PART

13

Adiabatic Quantum Computing

Until now, we have worked exclusively with quantum algorithms that can be written as a succession of unitary gates taken from a universal gate set. However, the gate-based model of quantum computing is far from being the unique way to think about quantum algorithms. This chapter introduces a different paradigm of quantum computing, called **Adiabatic Quantum Computing (AQC)** or **Quantum Annealing**[1]. Instead of using a discrete succession of gates, AQC relies on the slow analogue evolution of a physical system, with the objective of solving an optimization problem (see Figure 1).

While AQC is equivalent to the circuit model (up to a polynomial overhead), it has unlocked a different way of thinking about quantum algorithms, can serve as a heuristic to tackle NP-Hard problems, and has a very different experimental implementation with its own advantages and drawbacks.

This chapter begins with a description of the ground state problem—the optimization problem solved by quantum annealing—and its relation to the class of NP-Hard problems known as QUBO. We then examine the adiabatic algorithm and the adiabatic theorem that underlies it. Finally, we delve into a crucial component of this algorithm: the choice of the adiabatic path.

[1] *Some authors consider quantum annealing as the noisy and more realistic version of the ideal AQC model (with no adiabaticity or universality constraint). However, the difference is not rigorously defined in the literature and we will consider them as synonyms in this book (see History and Further Reading for more information about the usage of those terms)*

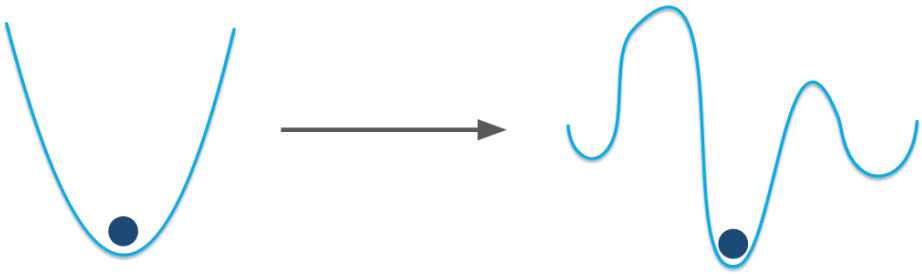

Figure 1. Schematic illustration of adiabatic quantum computing: by starting from the solution of a simple optimization problem (left) and slowly changing it to a complicated one (right), we are guaranteed by the adiabatic theorem to stay in the minimum during the whole evolution

1 The ground state problem

The objective of quantum annealing is to solve the **ground-state problem**, that can be stated as follow:

Definition 1.1 (Ground-state problem)**.** For an input Hamiltonian H, find an eigenstate $|\psi_0\rangle$ of H with the smallest eigenvalue E_0. The state $|\psi_0\rangle$ is called the **ground state** of H and is the solution to following optimization problem:

$$|\psi_0\rangle = \arg\min_{|\psi\rangle} \langle\psi|H|\psi\rangle \tag{1.1}$$

A simple proof of Eq. (1.1)—called the **variational principle**—is proposed in Exercise 1.1. Solving the ground state problem is of general interest in both the physics and

optimization communities. In physics, simulating a quantum system in equilibrium often amounts to finding the ground state of a Hamiltonian modelling this system. Therefore, quantum annealing has natural applications in quantum chemistry and condensed matter physics for instance. Maybe more surprisingly, Eq. (1.1) can also represent a large class of purely classical optimization problems. To see this, consider the following class of Hamiltonians, called **Ising Hamiltonians**:

$$H = \sum_{i,j} J_{ij}\sigma_i^z\sigma_j^z + \sum_i h_i\sigma_i^z \tag{1.2}$$

where σ_i^z is the Pauli-Z matrix applied on qubit i and J_{ij}, h_i, are real coefficients. Since H is diagonal in the computational basis, its ground state is an element of the computational basis. Noting $|\psi\rangle = |q_1...q_n\rangle$, we can rewrite Eq. (1.1) as:

