12-2020

Benchmarks and Controls for Optimization with Quantum Annealing

Erica Kelley Grant
University of Tennessee, Knoxville, egrant8@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Erica Kelley Grant entitled "Benchmarks and Controls for Optimization with Quantum Annealing." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Energy Science and Engineering.

Travis S. Humble, Major Professor

We have read this dissertation and recommend its acceptance:

Travis S. Humble, Peter L. Fuhr, Bruce J. MacLennan, James P. Ostrowski

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Benchmarks and Controls for Optimization with Quantum Annealing

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Erica Kelley Grant

December 2020

Dedication

I would like to dedicate this dissertation to those in my life who recognized my potential and inspired me to pursue my science career. My partner, Jamie Stump, kept me accountable and gave me the love and support I needed to thrive as a student and researcher. My mother and father, Beverly and Timothy Grant, instilled me with work ethic and fostered my curiosity and entrepreneurial drive. My high school AP physics teacher, Jeremy Watts, was the first teacher/mentor to recognize my potential in academia. I would particularly like to thank my advisor, Travis Humble. I began studying with Travis in 2014 when interning at Oak Ridge National Laboratory. Because he spent the time with me that summer to answer my many questions about quantum computing, I left that internship feeling empowered and knowing that I wanted to be a quantum computing physicist/engineer. As an advisor, Travis was patient, spent the time to understand my research interests, sharpened my writing and time management skills, strategically gave me the tools that I needed to build my dissertation, and set me up for success after graduate school by enabling me to pursue entrepreneurial endeavors. I would not be where I am today without the impact that each of these people had on my life.

Acknowledgments

A large portion of the research background in Chapter 2 of this dissertation was written as part of a Oxford University Press' Oxford Research Encyclopedias article entitled "Adiabatic Quantum Computing and Quantum Annealing," authored by Erica Grant and Travis Humble. Permission was given to use the manuscript as part of this dissertation. There are two other research articles whose content is included in this dissertation that have been submitted for publishing, including "Benchmarking Quantum Annealing Controls with Portfolio Optimization" and "Benchmarking Embedded Chain Breaking in Quantum Annealing."

Abstract

Quantum annealing (QA) is a metaheuristic specialized for solving optimization problems which uses principles of adiabatic quantum computing, namely the adiabatic theorem. Some devices implement QA using quantum mechanical phenomena. These QA devices do not perfectly adhere to the adiabatic theorem because they are subject to thermal and magnetic noise. Thus, QA devices return statistical solutions with some probability of success where this probability is affected by the level of noise of the system. As these devices improve, it is believed that they will become less noisy and more accurate. However, some tuning strategies may further improve that probability of finding the correct solution and reduce the effects of noise on solution outcome. In this dissertation, these tuning strategies are explored in depth to determine the effect of preprocessing, annealing, and post-processing controls on performance. In particular, these tuning strategies were applied to a real-world NP (nondeterministic polynomial time)-hard optimization problem and portfolio optimization. Although the performance improved very little from tuning the spin reversal transforms, anneal time, and embedding, the results revealed that reverse annealing controls improved the probability of success by an order of magnitude over forward annealing alone. The chain strength experiments revealed that increasing the strength of the intra-chain coupling improves the probability of success until the intra-chain coupling strengths begin to overpower the inter-chain couplings. By taking a closer look at each physical qubit in the embedded chains, the probability for each qubit to be faulty was visualized and was used to develop a post-processing strategy that outperformed the standard, which chooses a logical qubit value from a broken chain. The results of these findings provide a guide for researchers to find the optimal set of controls for their unique real-world optimization problem to determine whether QA provides some benefit over classical computing, lay the

groundwork for developing new tuning strategies that could further improve performance, and characterize the current hardware for benchmarking future generations of QA hardware.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivations to Study Quantum Computation

The past 70 years hold a rich history of computational advancement. The first digital vacuum tube computer with prestored programs was the Electronic Numerical Integrator and Computer (ENIAC), developed in 1943. It was Turing-complete and used by the US Army's Ballistic Research Laboratory to calculate artillery firing tables during World War II [169]. The first transistor-based computing machine, IBM 608, was made commercially available in 1957 with $3,000$ germanium transistors [143]. After the invention of the transistor came the integrated circuit, which made circuit design manufacturing cheaper, faster, and more scalable. As of June 2020, the most powerful supercomputer is Supercomputer Fugaku from Japan, which has a theoretical peak performance of $513,854.7$ TFlop/s and $7,299,072$ cores [1]. Today, high-performance computers are rigorously used for diverse applications, including weather and climate modeling, nuclear and materials research, data analytics, and machine learning. These applications are pushing humanity toward a future with precision medicine, natural disaster prediction and preparation, and a deeper understanding of physical systems [11]. However, some key limitations in classical computational methods could slow down or even halt progress.

1.1.1 Limit to Moore's Law

The transistor is the building block of computational technology as we know it. A transistor is essentially a switch that either blocks or allows the flow of electrons by raising and lowering a potential barrier. The on/off electron flow registers as bits of information that can be set to either 0 or 1. Transistors create a network of logic gates that, when combined, do operations such as addition, subtraction, and multiplication [85]. The ability to perform these basic operations allows computers to do almost any calculation as the number of transistors increases, which includes everything from running a simple calculator program to graphically intensive video games and scientific simulations.

To reach today's power of computation, manufactures have used the scalability of integrated circuits to continuously develop better chips with more transistors. Moore's Law is a trend that shows the number of transistors on an integrated circuit doubling every two years. It was first introduced by Gordon Moore, cofounder of Intel, in a 1965 paper in which he predicted that this doubling would continue for at least a decade. Moore's Law is now the guide for computational advancement in the semiconductor industry and has held true for nearly five decades [117]. Today, manufacturers have scaled production-ready transistors down to 6 nm by developing high-purity silicon wafers in cleanrooms and using more precise production methods such as chemical etching, particle beams, and molecular optics [83].

The power of a computer directly depends on the number of transistors it uses. Metal-oxide-semiconductor field effect transistors (MOSFETs) and complementary metal-oxide semiconductors (CMOSs) are the most efficient and predominantly used transistor technologies to date. Unfortunately, as manufacturers continue to shrink the size of the CMOS and MOSFET, quantum mechanical effects become more dominant, which can introduce a significant amount of error. The state-of-the-art transistor is currently a 5 nm transistor developed by Samsung for mass production in 2019 [Mass], and Samsung has begun working on a $3nm$ transistor in 2020 [Schilling]. However, these dimensions make the transistor susceptible to quantum tunneling phenomena—the finite probability that electrons will tunnel through the dielectric barrier when the transistor is supposed to block the flow of electrons. The probability for quantum tunneling increases as the dielectric

barrier decreases in dimension, which causes substantial current flow leakage and increases redundancy incorporation [130]. Thus, computational performance and clock speed no longer scales with transistor size [43]. The future of Moore's Law is a widely debated topic. Quantum tunneling effects are already being observed, and MOSFETs and CMOSs will likely soon hit a hard limit as they reach around 5 nm in length [139]. Moore's Law may be extended by the development of new transistor models, such as the tunnel field effect transistor, which uses quantum tunneling by sustaining a high-energy barrier and fluctuating the current to manipulate the probability for electrons to tunnel through the dielectric barrier [10]. However, this type of technological advancement only slightly stretches the limit to Moore's Law. Transistors are currently 100 atoms in diameter. Even if transistors could overcome quantum phenomena and remain a reliable switch at these dimensions, a hard limit exists: the size of an atom. If single-atom transistors are developed and follow the current rate of advancement, the projected limit to Moore's Law is still expected to be reached between 2025 and 2030 [97]. Throughout the history of computation, the technologies used to process a logical bit of information have all reached their saturation points. Interestingly, Moore's Law has held true to this point for transistors because manufacturers have invested in new and better implementations. Although the integrated circuit has advanced computation more than any other device, it is also reaching saturation. Unconventional computing methods have the potential to continue the trend of increased computational power over time without the use of the transistor.

1.1.2 Thermodynamic Limits

As the size of the transistor continues to shrink, heat becomes a growing complication. Conventional computation is irreversible, meaning that after a gate performs an operation, the information is erased and the gate resets. The third law of thermodynamics dictates that entropy must always increase in a system [152]. Therefore, resetting a bit releases information in the form of heat proportional to the transistor's ambient temperature. The formal definition is given by Landauer's principle, which states that "any logically irreversible manipulation of information, such as the erasure of a bit or the merging of two computation paths, must be accompanied by a corresponding entropy increase in noninformation-bearing

degrees of freedom of the information-processing apparatus or its environment" [13]. To prevent a chip from melting, computational devices need to dissipate this heat quickly. Supercomputers implement large cooling systems, but some thermodynamic limits affect the effectiveness of cooling. The maximum number of bits of information that can be stored in a system is given by

$$N_b its = \frac{E}{k_b T ln(2)} + log_2 \sum_i e^{\frac{-E_i}{k_b T}} \tag{1.1}$$

where k_b is the Boltzmann constant, T is the temperature of the system, and E_i is the energy of each bit [152]. Thus, to increase the number of transistors on an integrated circuit, the temperature must continue to decrease. Manufacturers have already seen complications with high power density. In 2004, Intel cancelled its CPU project with 4-GHz clock frequencies because cooling requirements became too complex and inefficient [59]. Transistors currently operate at $1,000$ mV and the operation limit for transistors is approximately 200 mV. Lowering the voltage to these levels would decrease the heat production to some extent but would slow down the rate of computation [106]. Cooling apparatuses also have room for improvement. However, heat removal has a quantum limit given by

$$Q_{removal} \leq (\frac{\pi k_b^2 T^2)}{3\hbar} \tag{1.2}$$

where $\bar{h}$ is the reduced Plank's constant ($\frac{h}{2\pi}$) [135]. Computational methods must address these fundamental thermodynamic limits continue to increase computational power.

1.1.3 Energy Consumption Limitations

The internet has allowed people all over the world to communicate, educate, and share. This ability has sparked the globalization of information and a boom in computational devices. Shared information on this scale requires mass data storage. Every social media post, web search, news article, e-book, music file, and website requires data to be processed and stored on servers. Large collections of these servers are stored in data centers and are often referred to as "the cloud." The development of the cloud has increase data storage

efficiencies tremendously. Data centers in the United States account for 42.2% of US energy consumption, which is projected to begin increasing at an exponential rate as data centers reach 80–90% efficiency [106].

As the transistor shrinks in size, the amount of electricity, including voltage and current, required to power the transistor decreases. In 1974, Robert Dennard theorized that as the number of MOSFETs on an integrated circuit doubles, the power usage remains relatively constant [47]. This effect was dubbed "the Dennard scaling theory," and it held true for more than 30 years. However, in 2006, the Dennard scaling theory broke down and the power density began increasing with transistor count, primarily because of current leakage and an increased probability for thermal runaway—a positive feedback loop of current flow and temperature [19].

Since 2010, computational energy consumption has increased exponentially. This increase is the combined result of Moore's Law, the breakdown of the Dennard scaling theory, data storage, and the increased number of computational devices. Efficiencies are continuously researched and implemented to slow this trend. However, efficiency can only go so far. The Landauer limit is the maximally efficient use of irreversible computation at $3 \times 10^{-21)} \frac{J}{bit}$. Per the US Energy Information Administration, the world's net electricity production is only projected to increase 69% by 2040 to account for population inflation [162]. To power transistor-based computers, the world's energy production will need to increase at an exponential rate, which is currently unsustainable.

1.1.4 Programming Limitations

The limitations to conventional hardware and to the programs implemented on the hardware must be considered. Whether programs can become significantly more optimized is highly debated. The following information is based on the current understanding of conventional computation.

Computational problems are categorized in classes according to the level of difficulty. The difficulty is solely based on the best-known algorithms for solving these problems. It determines the number of transistors, links of communication between transistors, amount of time, and any other resources needed to solve the problem.

The Turing machine is a model used to define the required resources. The model manipulates bits that are divided into cells on an infinitely long piece of tape. The head reads and writes the bit and the state onto a register and then moves to a cell on the left or right depending on the set of rules in place. The Turing machine used in conventional computing is deterministic, meaning the rules are fixed.

The Turing machine is used to determine the complexity of a problem. For a deterministic Turing machine, the time taken to solve a problem is a function of the total number of steps before the head halts and gives an answer [146]. Thus, a problem is solvable in some time T if a Turing machine that can solve it exists. Problems are categorized based on their determined complexity. Complexity classes are determined based on the type of computational problem, the Turing machine used to solve the problem, and the resources available (time and computational space available). In conventional computing, four primary complexity classes exist: P (polynomial time), NP (nondeterministic polynomial time), NP-complete, and NP-hard.

- The P class contains a set of problems that can be solved in polynomial time, which means that for N bits of information, the amount of time taken to solve a problem scales N^k where k is a positive integer and thus can be solved efficiently by a deterministic Turing machine [156]. Notably, not all P-class problems are computationally fast. For a case in which k is large, the run time would still be long. However, in general, P-class problems are the simplest in terms of complexity. The P class contains problems such as multiplication and finding the greatest common divisor.

- The NP class contains problems that can be solved by a nondeterministic Turing machine in polynomial time. Thus, conventional computers cannot solve this class of problems efficiently because time would instead scale k^N. However, the problem solution can be verified in polynomial time. The P class is contained in the NP class because a nondeterministic Turing machine can solve P class problems efficiently. Finding the coprime factors of some integer h is a problem that is widely believed to be in the NP class because no known algorithm can solve it deterministically in

polynomial time. However, the coprime factors can be checked easily with a simple multiplication operation.

