithm, to find for optimal chain strength, as defined in IV-C. In this section, we perform experiments to evaluate the chain strengths recommended by our algorithm and ones suggested by the D-Wave chain strength calculation default function, *uniform_torque_compensation* [33]. Subsequently, we analyze the improvement in performance of quantum annealer using the chain strength recommended by our proposed algorithms.

Problem scales. We evaluate our proposed method using varying number of instances $N = \{5, 10, 15, \cdots, 40, 45\}$. With the number of instances determined, we can easily calculate several other parameters, such as the number of Trotter slices P, which define the size of lattice along the imaginary-time dimension, and the temperature T. As previously mentioned, the number of Trotter slices P is given by $\lceil N/t \rceil + 1$. It is important to note that in the PIMC original work [27], the authors recommend to keep the product $PT = 1$. As a result, we can compute the temperature T as $1/P$. By varying the value of N, we are able to naturally update the value of P and T accordingly. The coupling strength J_{ij} between instances in the original Ising Hamiltonian is randomly choose from the interval $[-1, 1]$. Next, we uniformly generate chain strengths within the range, and steps specified in the previous section IV-C. As for the PIMC algorithm implementation, we adapt the open-source PIMC [1] as required. Table. I summaries parameters that we use in GAC and PIMC implementation.

Parameter	Description	Value
N	Number of instances	$\{5, 10 \cdots, 45\}$
P	Number of Trotter slices	$\lceil N/t \rceil + 1$
Γ_0	Initial value of the transverse field	0.1
Γ_T	Final value of the transverse field	10^{-8}
T	Temperature	$1/P$
$\mathcal{D}$	Number of generation in GAC	10
mcsteps	Number of sweeps	5

TABLE I: GAC and PIMC parameters summary

Algorithm. We compare the effectiveness of our chain strength to the chain strengths derived from the default function *uniform_torque_compensation* (UTC) of D-Wave, which employs the *root mean square* to calculate chain strength as:

$$J_f = \rho \sqrt{\frac{\sum_{i,j} J_{ij}^2}{n}} \tag{24}$$

Here, ρ is a prefactor for scaling with a default value of 1.414, J_{ij} is the coupling strength and n is the number of interactions in the Ising model.

Environment. In this study, we employ D-Wave 2000Q-6 quantum annealer, which features more than 2000 physical qubits. To minor-embed Ising Hamiltonian onto physical topology, we make use of minor-miner [2] function provided by D-Wave to embed our problem of interest. Fig. 6 illustrates an Ising problem is minor-embeded onto D-Wave 2000Q-6 Chimera topology. After annealing process, the D-Wave sampler can highlight chains that is break.

[1] https://github.com/therooler/piqmc

[2] https://docs.ocean.dwavesys.com/projects/minorminer/en/latest/

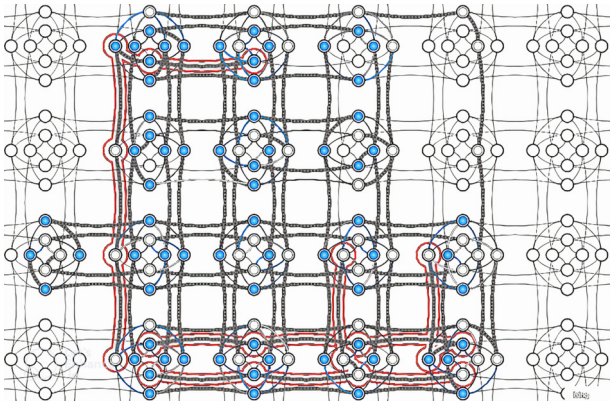


Fig. 6: A genuine minor-embedding of an Ising Hamiltonian onto D-Wave 2000Q-6 quantum annealer. The dashed gray lines represent physical qubits connected to form a chain, while the solid red lines indicate broken chains.

Evaluation Metrics. In our study, we analyze the effectiveness of the chain strength by:

- *Chain break fraction:* This crucial performance indicator represents the ratio of the number of broken chains to the total number of chains.
- *Ground state probability:* This metric quantifies the number of ground state samples found in the sample set (i.e how frequently ground state is sampled).
- *Total post processing time:* QPU handles samples in batches. Each batch is processed and sent through the post-processing solver. The goal of post-processing is to get a set of samples that align with a target Boltzmann distribution. The cumulative time spend on post-processing samples is called *total post processing time*.