$$\langle\psi|H|\psi\rangle = \sum_{i,j} J_{ij} \langle q_1...q_n|\sigma_i^z\sigma_j^z|q_1...q_n\rangle + \sum_i h_i \langle q_1...q_n|\sigma_i^z|q_1...q_n\rangle \tag{1.3}$$

$$= \sum_{i,j} J_{ij}(-1)^{q_i}(-1)^{q_j} + \sum_i h_i(-1)^{q_i} \tag{1.4}$$

$$= \sum_{i,j} J_{ij}s_is_j + \sum_i h_is_i \tag{1.5}$$

where we defined $s_i = (-1)^{q_i} \in \{-1, 1\}$ in the last line.

The problem of minimizing $f(s) = \sum_{i,j} J_{ij}s_is_j + \sum_i h_is_i$ belongs to a class of problems known as Quadratic Unconstrained Binary Optimization (QUBO) in computer science. More precisely, a QUBO problem is defined as the minimization of

$$f(x) = \sum_{i,j} Q_{ij}x_ix_j \tag{1.6}$$

where each $x_i \in \{0, 1\}$ instead of $\{-1, 1\}$. The equivalence between Ising and QUBO problems can be established by the change of variable $s_i \to 2x_i - 1$ in Eq. (1.5).

QUBO problems are known to be in the complexity class NP-HARD, and are therefore likely not solvable efficiently in general, even by a quantum computer (the implications for quantum annealing will be discussed in the next section). Due to the equivalence between all the NP-HARD problems, many famous optimization problems can be formulated as QUBO (see Box 1 and Exercise 1.2). Quantum annealing has therefore been studied in the context of combinatorial optimization (e.g. travelling salesman, nurse scheduling), machine learning (e.g. ensemble methods, clustering), finance (e.g. portfolio optimization), particle physics (e.g. tracking) and many other fields.

Box 1: MaxCut

A typical example of problem that can be framed as a ground state problem is MaxCut. Considering a graph with weighted edges, we define a cut as a partition of the edges into two regions. The maximum cut is the cut that maximizes the total weight of the edges connecting the two regions (see Figure 2). If we call J the adjacency matrix of the graph, i.e. J_{ij} is the weight of the edge connecting the nodes i and j, we can define the following Ising Hamiltonian:

$$H = \sum_{ij} J_{ij}\sigma_i^z\sigma_j^z \tag{1.7}$$

to which corresponds an energy function $E(s) = \sum_{ij} J_{ij}s_is_j$. Notice that every choice of s defines a partition of the nodes: those for which $s_i = 1$ and those for which $s_i = -1$. Minimizing E would lead to $s_i \neq s_j$ (i.e. $s_is_j = -1$) for high values of J_{ij} and $s_i = s_j$ for low values of J_{ij}, thereby solving the MaxCut problem.
While MaxCut is NP-HARD—meaning that it is very likely not solvable efficiently in general, even by a quantum computer—not all instances are necessarily hard. The hardness of a given instance (or family of instances) often depends on the properties of the associated graphs. For instance, MaxCut on planar graph can be solved in polynomial times. For this reason, the ability of a quantum annealer to represent a large variety of graph (which depends on the connectivity of the qubits) is crucial in the quest for a quantum advantage.

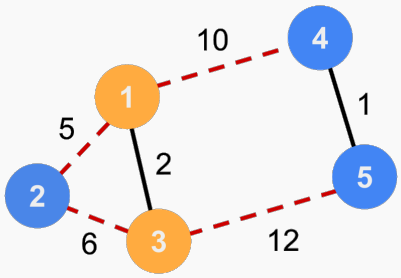

Figure 2. Example of graph to which we apply MaxCut. The maximum cut is represented in dashed red line and separate two regions: the blue nodes $\{2, 4, 5\}$ and the orange nodes $\{1, 3\}$. The total energy for this cut is $E = 1 + 2 - 10 - 12 - 5 - 6 = -30$