- The NP-complete class contains a set of problems to which all other NP problems can be reduced in polynomial time. In other words, all NP problems can be written as NP-complete problems [123].

- The NP-hard class contains problems that are NP-complete or harder. Thus, NP-complete problems are in both the NP and NP-hard classes. NP-hard problems outside of NP-complete can no longer be checked in polynomial time. An example of an NP-hard problem that is not NP-complete is the halting problem, which determines whether a program will run forever given the input [123].

These classes are formed from the current construction of algorithms. When considering the possibilities of computation, the P class only encompasses a small area of problems. To realize the full potential of computation, other Turing machines should be considered. Some researchers believe that P = NP, and the algorithms that solve NP problems on a deterministic Turing machine in polynomial time have not been discovered yet. This belief is widely considered to not be the case, and this paper assumes that P $\neq$ NP [60].

As these limitations become more concrete, new computing methods have garnered research and public attention. For problems that fall outside of P class, there is interest in unconventional forms of computation that may provide some computational resource advantage over classical computation. This dissertation is dedicated to analyzing quantum annealing as an unconventional method of computation for solving a real-world NP-HARD optimization problem. In particular, this quantum annealing device is benchmarked for probability of success and probability of errors against a brute force solver. Tuning strategies are implemented and studied for their ability improve the performance of the quantum annealer. These tuning controls are one part of working toward a quantum computer with resource benefits over classical computational methods.

1.2 Introduction to Adiabatic Quantum Computing and Quantum Annealing

Adiabatic quantum computing (AQC) is a model of computation that uses quantum mechanical processes operating under adiabatic conditions. As a form of universal quantum computation, AQC employs the principles of superposition, tunneling, and entanglement that manifest in quantum physical systems. The AQC model of quantum computing is distinguished by the use of dynamical evolution that is slow with respect to the time and energy scales of the underlying physical systems. This adiabatic condition enforces the promise that the quantum computational state will remain well-defined and controllable, thus enabling the development of new algorithmic approaches.

Several notable algorithms developed within the AQC model include methods for solving unstructured search and combinatorial optimization problems. In an idealized setting, the asymptotic complexity analyses of these algorithms indicate computational speed-ups may be possible relative to state-of-the-art conventional methods. However, the presence of non-ideal conditions, including non-adiabatic dynamics, residual thermal excitations, and physical noise, complicate the assessment of the potential computational performance. A relaxation of the adiabatic condition is captured in the complementary computational heuristic of quantum annealing (QA), which accommodates physical systems operating at finite temperature and in open environments. Although QA provides an accurate model for the behavior of actual quantum physical systems, the possibility of non-adiabatic effects obscures a clear separation with conventional computing complexity.

A series of technological advances in the control of quantum physical systems has enabled experimental AQC and QA. Prominent examples include demonstrations using superconducting electronics, which encode quantum information in the magnetic flux induced by a very weak current operating at cryogenic temperatures. A family of devices developed specifically for unconstrained optimization problems has been applied to solve problems in specific domains, including logistics, finance, materials science, machine learning, and numerical analysis. An accompanying infrastructure was also developed to support these experimental demonstrations and enable access to a broad community of users. Although

AQC is most commonly applied in superconducting technologies, alternative approaches include optically trapped neutral atoms and ion trap systems.

Significant progress in the understanding of AQC has revealed several open topics that continue to motivate research in this model of quantum computation. Foremost is the development of methods for fault-tolerant operation that will ensure the scalability of AQC for solving large-scale problems. Additionally, unequivocal experimental demonstrations that differentiate the computational power of AQC and its variants from conventional computing approaches are needed. Achieving these goals will require hardware development, algorithms and application development, and software tool development. Hardware development requires advances in the fabrication and control of quantum physical systems under the adiabatic restrictions. The development of quantum algorithms for applications that are known to be difficult or impossible for conventional computing provides the framework to benchmark quantum devices against classical computation at each generation of quantum processors. Quantum software tools allow researchers to interact with quantum processors in new ways to solve quantum algorithms and inform future hardware and software development.

The work of this dissertation uses tuning strategies available with quantum software tools to assess the quality of solutions returned by the quantum annealer for traditionally difficult optimization problems. The experiments apply a wide range of available tuning strategies to assess current quantum device performance, inform future software and hardware development, and provide a methodology for determining the optimal set of controls for an application of interest. All experiments were performed using the D-Wave 2000Q quantum annealer with currently available controls for manipulating tuning strategies.

1.3 Benchmarking Quantum Annealing

Over the past several years, various early-stage quantum processing units (QPUs) have been made available to researchers and institutions, which has driven interest in exploring the use of QPUs for research and commercial applications. Many approaches exist to building

quantum computing hardware and as the technology progresses, analyzing the performance at each development stage is important. This analysis process accomplishes several goals by

- Establishing common benchmarks for marking progress and comparing the performance of quantum computers with traditional and alternative computation methods
- Educating the larger scientific community on the current capabilities of quantum computing hardware for application implementation
- Informing future algorithm and hardware innovations

These goals can be accomplished in part with efforts to benchmark the solution quality when solving problems that are known to be difficult for classical computational methods and when adjusting the available tuning strategies of QA hardware.

The available QA tuning strategies take advantage of preprocessing, annealing, and post-processing controls. Preprocessing involves preparing the problem to be solved, including formulating the Hamiltonian, applying weights to the Hamiltonian (e.g., Lagrange multipliers), and embedding the problem onto the hardware. Modifying annealing parameters involves strategies for optimizing and tuning control options such as qubit initialization, reverse annealing, pause times, and anneal offsets. Post-processing considers manipulating the solutions returned by the quantum annealer.

The purpose of this research is to investigate the available QA controls to develop tuning strategies to improve performance. The benchmarks used in this dissertation measure the quantum annealers ability to solve an optimization problem by measuring solution quality through the probability of finding the optimal solution and the probability of non-adiabatic dynamics "breaking" the solution. Because the quantum computing field works from many angles to find computational advantages to quantum annealers, tuning strategies aid in assessing the current performance of QA machines. The findings from implementing tuning strategies also inform future hardware and software development.

1.4 Motivation to Study Quantum Annealing Performance

As the field of quantum computing advances, many development approaches have the goal of finding computational advantages from computers that use quantum mechanics. QA is one such approach that uses the principles of AQC, namely the adiabatic theorem, to solve problems. In many ways, QA is on the cutting edge of quantum computation. D-Wave Systems' quantum annealer was released in 2014 as the first commercially available quantum computer. Although many research groups develop universal quantum computers at a small scale to study the physics and investigate methods for scaling with error correction, D-Wave Systems has scaled much faster. D-Wave Systems' approach allows researchers to formulate classically difficult problems for the quantum annealer for testing, which drives interest across a wide variety of applications and opens opportunities to test hybrid computing strategies. However, this approach also has drawbacks. The D-Wave Systems quantum annealers are noisy, have no error correction, and do not use the full Hamiltonian necessary for universal quantum computation.

An opportunity exists to address the challenges of QA by developing and investigating tuning strategies that not only assess the performance of the quantum annealer but also inform future hardware development and software for tuning strategies. The preprocessing controls include all problem configurations an hardware settings prior to annealing. The annealing controls include the annealing schedule with timing constraints. The post-processing controls include all programs that process the output solution from the quantum annealer.

1.5 Tuning Strategies Research Design

The research conducted for this dissertation began by first building a framework to solve complex optimization problems with QA and then using that framework to test the capabilities of QA hardware to solve such problems. The primary research question is whether an optimal set of controls exists that can improve the solution quality returned

by the quantum annealer. This work seeks to answer the following questions: If an optimal set of controls exists, what is the methodology for finding such controls? Additionally, what do these tuning strategies reveal about the current capabilities and shortcomings of the QA hardware? This work also opens questions on how a problem's structure and complexity affect the probability of success and thus the benefit that tuning strategies may provide. Certain energy landscapes may positively or negatively affect the performance of the quantum annealer. This work is meant to provide a methodology for understanding the effects of problem complexity and structure on QA performance and determining an optimal set of controls for current applications of interest.

QA is in the early stages of development, and the limitations of this research center around the limitations to the QA hardware and controllable software. As mentioned previously, all experiments were performed using the D-Wave 2000Q hardware with the available Ocean software provided by D-Wave Systems [41]. The tuning strategies implemented in this dissertation used the currently available QPU hardware controls. The D-Wave 2000Q hardware has available controls in all three categories of preprocessing, annealing, and post-processing. However, theoretical controls such as local adiabatic evolution as discussed in Sec. 1.4 have not been fully realized in practice and, if available, could experimentally demonstrate quantum advantages for quantum algorithms over classical methods [148]. Advanced controls such as local adiabatic evolution are not currently available for QA hardware and were therefore not an experimental part of this research. However, control experiments were performed with the QA hardware controls available at the time of this experimentation.

This research focuses on developing a methodology for determining the optimal set of QA controls for combinatorial optimization problems by investigating the effect of QA controls on the probability to find the ground truth solution. Future researchers and application developers can use this methodology to determine whether their problem can be solved with high solution quality on a quantum annealer and whether an optimal set of controls exists for their application. This methodology also serves as an avenue for benchmarking the quantum annealer as the technology improves in future generations. Chapter 2 provides a detailed literature review covering a brief introduction to quantum computation, AQC,

and QA; the hardware implementations that are being developed in the field; the hardware controls that can theoretically affect AQC and QA performance; the experimental studies on currently available QA controls; the array of applications of interest that can use AQC and QA; and open questions in the field. Chapter 4 describes the methods by which all experiments were performed, including the QA platform for solving quadratic unconstrained binary optimization (QUBO) problems and the Markowitz portfolio optimization problem, which was used as a problem of interest in the experiments. Chapter 5 shows all major research findings organized by tuning strategy, including problem controls, preprocessing controls, annealing controls, and post-processing controls. Chapter **??** interprets all research findings as conclusions with suggestions for future research.

Chapter 2

Background

2.1 Quantum Computing

Quantum computing is a form of computation that explicitly uses quantum mechanical effects to process information; a quantum computer implements these principles to solve computational problems. Quantum computers are distinguished by the use of the quantum states of a physical system to store information [124]. Unlike classical bits of information that are either 0 or 1, a quantum bit of information, or qubit, is a linear superposition of 0 and 1. Formally, qubits represent normalized vectors within a two-dimensional Hilbert space that can be visualized as points on the Bloch sphere shown in Figure 1. In particular, the opposing poles of the Bloch sphere correspond to the orthogonal states of a physical system whereas every point on the surface represents a valid linear superposition [12].

In general, interactions between quantum physical systems extend the principle of superposition across multiple qubits, and the joint quantum states prepared by these interactions will manifest as correlations in behaviors for the physical systems. Quantum states are said to be entangled when these composite behaviors cannot be separated into independent processes—the behavior of one system directly depends on the behavior of the other. Entanglement represents a fundamentally new type of correlation that cannot be reproduced within the context of classical physical theory [72]. It is a hallmark of quantum mechanics and, consequently, quantum computing.

A quantum algorithm is a set of operations that prepares superposition and entanglement and transforms a quantum state. The quantum algorithms proposed for full-scale quantum computers are expected to provide significant speedups over best-in-class conventional methods for certain problems. For example, quantum computers are expected to efficiently simulate quantum mechanics, which would provide insights into the structure, properties, and behavior of systems with many particles or interactions [58]. Additional quantum-algorithmic results affect the development of new methods for optimization, unstructured search, integer factorization, systems of linear equations, and many other application areas [124] [116]. These potential performance gains continue to motivate the development of quantum computers and applications to practical problems.

The concept of a universal quantum computer formalizes how arbitrary computations can be performed [48] and enables a theory of quantum computational complexity [14]. Prominent classes from conventional complexity theory are found to have distinct quantum analogs that categorize new classes of problems. More information on complexity classes is mentioned in Sec.1.1.4. Conceptual requirements for an abstract quantum computer provide criteria that guide experimental efforts to realize this technology and are often referred to as "DiVincenzo criteria" [51].

Problem classes that are challenging to solve with conventional methods of computation are of specific interest for quantum computation. In particular, the NP class represents decision problems with solutions that can be verified in polynomial time but are not known to be solved in polynomial time. As the size of the input problem grows, the conventional resources required to find a solution grow super-polynomially with respect to input size. Problems from the NP class arise in many notable applications such as cryptography [172], routing and scheduling [100], and more. Therefore, quantum computation is of interest for this class of problems. Adiabatic quantum computation is especially focused on optimization problems that fall in the NP class but additionally must be decision problems. Such problems fit naturally into AQC because their solutions can be encoded in the ground state and translated into an Ising Hamiltonian in polynomial time [104]. Some example NP optimization problems that have been formulated for AQC include satisfiability (SAT) problems [71], finding cliques [32], and exact cover [36].

Experimental efforts to realize AQC and QA have used a variety of quantum physical systems, including the spin of individual electrons [158], the polarization of single photons [4], and the quantized magnetic flux in superconducting electronics [49]. However, interactions between such quantum physical systems and the environment may generate entangled or correlated states that represent a loss of information to the uncontrolled surroundings. The coherence time quantifies the timescale over which information may reliably persist in the primary system, which sets a fundamental limit on how many operations can be performed. The development of well-characterized, reproducible qubits continues to be an active area of experimental research for the realization of noisy, intermediate-scale quantum computing devices [140].