A. Chain strength and ground state probability

Fig. 7 presents a comparison of chain strengths derived from our proposed algorithm, GAC, and UTC. Upon our experiments, it is evident that GAC chain strengths are marginally smaller than those computed by UTC with both values exhibiting an increase as the number of instances grows. Notably, even though GAC chain strengths being somewhat lower, their performance in term of *ground state probability* is better to that of UTC. In particular, the disparity in ground state probability between GAC and UTC is substantial with $N = \{10, 15, 20\}$. However, as the number of instance N increases, the discrepancy becomes less prominent. In the following section, we will also explore varying D-Wave quantum annealer parameter settings to assess the impact of chain strength on QPU performance.

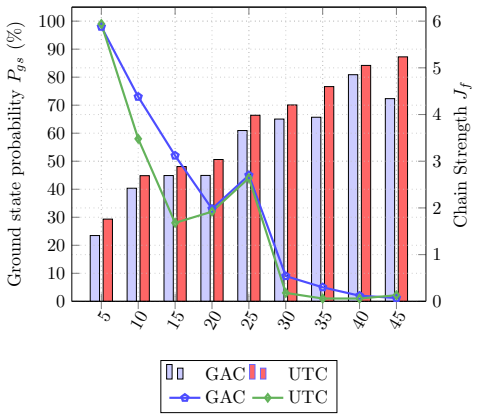


Fig. 7: Comparison between chain strengths derived from GAC and UTC. The ground state probability corresponds to each algorithm are also reported with *annealing_time* = 1000, *num_reads* = 100.

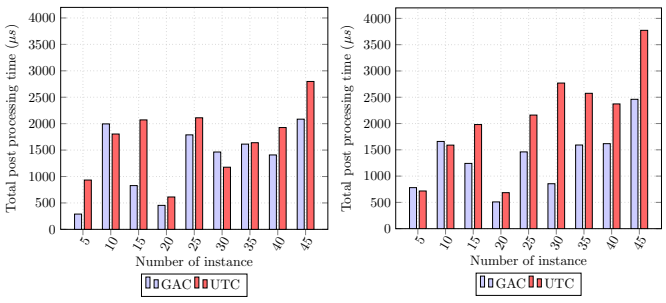
B. Benchmark on D-Wave quantum annealer performance

In order to study the performance of D-Wave quantum annealer with different chain strengths, we measure the metrics that is mentioned above: *chain break fraction*, *total post processing time*, *ground state probability* in D-Wave Chimera QPU's topology. To gain the best insight of the quantum annealer system, we vary the two important parameters: *annealing_time* and *num_reads*.

- *num_reads:* This parameter represents the number of states (output solutions) to be retrieved from the solver. It must be a positive integer within the range specified by the solver property. Intuitively, increasing the number of reads may yield better solutions.
- *annealing_time.* In addition to chain strength and *num_reads*, the annealing time is another crucial parameter. It determines the duration, in microseconds, of the quantum annealing process for each read. The time resolution is $0.01\mu s$ for Advantage systems and $0.02\mu s$ for D-Wave 2000Q systems, as detailed in [34].

Increasing both *annealing_time* and *num_reads* may increase the solution quality of a combinatorial problem. However, due to a fixed time budget for problem submission to the QPU, the optimal combination of *annealing_time* and *num_reads* to achieve the best solution depends on the specific problem.

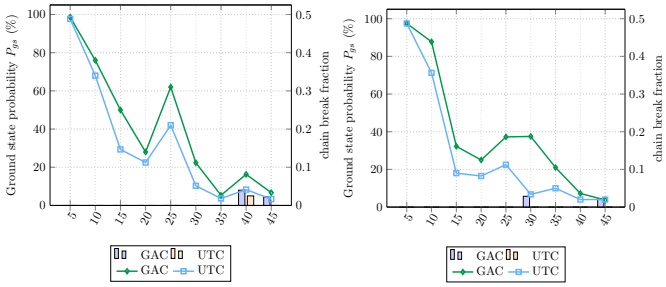
As it can be noticed from Fig. 8, in general, when we increase *annealing_time* by $500\mu s$ and *num_reads* by 100, the *total post processing time* corresponds to each number of instances also increase. However, in majority of the cases, our proposed chain strength exhibits a more efficient post-processing time performance in comparison to the values suggested by UTC with both settings. The performance gap between GAC and UTC also becomes more significant as the number of instances increases when we alter the parameter settings from Fig. 8(a) to Fig. 8(b). Although our proposed algorithm in IV-C recommends chain strength values slightly smaller than those produce by UTC, we are able to achieve better performance in terms of post-processing time compared to UTC. In addition, the GAC algorithm's improvement in post-processing time performance can provide practical advantages in real-world applications, such as reduced computational costs and shorter solution times. Consequently, it may enables us to tackle larger, further pushing the boundaries of QA.