Exercise 1.1 (Variational Principle). Show that the ground state $|\psi_0\rangle$ of a Hamiltonian H is the solution to the optimization problem

$$\min_{|\psi\rangle} \langle\psi|H|\psi\rangle$$

(Hint: start by showing $\langle\psi|H|\psi\rangle \geq E_0$ for all $|\psi\rangle$ and $\langle\psi_0|H|\psi_0\rangle = E_0$)

Exercise 1.2 (Max-2-SAT). Max-2-SAT is another typical example of NP-Hard problem. A satisfiability problem (SAT) is a decision problem that consists in determining if for a given Boolean formula on N variables, there exists an assignment of those variables that makes the formula true. Considering the propositions that can be written as a series of ANDs ($\wedge$) between ORs ($\vee$) of 2 variables, such as

$$(x_1 \vee x_2) \wedge (\overline{x_2} \vee x_3) \wedge (x_1 \vee \overline{x_3}),$$

where $\overline{x_i}$ is the negation of x_i, Max-2-SAT consists in finding the maximum number of OR clauses that can be satisfied simultaneously. In the example above, all the OR clauses can be satisfied (take $x_1 = x_2 = x_3 = 1$ for instance), but in

$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_2}),$$

only three OR clauses can be satisfied simultaneously (e.g. the first three with $x_1 = x_2 = 1$).
Can you formulate Max-2-SAT as a QUBO problem? *(Hint: you can use the fact that* $x_1 \vee x_2 = \overline{\overline{x_1} \wedge \overline{x_2}}$*)*

2 The adiabatic algorithm

To solve the ground state problem, adiabatic quantum computing works by taking advantage of an important result discovered in the early days of quantum mechanics, the **adiabatic theorem**. The theorem uses the notion of **gap** of a Hamiltonian, which is defined as the energy difference between the first excited state and the ground state:

$$\Delta = E_1 - E_0 \tag{2.1}$$

Moreover, a Hamiltonian is said to be **gapped** if $\Delta \neq 0$. We are now ready to state the theorem[2]:

[2] *Numerous statements of the adiabatic theorem can be found in the literature, with more or less precise analysis of the approximation error. A review of those different theorems can be found in [1].*

Theorem 2.1 (Adiabatic Theorem)**.** Let $H(t)$ be a smooth family of Hamiltonians parametrized by $t \in [0, T]$, such that $H(0) = H_I$ (initial Hamiltonian), $H(T) = H_P$ (problem Hamiltonian) and $H(t)$ is gapped for all t. If we initialize a system in the ground state $|\psi(0)\rangle$ of H_I and let it evolve with $H(t)$ for a time T following the time-dependent Schrodinger's equation

$$H(t)\,|\psi(t)\rangle = -i\frac{d}{dt}\,|\psi(t)\rangle \tag{2.2}$$

then $|\psi(T)\rangle$ will be the ground state of H_P for T sufficiently large. More precisely, T is required to grow as

$$T \gg \frac{1}{\Delta_{\min}^2} \tag{2.3}$$

where $\Delta_{\min}$ is the minimum gap taken along the path (see Figure 3).

The proof of the adiabatic theorem is slightly technical and out of scope for this introduction. The interested reader can find a self-contained proof in [2].

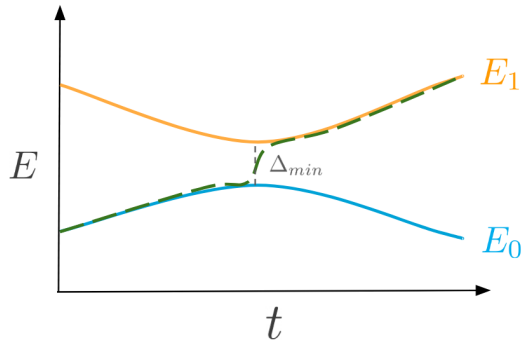

Figure 3. Ground-state and excited state energies of $H(t)$ along a trajectory between H_I and H_P. The minimum gap $\Delta_{\min}$ occurs in the middle of the path and a jump to the excited state (as illustrated by the dashed green line) is likely to occur at this point if $\Delta_{\min}$ is too small and the speed too fast.