A quantum computing model defines how a quantum computer should operate. Several models have been developed to implement universal quantum computation, each being computationally equivalent but operationally distinct. The model of AQC is described in detail in the following section, and a summary of the prominent circuit model can be found elsewhere [124]. The choice of quantum computing model may be tailored to the intended purpose of a quantum computer as some models more efficiently express certain algorithms. Additionally, limits on the controllability of a quantum physical system may also make certain computational models better suited.

2.2 Adiabatic Quantum Computing

AQC is a model of computation that uses quantum-mechanical processes operating under adiabatic conditions. This model employs continuous-time evolution of a quantum state $\psi(t)$ from a well-defined initial value to compute a final observed value. The evolution is modeled by the Schrödinger equation

$$i\hbar\frac{\partial\psi(t)}{\partial t} = H(t)\psi(t) \tag{2.1}$$

operating in the presence of adiabatic changes to the governing Hamiltonian $H(t)$ over the range $t \in [0, T]$, where $\hbar$ is Planck's constant divided by 2π. AQC is computationally equivalent to all other quantum computing models, including the circuit and topological models, and it can efficiently solve any problem in bounded-error quantum polynomial time

(BQP) [5]. However, AQC was originally proposed as a method for solving SAT problems [57] and it has received attention for the simplicity by which combinatorial optimization problems can be cast in Hamiltonian forms [104].

The principles of operation for AQC derive fundamentally from the adiabatic theorem, which states that a quantum mechanical system will remain in an instantaneous eigenstate of the Hamiltonian provided conditions on the internal energy and timescales are met [23]. In the simplest case, the adiabatic theorem requires:

1. An energy gap between the populated eigenstate and all other excited energy states
2. Sufficiently slow evolution time to suppress internal excitation

The significance of these conditions may be illustrated through the example of a time-dependent Hamiltonian:

$$H(t) = A(s(t))H_A + B(s(t))H_B, \tag{2.2}$$

where $s(t)$ is the control schedule, $A(s(t))$ and $B(s(t))$ are the time-dependent amplitudes which control the interpolation of the initial and final Hamiltonians, and H_A and H_B represent self-adjoint linear operators acting over a Hilbert space of dimension $N = 2^n$, with the integer n referring to the number of qubits. In particular, the schedules should be smooth and differentiable and should satisfy the boundary conditions $A(0) = 1$ and $B(0) = 0$ while $A(s = 1) = 0$ and $B(s = 1) = 1$. The j-th instantaneous eigenstate $\phi_j(t)$ for $H(t)$ is then defined as

$$H(t)\phi_j(t) = E_j(t)\phi_j(t), \tag{2.3}$$

where $\{E_j(t) : j = 0 \text{ to } N - 1\}$ is the instantaneous eigenspectrum of $H(t)$. If the quantum state is prepared in the j-th energy eigenstate of $H(0) = H_A$ at time $t = 0$, it will remain in the j-th instantaneous eigenstate under Schrödinger evolution provided the adiabatic conditions are met. The adiabatic theorem promises that the quantum state at time T will then be the corresponding j-th energy eigenstate of $H(T) = H_B$. The transformation from a known initial state to a final, potentially unknown state represents adiabatic quantum computation.

In practice, the instantaneous ground state of $H(t)$ is typically chosen for the computation, and the initial Hamiltonian $H(0)$ is selected to have a ground state that can be prepared directly. The energy spectrum gap between the quantum computational state and all other states must remain nonzero to ensure adiabatic evolution. Assuming the computational state is the ground state, the minimum spectral gap of $H(t)$ is defined as

$$g_{min} = \min_{0 \leq t \leq T} \min_{j \neq 0} [E_j(t) - E_0(t)], \tag{2.4}$$

where $E_j(t)$ is any higher-lying energy state. The minimum spectral gap g_{min} sets a lower bound for the smallest internal energy scale that contributes to undesired coupling of the computational ground state to erroneous, higher-lying eigenstates. These transitions arise from diabatic quantum dynamics, with the most common example typified by Landau-Zener transitions [176]. When the Hamiltonian changes quickly, a non-negligible probability exists for diabatic transitions that corrupt the computation. These transitions may be avoided by selecting the evolution time T to be much longer than the timescale set by the inverse of the spectral gap. Worst-case lower bounds for general Hamiltonian instances suggest T must scale as $O(g_{min}^{-3})$ [75], whereas more restrictive settings can improve this to $O(g_{min}^{-2})$ [55].

A central concern for assessing the computational efficiency of AQC is determining how the adiabatic timescale T must grow as the size n increases. Generally, the spectral gap g_{min} will decrease as n increases, but the rate of this decrease greatly affects the time complexity of the computation. For example, assuming $g_{min} \propto k^n$ for some positive constant k, the minimum time T needed to ensure the adiabatic condition would increase exponentially with problem size. This increase would indicate that an exponential increase in time is required to solve a general problem within the AQC model. Presently, theoretical estimates for how the minimum spectral gap scales with size are inconclusive for the general setting but may be developed for specific Hamiltonian models (see examples from Albash et al. [8]). Answering the general spectral gap question appears to be computationally difficult, if not impossible [40]. Additionally, the choice of temporal interpolation strongly affects the instantaneous spectrum. Some specific instances of Hamiltonians have been found to support

minimum computational times that are sub-exponential in n, provided a more general form of the temporal interpolation is employed [148].

As noted previously, the initial Hamiltonian $H(0)$ should enable convenient preparation of the quantum state from which the computation begins. A frequently used Hamiltonian is the form

$$H_A = -\sum_{i}^{n} \hat{x}_i, \tag{2.5}$$

where each term $\hat{x}_i$ represents the n-fold tensor product of $n-1$ identity operators I and the Pauli operator $\hat{x}$ for the i-th qubit. The energy eigenstates and eigenvalues for this Hamiltonian can be analytically constructed, and experimental methods for preparing those eigenstates have been developed [57].

The final Hamiltonian $H(T)$ encodes the problem to be solved by AQC as the prepared quantum state expresses the corresponding solution. The Hamiltonian complexity determines the types of problems that can be solved. For example, a Hamiltonian capable of expressing a sum of arbitrary 2-local interactions is Quantum Merlin Arthur (QMA)-complete, whereas solving the problem of a 2-local Ising Hamiltonian is NP-complete [89] [24]. In particular, the latter Ising model provides a direct connection to a variety of computationally significant problems. Although the Ising model was originally developed to describe the physical pairwise interactions between the spins in a magnetic material, it has since been used to describe many other systems composed from interacting binary variables. A quantum mechanical version of the Ising Hamiltonian may be cast in the form

$$H = -\sum_{i,j} J_{ij}\hat{z}_i\hat{z}_j - \sum_{j} h_j\hat{z}_j + \gamma, \tag{2.6}$$

where h_i represents a bias on qubit i, $\hat{z}$ is the z Pauli operator, γ is a constant, and J_{ij} describes the coupling strength between qubits i and j. This formulation can be used to model a wide variety of combinatorial optimization problems [104] in which they are reduced to finding the ground state of an appropriate Hamiltonian. Section 2.6.1 provides additional examples, but the type of Hamiltonian plays a prominent role in the problem complexity.

2.3 Quantum Annealing

Efforts to realize AQC using quantum physical systems are susceptible to nonideal conditions that undermine the promise of the adiabatic theorem, which presents a fundamental challenge in the control of quantum physical systems. A relaxation of the adiabatic condition is captured in the complementary computational heuristic of QA, which accommodates physical systems operating at finite temperature and in open environments. QA is a method for identifying the minimum of an objective function using an approach that is based on the principles of AQC but fails to meet its stringent requirements. Additional dynamical behaviors, including stochastic dynamics, may be present during the actual evolution of the quantum state [82], which revokes the guarantee of remaining in the instantaneous eigenstate, though it may be sufficiently close in practice.

In practice, QA evolves a quantum state under the time-dependent Hamiltonian in Eq. (3.2). However, the dynamics may not be modeled as a Schrödinger evolution. When the dynamics are insufficiently slow, the quantum state will mix with nearby energy eigenstates and the probability to observe the expected outcomes will decrease. Additionally, the nonzero temperature of operation for QA invalidates the pure state description. A statistical mixture of energy eigenstates is a more appropriate model for initialization of the computation. The mixture of the initial state depends on the local operating temperature as well as the energy splitting between levels used for the computational basis.

Analyzing the results from QA requires statistical sampling to build confidence in the observed outcome. The collected results may then be used in decision-making processes. For example, when the lowest energy eigenstate is the solution, higher energy states can be rejected. However, this sampling comes with a cost of computational resources proportional to the number rejected. Additionally, QA lacks a guarantee that the sought-after solution will be included in the observed results. If the dynamics nearly satisfy the adiabatic condition, the probability distribution should be concentrated at energies near the sought-after state.

Similar behaviors in performance are observed with several important methods of classical computation such as simulated (thermal) annealing (SA). In SA, the system state overcomes these energy barriers through random changes driven by temperature fluctuations [68].

However, a probability exists for the system to get trapped in a local minimum if the temperature fluctuations are not sufficient to cross a high energy barrier. In SA, the energy landscape in static and thermal excitations drive dynamics as seen in Figure 2. Conversely, QA changes the energy landscape to drive the system state toward the energetic minimum. QA can also exploit quantum tunneling through barriers.

Although QA more closely models the behavior of quantum physical systems, the possibility of nonadiabatic effects obscures a clear separation with conventional computing complexity. The use of quasi-adiabatic evolution of mixed quantum states in an open environment is typically characterized as a heuristic because it lacks many of the computational promises offered by AQC. This characterization has led to empirical evaluations of the efficacy of QA for solving both real and synthetic problems with mixed results [88, 168, 93, 21, 45]. In particular, how to best categorize the computational power of QA even under relatively well-understood Hamiltonians remains unclear. For example, a stoquastic Hamiltonian is defined to have only real, nonpositive off-diagonal matrix elements in the standard basis representation [24]. This restricted form captures the quantum transverse Ising Hamiltonian that underlies current experimental approaches to QA. Theoretical evidence indicates that stoquastic Hamiltonians are insufficient for universal AQC when restricted to the ground state, but permitting excited-state evolution is found to remove this limitation [81]. By contrast, non-stoquastic Hamiltonians are found to be more expressive for universal quantum computing, but the size of the minimum energy gap for these Hamiltonian remains unclear in general [8]. Solving non-stoquastic Hamiltonians efficiently may be an important step toward universal AQC and enhancing solution quality using QA [125].

2.4 Quantum Annealing Hardware

Designing quantum computing hardware to implement either AQC or the QA heuristic requires time-dependent control of the Hamiltonian governing an array of quantum physical systems [67]. For computation, the hardware must also encode a relevant problem into the time-dependent Hamiltonian by programming interactions between the physical qubits and measuring the final prepared quantum state. The leading technology to demonstrate

AQC is currently superconducting electronics, but other systems have been proposed for similar purposes (e.g., trapped ions [177]). A leading implementation of the QA heuristic is found in the recent family of processors produced by D-Wave Systems, which are based on a superconducting flux-qubit design [25]. A wide variety of qubit designs are possible, with many specific choices of superconducting electronics being developed to support the encoding of a two-level system through the use of Josephson junctions [170]. A Josephson junction is a nanoscale insulating layer on the superconducting loop and is a leading design choice for AQC/QA implementations [79]. Although electrons move freely on the surface of a superconductor with no resistance, electrons probabilistically tunnel through the Josephson junction to form a superposition of states and thus a two-level system. A flux qubit is an example that prepares a quantized magnetic flux [80] in a superposition of states using a Josephson junction as shown in Figure 3. Although in a superposition of states, external magnetic fields can be applied to change the potential energy landscape or "weight" of each qubit according to the adiabatic theorem. The qubits are also arranged in such a way that the magnetic field of each qubit directly depends on its surrounding qubits via magnetic interference. This magnetic coupling is analogous to entanglement and is mathematically identical for the purpose of quantum computation. After the evolution is complete (at final time T), superconducting quantum-interference devices, which are inductively coupled to each qubit, detect and read out the magnetic field/state of each qubit [74].

A key challenge in building a scalable AQC/QA device using superconducting electronics is maximizing the connectivity of the physical elements. An ideal design would enable all-to-all connectivity among the physical elements that represent the qubits as this ensures the most direct form of interactions for encoding an arbitrary problem Hamiltonian. However, the two-dimensional plane in which superconducting circuit technology is typically fabricated imposes a physical restriction on the degree of connectivity that can be fabricated. Consequently, trade-offs in the physical design are required. Such limited connectivity may still enable encoding of an arbitrary Hamiltonian but at the expense of redundant encoding of logical variables as chains of (perfectly) correlated spins. The challenge of mapping this smaller, more densely connected graph into the limited connectivity of available hardware is known as minor embedding [34], and a wide variety of techniques have been developed for

this preprocessing step [95, 166, 62]. The Chimera graph topology is used for all research in this dissertation and can be seen in Figure **??**.

A second key challenge for hardware implementations of AQC/QA is minimizing the noise arising from the environment and the applied control signals. In particular, design choices for superconducting electronics can enable certain features of the physical system to be more or less resilient to noise in the electrical control signals used [65]. For the flux-qubits that feature prominently in AQC/QA implementations, a known sensitivity to flux noise arises from magnetic-field fluctuations within the electronic circuits, which may be due to direct noise in the applied electrical fields or induced noise in nearby conductors [144]. In either case, the precision with which the time-dependent evolution may be controlled is limited by such noise.