(a) *annealing_time*= $1500\mu s$ and *num_reads*= 300

(b) *annealing_time* = $2000\mu s$ and *num_reads*= 400

Fig. 8: D-Wave 2000Q-6's performance comparison between GAC and UTC in terms of median *total post processing time* with varying number of instance and solver parameters (*annealing_time* and *num_reads*)



(a) *annealing_time*= $1500\mu s$ and *num_reads*= 300

(b) *annealing_time* = $2000\mu s$ and *num_reads*= 400

Fig. 9: D-Wave 2000Q-6's performance comparison between GAC and UTC in terms of *ground state probability* and *chain break fraction* with varying number of instance and solver parameters (*annealing_time* and *num_reads*)

Fig. 9(a) and Fig. 9(b) display the average *chain break fraction* and *ground state probability* for both algorithms under different parameter settings. As it can be seen that for $N < 25$, the *chain break fraction* performance of GAC and UTC in both settings remain 0. With higher number of instances, chain breakage presents with both algorithm. *chain break fraction* performance corresponds to GAC is marginally higher compared to UTC. Despite the *chain break fraction* performance of UTC is approximately 0 across parameter settings. By setting the chain strength significantly larger than the Ising Hamiltonian couplings, one can manipulate this ratio to nearly 0. This strategy ensures the robustness of the quantum computations by keeping the quantum chains unbroken. Nevertheless, it may result in a trade-off between the QPU performance and the ratio.

Regarding *ground state probability*, similar to 7, it decrease as we increase the number of instances. However, increasing *annealing_time* and *num_reads* from Fig. 9(a) to Fig. 9(b) setting results in smaller ground state probability for both GAC and UTC. Nevertheless, in all settings, GAC's chain show that it can give better *ground state probability* performance compared to its counterpart. In summary, although UTC performs slightly better with *chain break fraction* in comparison with our proposed chain strength, our proposed chain strength improve the *ground state probability* and *post-processing time*.

VI. Conclusion and Future work

This work presents a systematic approach for determining the optimal chain strength in quantum annealing. To achieve this, we have introduced a formal formulation of minor-embedding that encodes chain alignment constraints. In the next step, we have constructed the SK model from the original Ising Hamiltonian and investigated the behavior of the formulated Ising Hamiltonian in the QA system using the PIMC quantum simulation algorithm. We then evaluated the performance of the D-Wave 2000Q annealer using our proposed chain strengths. Experimental results demonstrate that although our suggested chain strength is slightly lower compared to the referenced chain strength, it successfully improves the performance of the QPU.

This work not only contributes to addressing the open question of determining the optimal chain strength in quantum annealing, but also helps enhance the performance and reliability of the quantum annealer solutions. In future, the initial range and GAC algorithm can be further improved to obtain better chain strength quality.

References

[1] J.-P. Aumasson, "The impact of quantum computing on cryptography," *Computer Fraud & Security*, vol. 2017, no. 6, pp. 8–11, 2017.

[2] A. C. Sparavigna, "Quantum computing logistics," *Available at SSRN 4249265*, 2022.

[3] T. Albash and D. A. Lidar, "Adiabatic quantum computation," *Reviews of Modern Physics*, vol. 90, no. 1, p. 015002, 2018.

[4] J. Davids, N. Lidströmer, and H. Ashrafian, "Artificial intelligence in medicine using quantum computing in the future of healthcare," in *Artificial Intelligence in Medicine*. Springer, 2022, pp. 423–446.

[5] R. Orús, S. Mugel, and E. Lizaso, "Quantum computing for finance: Overview and prospects," *Reviews in Physics*, vol. 4, p. 100028, 2019.

[6] Y. Cao, J. Romero, and A. Aspuru-Guzik, "Potential of quantum computing for drug discovery," *IBM Journal of Research and Development*, vol. 62, no. 6, pp. 6–1, 2018.

[7] D-Wave System Inc., "D-wave solver docs." [Online]. Available: https://rb.gy/9tq0k

[8] ——, "D-wave solver docs." [Online]. Available: https://docs.ocean.dwavesys.com/en/stable/concepts/bqm.html

[9] V. Choi, "Minor-embedding in adiabatic quantum computation: Ii. minor-universal graph design," *Quantum Information Processing*, vol. 10, no. 3, pp. 343–353, 2011.

[10] S. Zbinden, A. Bärtschi, H. Djidjev, and S. Eidenbenz, "Embedding algorithms for quantum annealers with chimera and pegasus connection topologies," in *High Performance Computing: 35th International Conference, ISC High Performance 2020, Frankfurt/Main, Germany, June 22–25, 2020, Proceedings*. Springer, 2020, pp. 187–206.