In practice, let's imagine that we want to find the ground state of an Ising Hamiltonian $H_P = \sum_{i,j} J_{ij}\sigma_i^z\sigma_j^z + \sum_i h_i\sigma_i^z$. The adiabatic theorem gives rise to the following algorithm:

1. Initialize the system in the ground state of a simple Hamiltonian H_I. For instance, we can take $H_I = -\sum_i \sigma_i^x$, whose ground state $|+\rangle^{\otimes n}$ can be easily prepared.

2. Find a continuous path—also called **adiabatic path** or **annealing schedule**—between H_I and H_P. A common choice is the linear interpolation

$$H(t) = \frac{t}{T}H_P + \left(1 - \frac{t}{T}\right) H_I \tag{2.4}$$

   but more advanced schedules can be considered in order to maximize the minimum gap along the path (and therefore minimize the total time, as of Eq. (2.3)). We will discuss the problem of finding an optimal schedule in Section 3.

3. Measure the final state $|\psi(T)\rangle$ in the σ^z-basis. If T is large enough, it should be the ground state of H_P, i.e. a computational basis state encoding the solution to the associated QUBO problem.

The reader may now wonder if we have finally found a procedure to solve any NP-HARD problem efficiently on a quantum computer. Unfortunately, as discussed in Chapter 6.6, it is unlikely that quantum computers have this ability (i.e. that BQP contains NP), even for quantum annealers. The reason is that hard instances of QUBO with no particular structure, such as random instances, have a gap that vanishes exponentially with the problem size, requiring an increasingly large annealing time to find the solution. The hope is that AQC could give a better (or faster) result on some particular instances than most classical heuristics.

Exercise 2.1 (Adiabatic Grover)**.** To see the equivalence between the gate and the adiabatic model, one can construct an equivalent of Grover's algorithm (Chapter 6) for adiabatic quantum computers. Let $H_P = I - |\boldsymbol{m}\rangle\langle\boldsymbol{m}|$ where $\boldsymbol{m}$ is the n-bit string that we are searching for—H_P is the equivalent of the oracle in Grover's original algorithm—and $H_I = I - |+\rangle\langle+|$ where $|+\rangle = |+\rangle^{\otimes n}$.

(a) Show that the $H(s) = sH_P + (1-s)H_I$ has a 2-dimensional invariant subspace, spanned by $|\boldsymbol{m}\rangle$ and $|\boldsymbol{m}^\perp\rangle = \frac{1}{\sqrt{N-1}}\sum_{i\in\{0,1\}^n\setminus\{\boldsymbol{m}\}} |i\rangle$.

(b) Write the restriction of $H(s)$ in this invariant subspace as a 2×2-matrix and

prove that its eigenvalues are given by $\frac{1}{2}(1-\Delta(s))$ and $\frac{1}{2}(1+\Delta(s))$, where

$$\Delta(s) = \sqrt{\frac{N + 4(N-1)s(s-1)}{N}} \tag{2.5}$$

(c) Show that all the other eigenvalues of $H(s)$ are equal to 1 *(Hint: you can write* $H(s) = M(s) + \left(I - |\boldsymbol{m}\rangle\langle\boldsymbol{m}| - |\boldsymbol{m}^\perp\rangle\langle\boldsymbol{m}^\perp|\right)$ *where* $M(s)$ *belongs to the invariant subspace defined above and* $\left(I - |\boldsymbol{m}\rangle\langle\boldsymbol{m}| - |\boldsymbol{m}^\perp\rangle\langle\boldsymbol{m}^\perp|\right)$ *is a projector on the orthogonal subspace)*

(d) Deduce that $\Delta(s)$ is the gap of $H(s)$, and show that its minimum is

$$\Delta\left(s = \frac{1}{2}\right) = \frac{1}{\sqrt{N}} \tag{2.6}$$

Since the annealing time scales as $1/\Delta_{\min}^2$, we have found an algorithm that run in $O(N)$. We will see in Section 3 that with a better adiabatic path, we can recover the $O(\sqrt{N})$ of Grover's algorithm.