2.5 Quantum Annealing Controls

The control of AQC/QA hardware requires tuning the time-dependent Hamiltonian by which the physical system evolves. Methods for adjusting and tuning the initial and final Hamiltonian parameters as well as the interpolating time-dependent schedule are necessary to drive the dynamics of problem-specific quantum states. These controls may be used to improve the performance of AQC or QA programs by adjusting the interpolation to either better approximate adiabatic evolution or refine the problem definition. Several approaches exist, including varying the rate at which the energy landscape evolves, tuning the parameters or weights in the formulation of the Hamiltonian [34, 141], and adjusting the initial Hamiltonian. These controls determine the dynamics and can affect the quality of the solution as well as time to solution.

2.5.1 Theoretical AQC Controls

As described by the adiabatic theorem in Sec. 2, the time T needed to ensure adiabatic evolution of the problem Hamiltonian is inversely proportional to the minimum energy gap g_{min}. Therefore, the shortest possible total T is dictated by g_{min}, based on the assumption that the rate of evolution remains constant in time [57]. However, the total time T of a

quantum computation via adiabatic evolution can be improved by applying the adiabatic condition locally to infinitesimal time intervals dt for $H(t)$. As shown in Figure 4, the spectral gap changes over time and may only reach its minimum value in a very small region of the dynamics. Therefore, a more optimal schedule would account for the local spectral gap by increasing the rate of evolution when the energy gap widens and decreasing the rate of evolution when the energy gap narrows, as shown in Figure 5.

The advantages of local adiabatic evolution have been applied to an example of Grover's search algorithm, which returns a sought-after value encoded into a target Hamiltonian [148]. Traditionally, this search problem takes on average $N/2$ queries, where N is the number of entries in the database. Grover's algorithm for the circuit model of quantum computing was shown previously to obtain a quadratic speedup relative to the best classical method [124]. Roland and Cerf extended this analysis to AQC by showing that a similar $\sqrt{N}$ speedup is possible when using local control of the schedule and an evolution time [148]

$$T = \frac{\pi}{2\epsilon}\sqrt{N} \tag{2.7}$$

for recovery error $\epsilon \ll 1$.

To implement this sort of scheduling, some understanding is needed on how the energy landscape fluctuates in time, namely, how the energy gap g changes. Therefore, g_{min} must be approximated for each time slice dt. Knowing g_{min} for dt requires knowing the ground state and first excited state for all $H(dt)$. However, some potentially powerful tricks could approximate an evolution schedule prior to solving a problem. For instance, similarities may exist among problems that fall into various categories. Preprocessing methods could be used to sweep over parameters to find optimal controls, or sampling the energy landscape could reveal an optimal annealing schedule for a category of problems. Additionally, key features of a problem may be used to approximate an annealing schedule through machine-learning methods.

Another mechanism for improving total anneal time is to control the initial Hamiltonian. This mechanism was first suggested by Farhi et al. when they chose the initial Hamiltonian (H_A) for a set of 3-SAT optimization problems (SAT problem limited to a maximum of three

literals) known to have a very small g_{min} in the standard AQC model [56].

$$H_A = \frac{1}{2}\sum_{i=1}^{n} c_i(1 - X_i), \tag{2.8}$$

where c_i is chosen to be 1/2 or 3/2 randomly and with equal probability. With this method, the energy landscape is modified such that a high probability exists to remove the narrow energy gap and get a larger g_{min} if the problem instance is solved many times with different randomly selected H_A [56]. This method is attractive because problems can be solved quickly and globally without prior knowledge of the energy landscape. However, knowing how many different H_A to use for a particular problem instance to ensure that a solution close to the ground state is discovered can be difficult.

Another strategy is to formulate the initial Hamiltonian such that its ground state is the best guess for the ground state of the problem Hamiltonian [137]. If after running the problem some number of times, a lower energy state (e_{new}) is discovered, e_{new} is set as the new ground state of H_A, and the process is repeated. This process can be very useful for problems in which some prior intuition exists for what the answer might be; it is often referred to as the "warm-start approach."

2.5.2 Experimental QA Controls

In practice, nonideal behaviors arise in practical implementations of QA. Section 3.1.1 represents QA under ideal adiabatic conditions that are difficult to realize in actual quantum devices. Real-world quantum annealers have limits in the ability to control the Hamiltonian and quantum dynamics [133]. Additionally, the presence of ill-characterized environmental couplings give rise to flux noise [110]. The imperfect setting of the Hamiltonian parameters $(h, J_{i,j})$ by the analog control circuits gives rise to a small intrinsic control error [92]. These errors undermine the accuracy of the physical hardware [167, 133]. Finally, annealing too quickly may violate the essential adiabatic conditions [57], whereas annealing too slowly may lead to undesired thermal excitations of the quantum state due nonzero temperature fluctuations [126]. This multitude of effects complicates the description of QA and the assessment of its performance.

Given the implicit dependence on several competing factors, a variety of strategies have emerged for controlling QA to maximize probability of success in recovering the ground state, increasing the time to solution, and minimizing errors in the quantum computational solution. These control strategies include efficiently mapping the problem Hamiltonian onto the physical hardware Hamiltonian, tuning annealing schedule, applying variable transformations to mitigate control biases, and using reverse annealing to refine initial solutions [92, 174].

Various QA controls are currently available to employ tuning strategies. The preprocessing controls are defined by any parameters that prepare the logical problem and the physical hardware prior to annealing and include the embedding algorithm, chain strength c_s, and number of spin reversal transforms (gauge transformations) g [92]. The annealing controls are the parameters which dictate the path of the annealing schedule and include the optimal anneal forward annealing time T and the reverse annealing control schedule parameters such as the initial state e_i, pause time t_p, the point in the schedule that is annealed back to s, and the ramp t_r and quench times t_q. The post-processing controls are the parameters which process the state returned by the quantum annealer as it is converted into the logical state that solves the problem of interest. This can also include a classical local search that uses the quantum state as the initial state of the search to build a hybrid quantum-classical solver. In this section, these control will be further defined and demonstrated through review of prior literature.

Embedding Algorithms

QA hardware currently has limited coupling between qubits, which poses an obstacle for solving complex problems with many interactions. Minor embedding algorithms navigate this issue by mapping each logical spin of a problem to multiple physical qubits to increase the effective connectivity between qubits, but this method can be resource intensive because the higher the required connectivity, the larger the problem size n [35, 167]. The Hamiltonian encoding the computational problem must be mapped into the physical hardware and satisfy the constraints of limited connectivity. The D-Wave 2000Q hardware supports a sparse Chimera graph as visualized in Figure **??** with 400 unit cells in which physical qubits are

not fully connected but instead have a maximum connectivity of six couplers. The available D-Wave 2000Q computers that were accessed remotely for this research was the 2000Q_2 and 2000Q_5 both of which have some known hardware imperfections where a number of physical qubits with these imperfections were not used. The embedding algorithms used in this research avoided regions with these imperfections.A fully connected graph, such as in Figure 7, must be strategically embedded onto the more sparse Chimera graph. A single spin from the input Hamiltonian may be realized in hardware using multiple physical qubits that form a strongly interacting representative chain of spins. By judiciously choosing these chains and their interactions, the original input Hamiltonian may be constructed. This process, known as embedding, depends on the input problem as well as the target hardware connectivity. In general, embedding is NP-hard for arbitrary input graphs [35], and upper limits exist on the maximum graph that can be embedded [94]. For example, the largest fully connected problem that can be embedded onto the D-Wave 2000Q has ∼60 spins, whereas the limit in practice depends on the number of faulty/inactive physical qubits in the device.

Embedding algorithms that optimize chain length may greatly reduce the number of physical qubits required by considering problem symmetry as well as the location of faults in the hardware. Here, two embedding algorithms widely used in programming the D-Wave 2000Q are highlighted. The first method by Cai, Macready, and Roy is based on randomized placement and search for the weighted shortest path between spin chains [27]. This method, denoted as CMR, applies to arbitrary input graphs but typically creates a distribution of chain lengths. By contrast, a second method by Boothby, King, and Roy based on a clique embedding typically generates shorter and uniform chain lengths of size

$$l_c = \frac{n}{4} + 1 \tag{2.9}$$

for n logical spins [22]. Clique embedding was developed with fully connected graphs in mind to keep chain lengths small, and graphs with shorter chains can lead to better tunneling dynamics and a lower probability of errors in the chain caused by noise [175, 166, 22].

Chain Weight

Previously, Choi has demonstrated that minor embedding can be used to extend the connectivity of quantum annealing hardware [34, 37]. While the logical Hamiltonian is comprised of a set of vertices V_l with h_i qubit weights and edges E_l with $J_{i,j}$ coupler weights, the physical embedded graph is comprised of a new set of vertices V_p and edges E_p. In this way, the logical problem is embedded with the graph $G_l = [V_l, E_l] \rightarrow G_p = [V_p, E_p]$ such that each i^{th} vertex in V_l is embedded as a set of vertices in a chain (subgraph) T_i [151, 37]. In this process, a logical qubit is embedded onto the hardware as a chain of physical qubits such that the number of possible logical qubit couplings are extended. The new Hamiltonian is given by

$$H^* = -\sum_{l \in V^*} h_l^* \sigma_l^z - \sum_{(l,m) \in E^*} J_{l,m}^* \sigma_l^z \sigma_m^z \tag{2.10}$$

where $G^* = [V^*, E^*]$ is the hardware graph including the logical graph and physical embedding subgraphs. The new physical Ising coefficients are given by

$$h_l^* = \frac{h_i}{|T_i|} \tag{2.11}$$

for all $l \in V_{T_i}$ and

$$J_{l,m}^* = \begin{cases} \frac{J_{i,j}}{\text{edges}(T_i, T_j)}, & \text{for } l \in T_i, m \in T_j, \text{and } i \neq j \\ k, & \text{for } l \in T_i, m \in T_j, \text{and } i = j \\ 0, & \text{otherwise} \end{cases} \tag{2.12}$$

where k is the intra-chain strength for T_i and chosen to keep the intra-chain qubits highly correlated relative to the inter-chain coupler strengths $J_{i,j}$. A method for embedding developed by Boothby, King, and Roy based on a clique embedding typically generates shorter (relative to a random embedding) and uniform chain lengths of size

$$l_c = \frac{n}{4} + 1 \tag{2.13}$$

for n logical qubits [22]. Theory suggests that shorter chain lengths lead to a lower probability of errors caused by noise[175, 166, 22]. Ensuring an embedded chain of qubits collectively represents a single logical variable requires an intra-chain coupling (chain weight k) that is larger in magnitude than the inter-chain couplings. In other words, the chain of physical qubits must be strongly coupled to remain a single logical qubit. However, chains can become "broken" in so far as individual physical qubits within the chain differ in their final state.

In general, chain breaks arise from nonadiabatic dynamics that lead to local excitation out of the lowest energy state with theory suggesting that longer chain more susceptible to these effects [92, 54]. King et al. observed that the chains break with higher probability when k is too weak, but the errors from noise on the hardware can be amplified if the chains are weighted too strongly, which can decrease the overall probability of finding the ground state [92]. Venturelli et al. found that the solution quality from solving fully connected graphs on the D-Wave 2 improved when sweeping over k to minimize the number of ground states returned with broken chains [166]. Hamerly et al.'s experiments with the D-Wave 2000Q revealed observations of increased probability to find the ground state (up to 4 orders of magnitude increase) from increasing chain strength such that the probability of chains breaking reduced to the order of 10^{-1} [64]. This research on the impact of k on QA performance shows that chain strength plays an important role in the quality of results returned from the quantum annealer. Namely, the chains strength must be large enough that the chain continues to represent the logical qubit throughout the anneal, but an optimal range may exist at which a k too strong will again decrease the probability of success due to over-powering the inter-chain couplings [147]. The sweet spot for k likely depends on the inherent noise of the hardware, but how much the optimal value of k depends on problem structure is unknown because the majority of experiments use fully connected graphs.

Number of Spin Reversal Transforms

Interactions between embedded chains arise from the required coupling between the logical spins. However, imperfections in the control of these spins lead to small biases that can become non-negligible for larger qubit chains and contribute to the complex dynamics describing the device. In turn, the probability for finding the expected ground state solution

could decrease because of these bias errors. The influence of these errors on the computational result may be mitigated by using spin reversal transforms to average out biases. As a gauge transformation, spin reversal redefines the Hamiltonian by replacing the biases and couplings for a subset of spins with their negated value [92, 134]. This transformation maintains the ground state of the logical problem. However, this transformation flips the sign of randomly selected qubits so that on average, their bias is reduced. This strategy mitigates errors on individual spins by balancing the noise on the device prior to annealing [141]. The native D-Wave spin reversal algorithm selects a random set of embedded physical qubits to reverse for a specified number of transformations (g), which parameterizes the control technique. Early work demonstrated that adding $g = 8$ for the D-Wave 1 device and $g = 16$ for the D-Wave 2 device improved results [20, 149]. King et al. demonstrated that spin reversal transforms improved results when $g = 10$ for a variety of random optimization problems, and they observed that spin reversal had a more significant improvement on the results when solving the more difficult 3-SAT problems compared with random instances where $J_{i,j}$ is a random nonzero integer and $h_i = 0$ [92]. In 2019, Pelofske et al. developed a genetic algorithm that optimized which particular set of physical qubits should be reversed for a given problem; although they observed improvements from using the genetic algorithm, the D-Wave 2000Q's native spin reversal algorithm was highly optimized [134]. These results suggest that spin reversal can improve the results returned from the quantum annealer by mitigating some inherent device noise, and this improvement may be noticeably greater for more noisy devices and for problems with a more complex embedding.