[11] "D-wave solver docs." [Online]. Available: https://www.dwavesys.com/media/qvbjrzgg/guide-2.pdf

[12] V. Choi, "Minor-embedding in adiabatic quantum computation: I. the parameter setting problem," *Quantum Information Processing*, vol. 7, no. 5, pp. 193–209, Oct 2008. [Online]. Available: https://doi.org/10.1007/s11128-008-0082-9

[13] Y.-L. Fang and P. Warburton, "Minimizing minor embedding energy: an application in quantum annealing," *Quantum Information Processing*, vol. 19, no. 7, pp. 1–29, 2020.

[14] "D-wave solver docs." [Online]. Available: https://www.dwavesys.com/solutions-and-products/systems/

[15] T. V. Le, M. V. Nguyen, S. Khandavilli, T. N. Dinh, and T. N. Nguyen, "Quantum annealing approach for selective traveling salesman problem," in *IEEE International Conference on Communications*, 2023.

[16] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, "Quantum boltzmann machine," *Physical Review X*, vol. 8, no. 2, may 2018. [Online]. Available: https://doi.org/10.11032Fphysrevx.8.021050

[17] V. Martin-Mayor and I. Hen, "Unraveling quantum annealers using classical hardness," *Scientific reports*, vol. 5, no. 1, pp. 1–9, 2015.

[18] W. Vinci, T. Albash, and D. A. Lidar, "Nested quantum annealing correction," *npj Quantum Information*, vol. 2, no. 1, pp. 1–6, 2016.

[19] D-Wave System Inc., "D-wave solver docs." [Online]. Available: https://docs.dwavesys.com/docs/latest/c_solver_properties.html#property-h-range

[20] E. Aarts, J. Korst, and W. Michiels, "Simulated annealing," *Search methodologies: introductory tutorials in optimization and decision support techniques*, pp. 187–210, 2005.

[21] Y. Wang, S. Wu, and J. Zou, "Quantum annealing with markov chain monte carlo simulations and d-wave quantum computers," *Statistical Science*, pp. 362–398, 2016.

[22] T. Okuyama, M. Hayashi, and M. Yamaoka, "An ising computer based on simulated quantum annealing by path integral monte carlo method," in *2017 IEEE international conference on rebooting computing (ICRC)*. IEEE, 2017, pp. 1–6.

[23] Y. Wang, "Quantum monte carlo simulation," 2011.

[24] P. Thai, M. T. Thai, T. Vu, and T. N. Dinh, "Fasthare: Fast hamiltonian reduction for large-scale quantum annealing," in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2022, pp. 114–124.

[25] F. Glover, G. Kochenberger, and Y. Du, "A tutorial on formulating and using qubo models," *arXiv preprint arXiv:1811.11538*, 2018.

[26] G. Chapuis, H. Djidjev, G. Hahn, and G. Rizk, "Finding maximum cliques on the d-wave quantum annealer," *Journal of Signal Processing Systems*, vol. 91, pp. 363–377, 2019.

[27] R. Martoňák, G. E. Santoro, and E. Tosatti, "Quantum annealing by the path-integral monte carlo method: The two-dimensional random ising model," *Physical Review B*, vol. 66, no. 9, p. 094203, 2002.

[28] R. P. Feynman, A. R. Hibbs, and D. F. Styer, *Quantum mechanics and path integrals*. Courier Corporation, 2010.

[29] T. Kato and K. Masuda, "Trotter's product formula for nonlinear semigroups generated by the subdifferentials of convex functionals," *Journal of the Mathematical Society of Japan*, vol. 30, no. 1, pp. 169–178, 1978.

[30] M. Suzuki, "Generalized trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems," *Communications in Mathematical Physics*, vol. 51, no. 2, pp. 183–190, 1976.

[31] H. F. Trotter, "On the product of semi-groups of operators," *Proceedings of the American Mathematical Society*, vol. 10, no. 4, pp. 545–551, 1959.

[32] D. Venturelli, S. Mandrà, S. Knysh, B. O'Gorman, R. Biswas, and V. Smelyanskiy, "Quantum optimization of fully connected spin glasses," *Physical Review X*, vol. 5, no. 3, p. 031040, 2015.

[33] D-Wave System Inc., "D-wave solver docs." [Online]. Available: https://docs.ocean.dwavesys.com/projects/system/en/latest/reference/generated/dwave.embedding.chain_strength.uniform_torque_compensation.html

[34] ——, "D-wave solver docs." [Online]. Available: https://docs.dwavesys.com/docs/latest/c_solver_parameters.html#annealing-time