3 Choosing the adiabatic path

The equivalence between the gate model and the adiabatic model requires access to a large range of adiabatic paths between H_I and H_P, and not simply the linear schedule with constant speed defined in Eq. 2.4. Several strategies are possible to improve the adiabatic path.

3.1 Adaptive annealing speed

The first strategy is to keep a linear schedule but vary the speed during the evolution:

$$H(t) = s(t)H_P + (1 - s(t))H_I \tag{3.1}$$

where $s(0) = 0$ and $s(T) = 1$. If the gap is known, one can then apply the adiabatic condition Eq. (2.3) locally and try to have a low speed when the gap is small and a high speed when the gap is large. This strategy allows for instance to recover the quadratic speed-up in Grover's algorithm (Exercise 2.1), by using a schedule defined by the condition

$$\frac{ds}{dt} = \Delta(t)^2 \tag{3.2}$$

instead of a constant speed.

For more general problems where the gap is unknown, optimal control methods can be applied to find the optimal path. Those methods often consist in discretizing the trajectory and applying either gradient-based (e.g. GRAPE) or gradient-free (e.g. reinforcement learning) methods to find the trajectory that minimizes the final energy.

3.2 Navigator terms and nonstoquastic Hamiltonians

A second approach to improving the annealing schedule is to add one or several other terms to the path, sometimes called **navigator Hamiltonians**. For instance, we could have

$$H(t) = s(t)H_P + \lambda(t)H_{\text{nav}} + (1 - s(t))H_I \tag{3.3}$$

where $s(0) = 0, s(T) = 1$ and $\lambda(0) = \lambda(T) = 0$. This intermediate Hamiltonian can allow to explore more regions of the Hamiltonian space, possibly avoiding exponentially vanishing gaps.

The intermediate Hamiltonian can also be designed to avoid classical simulability. Indeed, to simulate quantum annealing classically, a common method is the path-integral Quantum Monte Carlo (QMC) algorithm. However, it can be shown that QMC only works well when there is **no sign problem** in the Hamiltonian. A Hamiltonian with no sign problem, also called **stoquastic** Hamiltonian, is defined as having only nonpositive elements in the off-diagonal (with respect the computational basis):

$$\langle i|H|j\rangle \leq 0 \quad \text{if } i \neq j \tag{3.4}$$

If H_P is an Ising Hamiltonian and $H_I = -\sum_i \sigma_i^x$, their linear combination will be stoquastic. To create a path that is hardest to simulate, one could add a term of the form:

$$H_{\text{nav}} = \sum_{ij} \sigma_i^x \sigma_j^x \tag{3.5}$$

to the annealing schedule. While there are some evidence that nonstoquastic navigators can improve the performance of quantum annealers, it has also been shown that adiabatic quantum computing with no-sign problem is not classically simulable in general. The exact role of stoquasticity in quantum annealing is still an open problem.

4 Quantum annealers in practice

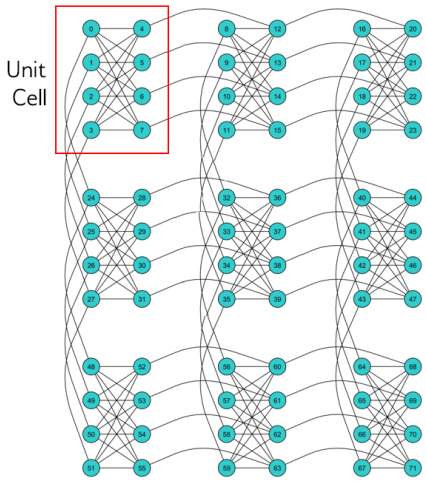

Figure 4. Example of **Chimera Graph**, representing the qubit connectivity on a D-Wave device [3]. A procedure known as **minor embedding** can be used to map an arbitrary QUBO instance on N variables to a Chimera graph on M qubits, where $M \geq N$, and often $M = O(N^2)$