Total Annealing Time

As shown in Sec. 2.2, the adiabatic theorem dictates that T must be slow enough such that the system does not jump to an excited state. The optimal T is proportional to the inverse of the minimum energy gap, which is problem dependent and difficult to identify. However, in practice, QA has thermal noise and long anneal times decrease the probability for the system to remain in the ground state [142, 7]. Early experiments in QA found that even for the smallest allowable annealing time of $T = 5$ µs for the D-Wave 1 with ∼100 qubits and $T = 20$ µs for the D-Wave 2 with ∼500 qubits, T was suboptimally slow for the problems

tested due to device noise [20, 149]. However, King et al. observed that the D-Wave 2 device showed optimal times above $T = 20$ µs for some problem classes [92]. In 2018, Albash et al. investigated optimal T for the D-Wave 2X device (∼1,000 qubits) and the state-of-the-art D-Wave 2000Q device (∼2,000 qubits) with a class of "logical planted" problems, which used the majority of physical qubits on the devices and are designed to be theoretically ideal for QA by forming tunneling energy barriers in the energy landscape. This study found that although the optimal anneal time for many problems was shorter than the minimum available $T = 5$ µs on the D-Wave 2X device, the optimal T for the problems solved on the D-Wave 2000Q were over 50 µs [9]. These studies suggest that finding an optimal annealing time depends on the inherent noise of the device and the problem class.

Reverse Annealing Controls

Another method of QA is reverse annealing, in which the system starts in a initial state e_i and evolves in the reverse direction back to some point in the anneal s_p with the option of pausing for some time t_p before annealing back in the forward direction to $s = 1$. This technique was first proposed in 2011 by Perdomo et al. so that instead of starting an anneal from a superposition of all possible states, one could start an anneal from a good guess e_i [137]. The differences in the control schedules of forward and reverse annealing are demonstrated in Figure 6, where a linear reverse annealing schedule is compared with a linear forward annealing schedule using the amplitudes $A(s) = (1 - s)$ and $B(s) = s$. Notably, forward annealing controls increase monotonically with time, whereas reverse annealing controls include a change in the direction of the control schedule where the ramp time from $s = 1$ to s_p is $t_r = t_1$, the time paused at s_p is t_p, and the quench time back from s_p to $s = 1$ is $t_q = T' - t_2$. Thus, the reverse annealing controls that can be tuned include e_i, s, t_p, t_r, and t_q.

In 2017, Chancellor et al. suggested that enabling quantum annealers to start from an a state e_i was crucial for hybrid classical-quantum algorithms in which the quantum annealer can start from the results found from a classical algorithm [30]. D-Wave also released a white paper in which the authors found an example of reverse annealing that was up to 150 times faster at finding the ground state than forward annealing; they also observed an

optimal range of s between .7 and .8 in which the system could tunnel to a lower energy state with high probability, but the information provided to the system with e_i was still preserved [171]. Reverse annealing was implemented on the D-Wave 2000Q processor in 2018 by King et al.; they observed topological phenomena by reverse annealing a set of 50 samples in which each sample used the final state of the previous sample as the new e_i [91]. Ohkuwa et al. studied reverse annealing analytically to solve the fully connected ferromagnetic p-spin model and observed that if e_i was close to the ground state, reverse annealing improved upon forward annealing [127]. Marshall et al. conducted experiments with reverse annealing for problems whose energy landscapes were known to see if reverse annealing with a pause at an s_p near the minimum energy gap improved the probability of tunneling to the ground state. They observed that the probability of finding the ground state increased by over an order of magnitude if system was paused for some t_p just after the minimum energy gap [108]. Most recently, Venturelli et al. used reverse annealing combined with a classical genetic algorithm as a hybrid heuristic for solving portfolio optimization problems. The portfolio optimization problems were first fed into a genetic algorithm to retrieve a local minimum that was then used as the e_i for reverse annealing. This hybrid heuristic was observed to be more than 100 times faster at finding the global minimum than forward annealing [165]. These studies show that reverse annealing provides improvements in time to solution over forward annealing for certain problem formulations and offers greater flexibility for hybrid computation.

Post-Processing Error Correction

An additional control is required for decoding embedded chains to recover the computed logical spin state. Ideally, all chains remain uniform through the anneal, but noise and nonoptimal k can cause chains to break. In the absence of chain breaks, the logical value is inferred directly from the unanimous selection of a single spin state by every physical qubit. In the presence of chain breaks, several strategies may be employed to decide the logical value, including majority vote, discard, and a greedy descent [41, 92]. Majority vote selects the logical spin value as the value that occurs with the highest frequency in a chain. Discard ignores any solutions with broken chains. Greedy descent is a hybrid computing technique that takes the solution with broken chain returned by the D-Wave and feeds it

into a classical gradient descent algorithm to locally search for the solution. This greedy descent flips random bits in the broken chains of the solution to find the lowest energy. As mentioned in Sec. 2.5.2, reverse annealing can also be used as a post-processing technique for forward annealing. King et al. found small improvements in time to solution by implementing majority vote and an order-of-magnitude difference when implementing the greedy descent method [92]. In summary, post-processing can be used to interpret results from broken chains. Some of these methods such as discard and majority vote simply attempt to clean up random errors for benchmarking results. However, reverse annealing and greedy descent can be used to apply a local search around the embedded solution returned by the quantum annealer.

2.6 Quantum Annealing Applications

Several applications have been developed within the AQC model to take advantage of its explicit representation of optimization [115]. Suitable combinatorial optimization may be found in diverse areas, including unconstrained and constrained optimization [69, 153, 50, 16], number theory and graph theory [17, 163, 31, 78], and machine learning [122, 53, 119, 2]. Extension of these ideas to specific application problems has also received significant attention [120, 136, 166, 39, 150, 84, 113].

A second area of AQC applications is data analytics, and in particular, several applications have been developed to leverage QA to investigate probability distributions. Sampling from the prepared distribution provides a convenient method for calculating expectation values and other statistics. This data analytics technique forms the basis of many machine-learning methods [3, 46, 155, 119, 138].

A third emerging application area is the simulation of quantum Hamiltonian models, which is an important study focus for the physical sciences (e.g., in high-energy physics, chemistry, materials science, and biology). These applications sample the quantum state prepared by a model adiabatic process to estimate the physical features of quantum-mechanical systems [66, 90, 173].

2.6.1 Combinatorial Optimization

Optimization problems seek the best solution within a set of many candidates. They are often difficult to solve because the solution may not be obvious or it may not be easy to quickly search the candidates. Solvers for optimization problems have found a natural implementation within the AQC model that can be designed to follow evolution of the lowest energy state of a model Hamiltonian. By designing the model Hamiltonian to mimic an optimization problem, the preparation of a final quantum state can represent the solution to the original optimization problem. Moreover, AQC offers the promise that the solution is optimal when the adiabatic condition is met throughout the computation. The QA heuristic may also be used for optimization, but it does not guarantee that the optimal solution will be found.

The choice of the target Hamiltonian determines the type of optimization problems that can be solved using either AQC or QA. For example, currently available QA hardware relies on a target Hamiltonian that models an Ising Hamiltonian, as described in Eq. 2.10. A wide variety of problems have been reduced to the Ising form [104, 44]. The Ising Hamiltonian itself is naturally related to QUBO. The QUBO problem is formulated to find the minimum of a quadratic polynomial with binary variables, meaning

$$E(x) = \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{i,j} x_i x_j, \tag{2.14}$$

where $x \in \{0, 1\}$, c_i represents the linear term to be minimized, and Q_{ij} describes the quadratic interactions or correlations between variables. The conversion from QUBO to Ising form is performed by using the transformation of variable $x \in \{0, 1\}$ to spin $s \in \{-1, 1\}$. The classical spin variable can then be substituted by the corresponding quantum operator to achieve a quantum Ising Hamiltonian as defined by Eq. 2.6. A variety of different combinatorial optimization problems have been reduced to this form [104]. Several examples are described below.

3-SAT. A SAT problem determines whether an assignment for Boolean variables that satisfies a set of logical clauses exists. Such a problem arises in many practical applications, including product model checking and verification, planning and protocols, structural design,

and others [18]. The well-known Cook-Levin theorem from computational complexity theory places the 3-SAT problem in the NP-complete complexity class [38] [101], where this variant is specialized to cases for which every clause has, at most, three variables. Moreover, 3-SAT provides a constructive means by which a variety of other problems can be shown to lie in the NP hierarchy [86]. This hierarchy includes the Ising problem introduced above, which was identified by Farhi et al. as an important step in solving 3-SAT within AQC.

When introducing AQC as a method for solving optimization problems, Farhi showed how AQC could be used to solve 3-SAT using a 3-local Hamiltonian:

$$H_p = \sum_{i}^{n} H_i, \tag{2.15}$$

where H_i is a k-local Hamiltonian corresponding to the i-th clause [57]. If the smallest eigenvalue of H_p is 0, each clause of the problem is satisfiable. [8].

Binary Integer Linear Programming (BILP). BILP is an NP-hard problem that maximizes or minimizes an objective function subject to a series of constraints. Practical examples include portfolio optimization [165] [150], scheduling, networking [121], and more.

$$\begin{aligned} \max_{x} \quad & \sum_{i} a_i x_i \\ \text{such that} \quad & \sum_{i} b_i x_i = c, \end{aligned} \tag{2.16}$$

where a_i and b_i are variables, c is a hard constraint, and $x_i \in \{0, 1\}$. To solve this problem with AQC, it must transform into an unconstrained problem. The Ising Hamiltonian representation is given by

$$H = \theta_1 \sum_{i} x_i a_{ii} x_i - \theta_2 (\sum_{i} x_i b_i x_i - c)^2, \tag{2.17}$$

where the hard constraint around c becomes an unconstrained penalty for any deviation around c, and θ_1 and θ_2 are weights that balance the first and second terms of Eq. 2.17 during the maximization of the a parameters.

Graph Theory. Graphs are used to represent networks in which each node is an object and the lines connecting each node represent the relationships between objects. Graph theory can be used to solve many NP-hard optimization problems such as set cover [86], graph partitioning [61], graph coloring [77], and the NP-complete traveling-salesman problem [99]. The traveling-salesman problem, for instance, aims to find the shortest route that hits all desired destinations. These problems have been mapped successfully to QA hardware [104, 160, 111, 28].

2.6.2 Machine Learning

Machine learning using AQC- and QA-based methods has attracted significant interest for a number of applications. Broadly, machine learning infers correlations from data and several different approaches have been developed for this purpose within the AQC model. These approaches include both supervised and unsupervised training methods, which cast training as a global optimization problem that may be reduced to finding the lowest energy state of a corresponding Hamiltonian [103]. Because QA can find a solution that is close to optimal within a large number of possibilities, using this approach to either optimize or accelerate the training state of machine learning is appealing. As part of an unsupervised machine learning algorithm, O'Malley et al. used QA to recognize facial-feature patterns [128].

In another application, QA has been used to train a Boltzmann machine used in classification methods. A Boltzmann machine is an artificial neural network with visual and hidden nodes that encode information in their weighted couplings [70]. Whereas general Boltzmann machines do not restrict connections between nodes, a restricted Boltzmann machine only permits connections between nodes in different layers. In either model, the underlying network is expressed in terms of a Ising model that uses the spin variables as the nodes and the couplings to define connectivity [15]. QA with the Ising Hamiltonian can therefore be used with either type of Boltzmann machine to find the optimal weighted couplings. Training Boltzmann machines with AQC/QA has been tested experimentally [102, 138].

2.6.3 Quantum Simulation

An emerging application area for AQC/QA is the simulation of condensed-matter systems, where quantum many-body effects are often critical to the behavior of the modeled material. Understanding the behavior of materials is a challenge because simulating quantum many-body systems on a classical computer is computationally expensive. As originally proposed by Feynman [58], quantum computing offers a natural paradigm in which to both model and simulate these highly correlated materials. For example, a key problem for materials science is characterizing the energetic ground state of a material system; the formulation of AQC in terms of a target Hamiltonian provides a natural connection to this problem.

One approach to this materials science application is to use QA to simulate the magnetic phase transitions in an Ising Hamiltonian over a multidimensional lattice. This application prepares the ground state of the Ising Hamiltonian defined in Eq. (4.19) and then probes the prepared quantum state to recover the magnetization. By selecting the parameters for the Ising Hamiltonian, an expected phase of matter can be programmed and characterized. A recent demonstration validated the observed magnetization for different phases of a spin-glass system [66].

This approach may also be used to simulate quantum phase transitions provided the underlying Hamiltonian supports a model for such a system. For example, the Kosterlitz-Thouless phase transition can be simulated in a transverse Ising Hamiltonian over a square-octagonal lattice. The Kosterlitz-Thouless phase transition arises from frustrations and quantum fluctuations within this model Hamiltonian, and QA-based simulations have been validated directly against classical simulations [90]. The programmability of the target Hamiltonian enables simulation by QA to test many different model phases of matter.

Chapter 3

Tuning Strategy Methods

Many prominent questions remain about the expected physical and computational behavior of AQC, including clarification about how nonadiabatic effects impact the performance of QA devices as well as how the optimal run-time can be realized without prior knowledge of the underlying energy landscape [145, 108]. The design, development, and demonstration of reliable and programmable adiabatic quantum computers also remains an open endeavor. The idealized setting for satisfying the adiabatic condition exactly has yet to be realized in practice, and robust models for describing nonadiabatic effects will require better characterization about the underlying physical systems. Existing demonstrations that relax the adiabatic condition have made remarkable progress in the controllability of QA, but experimental evidence remains mixed on the computational performance, largely because of the relatively small amount of data that can be processed by these devices, which is insufficient for best-in-class comparisons.

Further experimental investigations into computational scaling using larger-capacity devices will help identify the significance of nonadiabatic effects. However, scaling AQC to arbitrarily large capacities will likely require methods for managing and correcting faults from noise in the devices and errors in the controls. The principles of fault-tolerant operation are well defined within the context of the circuit model of quantum computation, where redundantly encoded quantum states are actively corrected in the presence of noise. These principles may also apply to AQC, but a complete theory of fault-tolerant adiabatic quantum computation has yet to be developed.

Assuming the engineering of scalable quantum devices is achieved, AQC can be expected to have significantly affect computational science. Already, the formal reduction of many combinatorial optimization problems to the Ising problem have made AQC an attractive model for numerous known applications. However, AQC supports an even broader class of Hamiltonians, including those that are complete for BQP and QMA, and how this can be leveraged for new methods of quantum computation remains to be seen.

This research focuses on developing a benchmarking strategy for QA, using QA for combinatorial optimization, and understanding how tuning strategies/controls can mitigate nonadiabtic dynamics and improve the quality of solutions. This dissertation involved developing a framework for solving problems using the D-Wave 2000Q with the SAPI libraries in Python 2.7 and later the Ocean tools with Python 3.4. This software contains the tools needed to interact with the D-Wave hardware to solve problems as Ising Hamiltonians and configure the pre-processing, annealing, and post-processing controls [41].

This framework takes any QUBO or Ising Hamiltonian, converts it to quantum Ising, prepares a set of controls to be applied to the D-Wave hardware or Hamiltonian, embeds the Hamiltonian onto the hardware, runs a certain number of samples with specified controls, receives the solution state back from the D-Wave, detects errors, interprets those results with post-processing, and benchmarks the solution states for quality against a brute force solver. A formulation was then developed for Markowitz portfolio optimization that could be solved using QA. This formulation was implemented with random data (modeled after the stock market) to investigate the performance of the quantum annealer and how QA controls affect performance according to our benchmarks.

3.1 Quantum Annealing Platform for solving Quadratic Unconstrained Binary Optimization Problems

Optimization is integral to many scientific and industrial applications of applied mathematics, including verification and validation, operations research, data analytics, and logistics, among others [131, 161]. In many cases, exact methods of solution, including stochastic

optimization and quadratic programming, are computationally intractable; novel heuristics are used frequently to solve problems in practice [96]. QA offers a novel meta-heuristic that uses quantum mechanics for unconstrained optimization by encoding the problem cost function in a Hamiltonian [57, 118]. Recovery of the Hamiltonian ground state solves the original optimization problem and this approach has been mapped to a variety of application areas [52, 121, 157, 112]. Several experimental efforts have configured quantum annealers [79, 98, 164], and application benchmarking of these systems has shown that QA can find the correct result with varying probability of success [87, 93, 178, 76, 129, 9, 6].

3.1.1 Quantum Annealing

As discussed for AQC in Chapter 1, under ideal conditions, forward annealing evolves a quantum state $\Psi(t)$ under the time-dependent Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}\Psi(t) = H(t)\Psi(t) \qquad t \in [0, T] \tag{3.1}$$

where T is the total forward annealing time and the time-dependent Hamiltonian is

$$H(t) = A(s(t))H_0 + B(s(t))H_1. \tag{3.2}$$

where $s(t) \in [0, 1]$ is the control schedule and time-dependent amplitudes $A(s)$ and $B(s)$ satisfy the conditions $A(0) \gg B(0)$ and $A(1) \ll B(1)$. I consider the initial Hamiltonian $H_0 = -\sum_i^n \sigma_i^x$ as a sum of Pauli-X operators σ_i^x over n spins. The final Hamiltonian H_1 represents the unconstrained optimization problem with a corresponding ground state that encodes the computational solution. Only problems represented using the Ising Hamiltonian were considered:

$$H_1 = \sum_i h_i\sigma_i^z + \sum_{i,j} J_{i,j}\sigma_i^z\sigma_j^z + \beta \tag{3.3}$$

where h_i is the bias on the i^{th} spin, $J_{i,j}$ is the coupling strength between the i^{th} and j^{th} spin, σ_i^z is the Pauli-Z operator for the i^{th} spin, and β is a problem-specific constant. The Ising Hamiltonian is well known for representing a variety of unconstrained optimization problems [105].

Instantaneous eigenstates at time t are defined as

$$H(t)\Phi_j(t) = E_j(t)\Phi_j(t) \tag{3.4}$$

where j ranges from 0 to $N-1$ with $N = 2^n$ the dimension of the Hilbert space. For an initial quantum state prepared in the lowest-energy eigenstate at time $t = 0$, meaning $\Psi(0) = \Phi_0(0)$, adiabatic evolution under the Hamiltonian in Eq. (3.2) to time T will prepare the final state $\Psi(T) = \Phi_0(T)$ with high probability provided T is sufficiently large. In particular, the evolution must be much longer than the inverse square of the minimum energy gap between the ground and first excited states [57]. At time T, the prepared quantum state is measured in the computational basis to generate a candidate solution for the encoded problem.

Another variation of QA reverses the time-evolution process by beginning in an eigenstate of H_1. Known as reverse annealing, the initial quantum state evolves under Eq. (3.2) in the reverse direction. The Hamiltonian starts as H_1 at time $t = 0$ and evolves backward to a point s_p in the control schedule that corresponds to time t_1. The Hamiltonian then pauses for a time $t_p = t_2 - t_1$ before evolving in the forward direction from the value s_p at time t_2 back to the final Hamiltonian at time T', where the latter time represent the reverse annealing time. The control schedule for reverse annealing is then defined as [174, 132]

$$s'(t) = \begin{cases} 1 + \frac{(s_p-1)}{t_1}t, & 0 \le t \le t_1 \\ s_p, & t_1 \le t \le t_2 \\ s_p + \frac{(1-s_p)}{(T'-t_2)}(t-t_2) & t_2 \le t \le T' \end{cases} \tag{3.5}$$

The differences in the control schedules of forward and reverse annealing are demonstrated in Figure 6, where a linear reverse annealing schedule is compared with a linear forward annealing schedule using the amplitudes $A(s) = (1 - s)$ and $B(s) = s$. Notably, forward annealing controls increase monotonically with time whereas reverse annealing controls include a change in the direction of the control schedule in which the ramp time from $s = 1$ to s_p is $t_r = t_1$, the time paused at s_p is t_p, and the quench time back from s_p to $s = 1$ is $t_q = T' - t_2$.

3.1.2 Computational Methods

The D-Wave 2000Q quantum annealer was used for all experiments. The probability of success, the probability of chain breaks, and the energy distribution across each problem instance was calculated. For each instance, these metrics were estimated by collecting $N_s = 1,000$ samples of the computed solution. D-Wave's SAPI with Python 2.7 and later ocean tools with Python 3 was used to solve each instance of Markowitz portfolio selection using the hardware controls outlined in Sec. 3.1.3 [41]. Experiments began with the D-Wave 2000Q_2 which includes the spin reversal, embedding, anneal time experiments, and the majority of reverse annealing experiments. However, this machine was decommissioned during the reverse annealing experiments and the remaining experiments including the rest of the reverse annealing and chain strength experiments were solved using the D-Wave 2000Q_5 which is thought to have lower noise on the hardware. However, there was no major differences in probability of success and probability of errors between two machines for problems with the same set of controls. 1,000 samples were collected per problem over a set of 1,000 problems for forward annealing examples an 100 problems for revere annealing examples. The majority vote post-processing technique was implemented for any broken chains to interpret raw solutions returned by the D-Wave 2000Q. The program implementation of these experiments are available online and data is available upon request [Grant and Humble].

For benchmarking purposes, each problem instance was also sovled using brute force search for the minimal energy solutions of the QUBO formulation. The complete energy spectrum was computed for each portfolio instance. These energy spectrum and the corresponding states were then used as ground truth for testing the accuracy of results obtained from QA. By sorting the spectrum, the probability of success was benchmarked for reverse annealing using initial states e_i sampled from these different parts of the spectrum. The framework used to solve all QUBO problems is provided in Appendix A.

3.1.3 Quantum Annealing Controls

A subset of controls was investigated which are available for the D-Wave 2000Q, a programmable quantum annealer composed from an array of superconducting flux qubits

operated at cryogenic temperatures [26]. The D-Wave 2000Q consists of up to 2048 physical qubits arranged in a sparsely connected array whose governing Hamiltonian is described by a time-dependent, transverse Ising Hamiltonian [159] for which the Hamiltonian parameters in the device can be programmed individually. This research used the D-Wave 2000Q_2 and 2000Q_5 both of which had fewer than 2048 qubits due to hardware defects. This design enables a broad variety of computational problems, such as portfolio optimization, to be realized. The following sections briefly review some of the controls available to influence the success of solving these problems using QA.

Problem Embedding

As discussed in Sec. 2.5.2, embedding algorithms map the logical problem onto the physical hardware of the QA device. The D-Wave 2000Q quantum annealer has a maximum connectivity of 6 with a Chimera graph structure. Problems with high connectivity can be embedded on this Chimera graph structure with minor embedding algorithms such as CMR and clique embedding. A representative example of the output from these methods is shown in Figure 7 using a fully connected problem with 20 logical spins. Both methods are available in the D-Wave Ocean software library [42]. CMR and clique embedding algorithms were used to test which embedding protocol produces better quality solutions. Because an optimal embedding algorithm may be problem-dependent, this methodology can be used to test embedding algorithms for different problem structures.

The research that has developed and implemented embedding algorithms shows that these algorithms enable solving problems with higher connectivity than the hardware would otherwise allow. Experiments for this dissertation compared CMR with clique embedding for a set of fully connected problems for solution quality.

Chain Strength

As discussed in Sec. 2.5.2, chain strength k is negative bias strength applied to the intra-chain couplings of an embedding. For all experiments, the magnitude of k was high relative to the inter-chain couplings. The default chain strength for the D-Wave 2000Q quantum annealer is -1 [41]. For experiments that investigated annealing controls and embedding controls, all

h_i and $J_{i,j}$ values were normalized to range $[-1, 1]$. The chain strength was then set to -1, which is much stronger than the inter-chain couplings, which were around 10^{-2}. Experiments for this dissertation also investigated the optimal chain strength; for these experiments, h_i and $J_{i,j}$ values were again set to a range $[-1, 1]$ and chain strength varied between 0 and -2 to test the effects on solution quality.

Post-Processing

As mentioned in Sec. 2.5.2, post-processing methods are used to interpret chains with inconsistent values. The controls include discard, which throws out any solutions with broken chains; majority vote, which chooses the binary value that is counted to appear most frequently within the chain; and greedy descent, which is a hybrid algorithm that searches for the lowest energy by randomly flipping the value of bits that had a broken chain. Although discard is perhaps the most useful for analyzing the true performance of the quantum annealer, majority vote is useful in many cases for circumnavigating small errors in the chain. If many broken chains are split randomly between binary values, the D-Wave should reveal a low probability of success. If a tie exists in the vote for a chain, the D-Wave 2000Q post-processing program chooses a 1 [41]. Thus, majority vote does not hide inherent issues with noise or configuration. However, implementing the greedy descent could return high probability success in cases in which chains are broken with high probability, and therefore can mask poor QA performance. Greedy descent may be useful for studies in hybrid quantum-classical computing. Majority vote was used until the later experiments that explored each of these post-processing methods and their effect on probability of success relative to chain strength.

Spin Reversal

As discussed in Sec. 2.5.2, spin reversal transforms are used to mitigate the influence of bias errors in the device that appear from complex dynamics [92]. Spin reversal was implemented using D-Wave's native algorithm, which redefines the Hamiltonian by flipping the sign of the qubit biases and couplings for a random set of individual spins $s_r \rightarrow -s_r$ prior to annealing,

where for a spin being transformed,

$$\begin{aligned} h_r &\rightarrow -h_r \\ J_{i,j} &\rightarrow -J_{i,j} \end{aligned} \tag{3.6}$$

for either $i = r$ or $j = r$. Because this mapping transforms entire spins, the Ising Hamiltonian energy is preserved and the problem is not altered. The number of spin reversal transforms g for a set of samples N_s is the tunable control in which the samples are equally distributed among the g transformations. Spin reversal experiments described in Chapter 4 searched for optimal g for a particular optimization problem with random data.

Annealing Schedules

Tailoring the annealing amplitudes $A(s(t))$ and $B(s(t))$ is perhaps the most direct method to control forward annealing. The annealing schedules control the rate of change of the $H(t)$, which must be sufficiently slow to approximate the adiabatic condition [33]. Figure 8 shows the D-Wave 2000Q schedule. While forward annealing on the D-Wave 2000Q, $A(s(t)) >> B(s(t))$ at $t = 0$, $A(s(t))$ decreases and $B(s(t))$ increases for $0 < t < T$, and $B(s(t)) >> A(s(t))$ at $t = T$.

The optimal annealing time is problem-dependent and inversely proportional to the minimum energy gap [57], and in general, the value and position of the minimum energy gap for a given $H(t)$ is typically unknown and hard to identify. Extending the annealing time T arbitrarily long may not only be limited by hardware parameters but also be counter-productive because of competing thermal processes that depopulate the ground state [142, 7]. An upper limit exists for the total job time ($N_s T \leq 1$ s) as well as total annealing time ($T \leq 2$ s) on the D-Wave 2000Q.