When proposed, quantum annealing was seen as a promising avenue towards a first experimental quantum computer with an advantage over classical machines: it does not require the implementation of fast quantum gates, can tolerate a moderate level of noise, and the first numerical investigations on optimization problems suggested an improvement compared to classical heuristics. With this in mind, the Canadian company D-Wave started building quantum annealers as early as 1999 and has reached 5,000 qubits in 2020. However, the quest for a quantum advantage in optimization problems is still ongoing as of

2020. One of the reasons is that practical implementations can be far from the ideal model discussed previously. In particular:

1. The qubit connectivity is often limited, meaning that not every Ising Hamiltonian can be implemented in the device without a large (polynomial) overhead (see Figure 4). Since QUBO problems on simple graphs are often the easiest to solve numerically, it can hinders the power of the quantum device.

2. Not every adiabatic paths can be implemented in practice. Historically, D-Wave has been implementing linear interpolation between the two Hamiltonians, but there is no guarantee that the optimal path lies in this space.

3. Large levels of noise can be detrimental to the performance of the device.

In light of those limitations, what role could quantum annealing take in the future of quantum computing? While we do not know yet if quantum annealers will ever be useful for solving generic optimization problems that arise in real-life applications, their potential for quantum simulation seem more amenable in the near-term. Recent experiments have for instance demonstrated a million-fold speed-up compared to classical algorithms for the simulation of frustrated magnets. Understanding the power of adiabatic quantum computing is still an open but fascinating question for both theorists and experimentalists, that has the potential to illuminate important aspects of quantum information.

Summary of Chapter 13: Adiabatic Quantum Computing

- **Ground state problem**: finding the ground state of a Hamiltonian comes down to solving the optimization problem

$$\min_{|\psi\rangle} \langle\psi|H|\psi\rangle$$

- **Ising Hamiltonians** are of the form

$$H = \sum_{i,j} J_{ij}\sigma_i^z\sigma_j^z + \sum_i h_i\sigma_i^z.$$

  Finding their ground state is equivalent to solving a QUBO problem:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \sum_{i,j} Q_{ij}x_ix_j$$

  where $x_i \in \{0, 1\}$

- **Adiabatic theorem**: a system will remain in the ground state during a Hamiltonian evolution as long as the time scales as $1/\Delta_{\min}^2$, where $\Delta_{\min}$ is the minimum gap along the trajectory.

- **The adiabatic path** has a drastic effect on the solution found by the algorithm. The optimal schedule can be found theoretically or with optimal control methods.

- **Practical quantum annealers** are often limited by their connectivity, the lack of freedom in the choice of an adiabatic path and noise.

History and further reading

The adiabatic theorem dates back to the early days of quantum mechanics. It was first introduced by Born and Fock in 1928 [4] and put into a more rigorous mathematical ground by Kato in 1950 [5]. Adiabatic quantum computing was established independently by two teams, with two different perspectives and two different names. Kadowaki and Nishimori created quantum annealing in 1998 by analogy with simulated annealing, with the objective of finding the ground state of an Ising model by exploiting quantum fluctuations [6]. The idea of using analogue evolution as a quantum computing model originates from Farhi and Gutmann who found a first analogue version of Grover's algorithm [7]. They generalized their approach and linked it to the adiabatic theorem in 2000 [8]. Several proofs of the equivalence between the two models were established in the following years [9–13]. In 2009, it was shown that AQC fails for random NP-Hard instances due to an exponentially vanishing gap [14].