When reverse annealing, three primary parameters for control exist: initial state, pause point s_p, and pause duration t_p. Reverse annealing starts from an initial state that may be a best-guess state, a known state, or a random state. Two reverse annealing heuristics are reinitialized and iterative. Although both heuristics start the set of samples with a specified initial state, reinitialized reverse annealing resets the initial state to the specified state for

each sample whereas iterative reverse annealing sets the initial state for each subsequent sample to the found/read-out state of its preceding sample.

Chapter 4

Benchmarking Methods

4.1 Quantum Annealing Metrics

The research in this dissertation benchmarks the quality of solution returned by the quantum annealer both in the probability of finding the most optimal solution and the probability for the solution to contain errors from non-adiabatic dynamics. These metrics are obtains by comparing the solution of each problem returned by the quantum annealer to the optimal solution found with a brute force solver. QA is characterized QA using the probability of success

$$p_s = |\langle \Phi_0(T)|\rho|\Phi_0(T)\rangle|^2 \tag{4.1}$$

defined as the overlap of the final, potentially mixed quantum state ρ prepared by QA with the pure state describing the expected computational outcome $\Phi_0(T)$. Empirically, the probability of success is estimated from the frequency with which the observed solution state matches the expected outcome. When the expected ground state solution is known, the statistic $\delta_i = 1$ if the i-th sample matches the known ground state and $\delta_i = 0$ if it does not. For the k-th problem Hamiltonian instance, the estimated probability of success is then defined as

$$\tilde{p}_s^{(k)} = \frac{1}{N_s}\sum_{i=1}^{N_s}\delta_i \tag{4.2}$$

where N_s is the total number of samples. The average over an ensemble of N_p problem instances is defined as

$$\tilde{p}_s = \frac{1}{N_p} \sum_{k}^{N_p} \tilde{p}_s^{(k)}. \tag{4.3}$$

A second metric for characterizing QA performance, and especially the non-adiabatic dynamics, is the number of chain breaks observed in the recovered solution samples. As noted previously, a chain break is observed when the chain of physical qubits embedding a logical spin has more than one unique spin value. The probability of chain breaks for a problem instance is given by

$$\tilde{p}_b^{(k)} = \frac{1}{N_s} \sum_{i=1}^{N_s} \epsilon_i \tag{4.4}$$

where the statistic $\epsilon_i = 1$ when the i-th sample solution contains at least one broken chain for any of the logical spins and $\epsilon_i = 0$ when no embedded chain is broken. The average probability of chain breaks over an ensemble of N_p problem instances is then defined as

$$\tilde{p}_b = \frac{1}{N_p} \sum_{k}^{N_p} \tilde{p}_b^{(k)}. \tag{4.5}$$

The effects of chain breaks can be mitigated by post-processing methods, such as majority vote, which make hard decisions on the logical spin value. The density of chain breaks is also analyzed for each problem to determine how chain strength control affects the severity of chain breaks from noise. The average ratio of broken chains per problem is given by

$$\tilde{r}_b^{(k)} = \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{c_b}{N} \tag{4.6}$$

where c_b is the number of broken chains and N is the number of qubits and therefore the total number of chains in the sample. This benchmark is used to plot the average ratio of broken chains for each of the problems for a particular problem size. The average ratio of broken chains for all problems is given by

$$\tilde{r}_b = \frac{1}{N_p} \sum_{k}^{N_p} \tilde{r}_b^{(k)}. \tag{4.7}$$

This benchmark is used to plot the average ratio of problem chain breaks for each problem size. The final benchmark was used to determine the probability for each qubit in an intra-coupling to differ from the global minimum solution when a chain breaks ($\tilde{p}_q$).

$$\tilde{p}_q = \frac{1}{N_b} \sum_{i=1}^{N_b} q_b \tag{4.8}$$

where N_b is the number of broken samples for each problem and q_b is a binary variable indicating whether the qubit in the broken chain is incorrect. This benchmark was used to plot a heatmap of the probability of each qubit to be faulty for all chains in the embedding for each problem size.

Although these metrics quantify the probability with which QA recovers the correct solution, additional information about computational performance comes from the distribution of all solution samples obtained. In particular, the distribution over sample energies provides a representation for the weight of errors in the solution samples. A distribution concentrated around the lowest energy indicates a small number of errors in the computed solutions, whereas a broad or shifted distribution hints at a larger number of errors. The energy computed from the i-th solution sample is denoted as $E(i)$ and the j-th energy bin is defined as h_j. The bin h_j counts the number of samples with an energy in the range $[j, j+1]\Delta$ where Δ controls the granularity of binning the energies. The resulting set $\{(j\Delta, h_j)\}$ approximates the energy distribution of the sampled solutions.

4.2 Portfolio Selection

Portfolio optimization selects the best allocation of assets to maximize expected returns while staying within the budget and minimizing financial risk. The Markowitz theory for portfolio selection focuses on diversifying the portfolio for risk mitigation [107]. Instead of allocating high percentages of a budget toward assets with the highest projected returns, the budget is distributed over assets that minimize correlation between the asset's historical prices. In this model, the covariance between purchasing prices serves as a proxy for risk in

which assets are considered to be more risky the more positively correlated they are. This section reviews the methods by which the benchmark problems are generated and solved.

Markowitz portfolio optimization is condisered as a quadratic programming problem that determines the fraction of available budget b to allocate toward purchasing assets with the goal of maximizing returns while minimizing risk. This approach to portfolio optimization is known to be NP-HARD [29] which makes quantum annealing a promising method for solving these problems [165, 150, 113].By selecting a partition number w, the fraction $p_w = \frac{1}{2^{(w-1)}}$ represents the granularity of the partition. The portfolio optimization problem selects how many of those partitions to allocate toward each asset with an integer z_u. Thus, the fraction of b to invest in each u^{th} asset is given by $p_w b z_u$, and portfolio optimization identifies how much of the m assets to select given the budget b and a risk threshold c. Thus, portfolio selection is cast as

$$\begin{aligned} \max_{z} \quad & \sum_{u=1}^{m} r_u z_u \\ \text{s.t.} \quad & \sum_{u=1}^{m} p_w b z_u = b, \\ & \sum_{u,v=1}^{m} c_{u,v} z_u z_v \leq c \end{aligned} \tag{4.9}$$

where for the u^{th} asset, r_u is the expected return and $c_{u,v}$ is the historical price correlation between assets u, v.

In Eq. (4.9), the first term represents maximization of the expected returns over the available assets. Many methods can forecast expected returns (e.g., based on market price, expert judgement, and historical price data) [73, 109]. For simplicity, expected returns are modeled as

$$r_u = p_w \bar{a}_u \tag{4.10}$$

where $\bar{a}_u$ is the average of a_u, the history of price data for the u^{th} asset. The first constraint in Eq. (4.9) places a hard constraint on the total allocation of assets to sum to b. This constraint penalizes portfolios that do not allocate the entire budget as well as those that over-commit. Finally, the second constraint accounts for diversification by asserting that the sum of covariance between asset prices $c_{u,v}$ be less than or equal to the risk threshold

c. The historical price covariance is calculated as the correlation between pairs of assets by comparing the p_w fraction of each asset's historical price data. Here, "covariance" is defined as

$$c_{u,v} = \frac{p_w^2 \sum_{l=1}^{N_f} (a_{u,l} - \bar{a}_u)(a_{v,l} - \bar{a}_v)}{N_f - 1} \tag{4.11}$$

where $a_{u,l}$ is the l^{th} historical price value for asset u and N_f is the number of price points in the historical data.

This variation of Markowitz portfolio selection is solved using QA by casting the formulation in Eq. (4.9) into QUBO. The integer variable z_u is expressed as a w-bit binary expansion

$$z_u = \sum_{k=1}^{w} 2^{k-1} x_{i(u,k)} \tag{4.12}$$

with $x_i \in \{0, 1\}$ and the composite index $i(u, k) = (u - 1)w + k$. The expected returns are then expressed as

$$r_u z_u = \sum_{k=1}^{w} 2^{k-1} r_u x_{i(u,k)} \tag{4.13}$$

while the allocation constraint becomes the penalty term

$$-\Big(\sum_{u=1}^{m}\sum_{k=1}^{w} 2^{k-1} p_w b x_{i(u,k)} - b\Big)^2 \tag{4.14}$$

A correlation threshold $c = 0$ is considered such that the correlation constraint becomes

$$\sum_{u,v}^{m} c_{u,v} z_u z_v = \sum_{u,v}^{m}\sum_{k,k'}^{w} 2^{k-1} 2^{k'-1} c_{u,v} x_{i(u,k)} x_{j(v,k')}. \tag{4.15}$$

Our formulation of Markowitz portfolio selection as an unconstrained optimization problem then becomes

$$\begin{aligned}\max_{x} \quad & \theta_1 \sum_{i}^{n} r_i x_i \\ & - \theta_2 (\sum_{i}^{n} 2^{k-1} b p_w x_i - b)^2 \\ & - \theta_3 \sum_{i,j}^{n} c_{i,j} x_i x_j \end{aligned} \tag{4.16}$$

where the problem size $n = mw$, $r_i = 2^{k-1} r_u$, $c_{i,j} = 2^{k-1} 2^{k'-1} c_{u,v}$, and θ_1, θ_2 and θ_3 are Lagrange multipliers used to weight each term for maximization or penalization. More information regarding the binary expansion technique used when generating Markowitz portfolio optimization see Appendix B.

4.2.1 Unconstrained Markowitz Formulation

The unconstrained portfolio optimization problem in Eq. (4.16) is formalized to QUBO form as

$$\min_{x} \quad \Big(\sum_{i}^{n} q_i x_i + \sum_{i,j}^{n} Q_{i,j} x_i x_j + \gamma \Big) \tag{4.17}$$

where q_i is the linear weight for the i^{th} spin, $Q_{i,j}$ is the quadratic weight for interactions between the i^{th} and j^{th} bits, and γ is a constant. Our definition of QUBO expresses optimization as minimization by switching the sign of Eq. (4.16) to be consistent with the use of QA to recover the lowest-energy state. The corresponding relationships with the original problem instance are given as

$$\begin{aligned} q_i &= -\theta_1 r_i - 2\theta_2 b^2 p_w \\ Q_{i,j} &= \theta_2 b^2 p_w^2 + \theta_3 c_{i,j} \\ \gamma &= \theta_2 b^2 \end{aligned} \tag{4.18}$$

Similarly, the quadratic binary form may be reduced to a classical Ising Hamiltonian

$$H(s) = \sum_{i} s_i h_i + \sum_{i,j} s_i s_j J_{ij} + \beta \tag{4.19}$$

where spin $s_i \in \{-1, 1\}$ is defined by $s_i = 2x_1 - 1$ with $s = (s_1, s_2, \ldots, s_n)$ while h_i is the spin weight, J_{ij} is the coupling strength, and β is a problem-specific constant. The parameters for the Ising Hamiltonian are given as

$$\begin{aligned} J_{i,j} &= \frac{1}{4} Q_{i,j} \\ h_i &= \frac{q_i}{2} + \sum_j J_{i,j} \\ \beta &= \frac{1}{4} \sum_{i,j} Q_{i,j} + \frac{1}{2} \sum_i q_i + \gamma \end{aligned} \tag{4.20}$$

The classical Ising formulation is then converted into a corresponding quantum Ising Hamiltonian given by Eq. (3.3) using the correspondence $s_i \to \sigma_i^z$.

4.2.2 Markowitz Data Generation

For purposes of benchmarking, an ensemble of problem instances was generated by sampling from uniform random price data with a seed of $b/5$. A random number was drawn as the initial price $a_{u,1}$ and every subsequent historical price point up to the purchasing price was -25% to $+25\%$ of the previous price $a_{u,l}$. The price range was set to be between $b/10$ and b with $N_f = 100$ historical price points per asset. Additionally, all $a_{u,l}$ are normalized by a_{u,N_f} to keep all asset prices in a similar range. Problem formulation and data generation methodology can be found in Appendix **??**

Problem controls are set to $\theta_1 = 0.3, \theta_2 = 0.5, \theta_3 = 0.2$ in the problem instances in which θ_2 was set higher to enforce the budget constraint. These weights were chosen after testing which combination stayed within budget and gave some diversity. By keeping θ_2 constant and increasing θ_3 while decreasing θ_1, an investor could increase the diversity relative to the potential returns and vice versa when decreasing θ_3 relative to θ_1. There are $1,000$ problems generated for each problem size with $m = 2, 3, 4, 5$ assets and $w = 4$ slices. In Fig. 9, the histogram of all $J_{i,j}$ coupler values and h_i linear qubit weight values is shown for the 1000 problems in each problem size n. Together the $J_{i,j}$ and k values compose the inter-chain and intra-chain coupler weights respectively as shown in Equation 2.10.

Consistently, the same clique embedding is used for all problems and all four problems sizes n as pictured in Figure 21. These graphs are embedded onto the D-Wave 2000Q processor and chain strength k is tuned to benchmark performance. All benchmarks are implemented and the $\tilde{p}_s$ is found by comparing the quantum annealing solutions to that of a brute force solver which finds the global minimum solution including any degeneracy.

Chapter 5

Quantum Annealing Benchmarking Results

5.1 Problem Controls

QA controls are benchmarked by evaluating their influence on the probability of success and probability of chain breaks across problem instances. I first characterize how problem parameters influence the baseline performance by estimating the probability of success for forward annealing using $T = 15\ \mu s$, $g = 0$, and a randomized embedding strategy. As shown in Figure 11, I compare $\tilde{p}_s$ for two cases of $w = 1$ and $w = 4$ across increasing n. The estimated probability of success for problems with $w = 4$ is consistently higher for problems with no slicing.