Numerical and theoretical evidence that having a nonstoquastic navigator Hamiltonian in the adiabatic paths can improve a schedule were put forward in [15–17]. In 2020, Hastings showed that AQC with no sign-problem is not classically simulable in general [18]. A first experimental demonstration of quantum advantage for condensed matter simulations has been demonstrated on 1,440 qubits of a D-Wave quantum annealer in 2021, outperforming classical simulation techniques by six orders of magnitude [19].

Exercise 2.1 is inspired by [1, 20, 21].

References

[1] Tameem Albash and Daniel A Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1):015002, 2018.

[2] Satoshi Morita and Hidetoshi Nishimori. Mathematical foundation of quantum annealing. *Journal of Mathematical Physics*, 49(12):125210, 2008.

[3] What is the chimera topology? https://support.dwavesys.com/hc/en-us/articles/360003695354-What-Is-the-Chimera-Topology-, 2018. Accessed: 2021-02-25.

[4] Max Born and Vladimir Fock. Beweis des adiabatensatzes. *Zeitschrift für Physik*, 51(3-4): 165–180, 1928.

[5] Tosio Kato. On the adiabatic theorem of quantum mechanics. *Journal of the Physical Society of Japan*, 5(6):435–439, 1950.

[6] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Physical Review E*, 58(5):5355, 1998.

[7] Edward Farhi and Sam Gutmann. Analog analogue of a digital quantum computation. *Physical Review A*, 57(4):2403, 1998.

[8] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*, 2000.

[9] M Stewart Siu. From quantum circuits to adiabatic algorithms. *Physical Review A*, 71(6): 062314, 2005.

[10] Dorit Aharonov, Wim Van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM review*, 50(4):755–787, 2008.

[11] Roberto Oliveira and Barbara M Terhal. The complexity of quantum spin systems on a two-dimensional square lattice. *arXiv preprint quant-ph/0504050*, 2005.

[12] Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006.

[13] Ari Mizel, Daniel A Lidar, and Morgan Mitchell. Simple proof of equivalence between adiabatic quantum computation and the circuit model. *Physical review letters*, 99(7): 070502, 2007.

[14] Boris Altshuler, Hari Krovi, and Jérémie Roland. Anderson localization makes adiabatic quantum optimization fail. *Proceedings of the National Academy of Sciences*, 107(28): 12446–12450, 2010.

[15] Hidetoshi Nishimori and Kabuki Takada. Exponential enhancement of the efficiency of quantum annealing by non-stoquastic hamiltonians. *Frontiers in ICT*, 4:2, 2017.

[16] Shunji Matsuura, Takeshi Yamazaki, Valentin Senicourt, Lee Huntington, and Arman Zaribafiyan. Vanqver: the variational and adiabatically navigated quantum eigensolver. *New Journal of Physics*, 22(5):053023, 2020.

[17] Layla Hormozi, Ethan W Brown, Giuseppe Carleo, and Matthias Troyer. Nonstoquastic hamiltonians and quantum annealing of an ising spin glass. *Physical review B*, 95(18): 184416, 2017.

[18] Matthew B Hastings. The power of adiabatic quantum computation with no sign problem. *arXiv preprint arXiv:2005.03791*, 2020.

[19] Andrew D King, Jack Raymond, Trevor Lanting, Sergei V Isakov, Masoud Mohseni, Gabriel Poulin-Lamarre, Sara Ejtemaee, William Bernoudy, Isil Ozfidan, Anatoly Yu Smirnov, et al. Scaling advantage over path-integral monte carlo in quantum simulation of geometrically frustrated magnets. *Nature Communications*, 12(1):1–6, 2021.

[20] Wim Van Dam, Michele Mosca, and Umesh Vazirani. How powerful is adiabatic quantum computation? In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 279–287. IEEE, 2001.

[21] Jérémie Roland and Nicolas J Cerf. Quantum search by local adiabatic evolution. *Physical Review A*, 65(4):042308, 2002.