These results are explained by the energy spectra for the different problem parameters, which indicate sharp differences in the density of states. As shown in Figure 12, a typical problem instance with $w = 4$ has a much higher density of states than those with no slicing ($w = 1$). Intuitively, the single-slice behavior results from the specification that the price for each asset is proportional to budget, and, therefore, only a single asset may be selected without penalty when $w = 1$. However, the number of satisfying solutions v increases for

arbitrary w combinatorially, and, as shown in Appendix 5.4,

$$v = \frac{(2^{w-1} + m - 1)!}{(2^{w-1})!(w-2)!}. \tag{5.1}$$

Consequently, the probability to recover the lowest-energy state competes with these closely spaced, higher energy solutions, which leads to a corresponding decrease in the probability of success. For the remaining benchmark tests below, $w = 4$ was chosen as it represents a more challenging test for the quantum annealer as well as a greater interest to real-world financial applications.

5.2 Pre-processing Controls

5.2.1 Embedding Benchmarks

Embedding generates and places the physical spin chains for each logical spin on the QA hardware. The CMR and clique embedding algorithms described in Sec. 3.1.3 were evaluated by estimating the probability of success across problem sizes of $m = 8, 12, 16$, and 20 logical spins. For all problem instances of a same problem size, I use the same embedding because they require the same number of fully connected logical spins. The parameters of the embedded Ising Hamiltonian were set by scaling the inter-chain couplings $J_{i,j}$ to lie in the range $[-1, +1]$. Allll $J_{i,j}$ were scaled using a rescale factor of $\frac{1}{j_{max}}$, where j_{max} is the largest $J_{i,j}$ so all embedded $J_{i,j} = \frac{1}{j_{max}} J_{i,j}$. This scales all $J_{i,j}$ to be between 1 and -1. The intra-chain coupling strength is set to -1 to have a negative bias stronger than the $J_{i,j}$ values which range $-10^{-1} \leq J_{i,j} \leq 10^{-1}$ due to our data generation and normalization techniques.

The average chain length $\langle l_c \rangle$ from CMR and clique embedding methods grows with the number of logical spins n. The average is computed with respect to all chains in an embedding and plotted with respect to n in Figure 13. As expected by Eq. (2.13), the clique embedding method has a uniform chain length for each n. By contrast, the CMR method generates chains of variable length, as indicated by the the average chain length and variance shown in the plot.

From each of the embedding methods, the probability of success and probability of broken chains was estimated. As shown in Figure 14, very small differences were observed in both metrics with increasing problem size. From fitting the resulting point to an exponential, $\tilde{p}_s$ decays sub-exponentially with respect to n with rate -0.523 for the CMR embedding and rate -0.528 for the clique embedding. It is found that $\tilde{p}_b$ grows at a sub-exponential rate of 0.1824 for CMR embedding and 0.1656 for clique embedding as n increases. There is not a significant difference in the $\tilde{p}_s$ performance between CMR and clique embedding, but clique embedding requires a fewer number of spins as n increases and shows a slight improvement in $\tilde{p}_b$. Therefore, I chose to use clique embedding for subsequent benchmarks.

5.2.2 Spin Reversal Benchmarks

As discussed in Section 2.5.2, embedding algorithms map a logical spin to many spins to create chains. Coupling spins together via $J_{i,j}$ on the D-Wave hardware can cause small biases that are amplified by spin chains spread across the hardware. To mitigate these bias errors, spin reversal transforms were employed which simply translate spin-up to spin-down and vice versa for a random set of spins. This does not change the logical problem but is instead aimed at mitigating the bias error across a spin chain. The control used for conducting spin reversals is the number of transforms g to perform for a given job. There are $\frac{N_s}{g}$ samples per transform. Nominal improvements were observed in Figure 15 by using at least $g = 2$, with no advantage to using $g > 2$.

5.2.3 Chain Strength Benchmarks

As discussed in Section 2.5.2, embedding algorithms map a logical spin to many physical spins to create chains. The first error is an excitation of the system in which the measured sample solution is an excited state, and the second error excites only some physical qubits in the chain, which causes a broken chain in the sample solution. One or both of these errors can occur in a single sample. Increasing the chain strength k as shown in Equation 2.12, should reduce the probability of chains breaking, but increasing k too much can result in the intra-coupling strengths k overpowering the inter-chain coupling strengths $J_{i,j}$ and thus

increase the probability of chains breaking and the probability for the system to jump to an excited state. By sweeping over chain strengths $k = [0 \rightarrow -2]$ where the strength of k is characterize by it's magnitude, it is observed that $\tilde{p}_s$ decreases as problem size n increases for all k. This is visualized in Figure 16 where samples with broken chains are discarded and counted as incorrect solutions. As expected, the chains with no intra-chain strength at $k = 0$ have a $\tilde{p}_s = 0$ because there is no intra-chain coupling. It is then observed that the $\tilde{p}_s$ increases as chain strength increases until k reaches -0.5 or -1. For large k, a decrease in the $\tilde{p}_s$ is observed. Similarly, as n increases, the $\tilde{p}_s$ decreases for all k , but the optimal k varies by problem size. The reason for this can be found in Figure 9. Here it is observe that for $n = 8$ the optimal $k = -1.0$ with maximum $J_{i,j}$ value at 0.57 and for all other problem sizes the optimal $k = -0.5$ where the $J_{i,j}$ is below 0.3. As k becomes significantly stronger, than the $J_{i,j}$ strengths, the $\tilde{p}_s$ gradually decreases.

The type of errors that occur as k increases can be distinguished by analyzing the probability of a sample solution containing broken chain(s) $\tilde{p}_b$. This can be observed in Figure 17 where samples with weak k have a high $\tilde{p}_b$ and samples with strong k have a much lower $\tilde{p}_b$. In particular, the errors shaping the $\tilde{p}_s$ as observed in Figure 16 for $k = 0, -0.25$, and -0.5 are most evidently caused by the chains breaking. For small k, this is most evident in Figure 18 where the difference in average ratio of chains that break for samples between $k = 0$ and -0.25 reveals that fewer chains break per sample as k increases. For the case of $k = -0.5$ it is observed that $\tilde{p}_b$ decreases as n increases again due to $J_{i,j}$ being close to 0.5 in magnitude for smaller problem size. This demonstrates the likelihood for chains to break when k and $J_{i,j}$ are too close in magnitude. However, as k increases in magnitude above -0.5, it is observed that the $\tilde{p}_s$ decreases up to an order of magnitude. While $\tilde{p}_b$ does increase between $k = -1.5$ and -2, the $\tilde{p}_b$ for $k > -0.5$ is smaller than the p_b of $k = -0.5$ but the $\tilde{p}_s$ is also smaller. This shows that as k become too strong, the errors that reduce the $\tilde{p}_s$ are from the sample solution existing in an excited state. In accordance with the literature, I observe evidence of a "sweet spot" for k in Figure 16 where the $\tilde{p}_s$ for $k = -0.5$ is up to an order of magnitude higher than for stronger k at $n = 20$. However, the $\tilde{p}_b$ and $\tilde{r}_b$ is also one or more magnitudes higher for $k = -0.5$ than for stronger k. Dips in the $\tilde{p}_s$

are evident when k is too weak causing a high $\tilde{p}_b$ and when k is too strong causing a higher probability for the solution to exist in an excited state.

Although there is a clear link between the intra-chain strength k and the error rates impacting $\tilde{p}_s$, observing which chains break most frequently and where those chains break reveals other factors that play a role in error rates. This can be seen in Figure 19 which shows the probability for each qubit in a chain to be faulty for samples that have at least one broken chain. In accordance with the theory, the probability of chains breaking increases as the chain length increases with n. In addition, the qubits that are faulty with highest probability are always on the end of a chain which suggests that errors propagate in a chain starting from one of the ends. There is also a pattern in which chains break most frequently. This suggests that the error rates are tied to the embedding. There is a distinction in which chain indices break with higher probability (qubit chain index 0, 4, 8, 12, 16). If patterns in chain breaks can be spotted through the embedding, then better post-processing methods could be developed. The relative variance is high, ranging from 12% to 25% , but I assumed that this is a good benchmark.

Upon closer analysis, comparing Figure 19 with the embeddings in Figure 21, broken chains all follow a distinct pattern with clique embedding. In particular the chains with high $\tilde{p}_q$ (qubit chain index 0, 4, 8, 12, 16) utilize the top-most physical qubits across all unit cells. In addition, the 2 physical qubits which break with lowest probability in each of those chains are always coupled within the unit cell as apposed to across unit cells as seen in Figure 20. However, the placement on the hardware does not appear to have a strong impact where Figure 21 shows the placement of the four embeddings used for all problems. These results indicate that $\tilde{p}_q$ is strongly linked to the hardware embedding and which physical qubits are utilized on the unit cell but not which unit cells are utilized.

5.3 Annealing Controls

5.3.1 Forward Anneal Time Benchmarks

According to the adiabatic theorem, forward annealing more slowly should increase the probability of the system remaining in the ground state and thus increase the probability of success. The forward annealing time T is varied from 1 μs to 999 μs, which is the broadest range accessible on the D-Wave 2000Q. As shown in the upper panel of Figure 22, statistically insignificant changes in the probability of success are shown as annealing time increased at each problem size. Fitting the average probability of success with respect to problem size for the annealing time $T = 100$ μs yields a sub-exponential decay rate for $\tilde{p}_s$ given by -0.528 and a sub-exponential growth rate for $\tilde{p}_b$ given by 0.1628 as n increases. There is a statistically significant difference in the estimated probability of chain breaks $\tilde{p}_b$ with respect to forward annealing time, as shown in the lower panel of Figure 22. For $T = 100$ μs, a growth rate of 0.1656 is recovered for the probability of chain breaks with respect to problem size.

5.3.2 Reverse Annealing Benchmarks

From the reverse annealing controls listed in Sec. 3.1.3, I designed three experiments based on the e_i for the reverse annealing heuristic that include (i) starting in the known ground state e_0, (ii) starting in the known first excited state e_1, and (iii) starting in the lowest-energy state obtained from 1000 forward annealing samples e_f. I then swept over various schedules to find the optimal s_p with a range of $[0.1, 0.9]$ and t_p with a range of $[15 \rightarrow 800]\mu s$. The t_r and t_q parameters were set to be constant and symmetric at 5 μs each. Thus, the total anneal time is $T' = t_r + t_p + t_q$, where t_p is the time parameter that was chosen for analysis. For all experiments, 1000 samples were collected using the iterative reverse annealing schedule for 100 of the problems that were also sovled used in the forward annealing experiments. The probability of success for reverse annealing was estimated with respect to different choices for e_i, s_p, and t_p. I compared the combined heuristic of reverse annealing with forward annealing

with forward annealing alone with $\tilde{p}_s$, $\tilde{p}_b$, as well as the frequency of finding energies in excited states to forward annealing alone [1].

By setting e_i to the ground state, it was tested for parameters s_p and t_p that decrease $\tilde{p}_s$ when the quantum annealer is fed the correct solution. For this experiment, $\tilde{p}_s$ can be thought of as the probability of staying in e_0.

$$\tilde{p}_s(e_0 \rightarrow e_0) = p_f * \tilde{p}_s, \tag{5.2}$$

$$p_f * \tilde{p}_s = \frac{\sum_i^{N_p} \alpha_i}{N_p} * \frac{\sum_i^{N_p} \sum_j^{N_s} \delta_{ij}}{N_s}, \tag{5.3}$$

where p_f is the probability that forward annealing found the ground state, $\alpha_i \in \{0, 1\}$ indicates whether forward annealing was found the ground state for the i^{th} problem prior to reverse annealing, and $\delta_{ij} \in \{0, 1\}$ is a variable indicating whether the j^{th} sample of the i^{th} problem was measured to be the ground state with reverse annealing. By setting $e_i = e_1$, I tested whether reverse annealing enhances the probability to populate the ground state. For these tests, $\tilde{p}_s$ estimates the probability of moving from an excited state to the ground state.

$$\tilde{p}_s(e_e \rightarrow e_0) = (1 - p_f) * \tilde{p}_s, \tag{5.4}$$

$$(1 - p_f) * \tilde{p}_s = \frac{\sum_i^{N_p} (1 - \alpha_i)}{N_p} * \frac{\sum_i^{N_p} \sum_j^{N_s} \delta_{ij}}{N_s}. \tag{5.5}$$

In addition to testing reverse annealing at $e_i = e_0$ and e_1, reverse annealing was tested in combination with forward annealing for which $\tilde{p}_s$ estimates the cumulative probability of finding the correct solution state.

$$\tilde{p}_s(R) = \tilde{p}(e_0 \rightarrow e_0) + \tilde{p}(e_e \rightarrow e_0). \tag{5.6}$$

[1] After completing the majority of experiments on the D-Wave processor DW_2000Q_2_1, the remaining experiments were performed on D-Wave processor DW_2000Q_5. This included the parametric tests of reverse annealing with respect to s and t_p. Prior to testing, computational consistency was confirmed between the results generated using the first device and those using the second. The differences in $\tilde{p}_s$ and standard deviation were evaluated between the processors by comparing a previous reverse annealing experiment on the DW_2000Q_2_1 with the same experiment on the DW_2000Q_5. It was found that the same $\tilde{p}_s$ using both devices and a standard deviation that was within 10^{-5} of the measurements on the previous D-Wave processor.

For these experiments, I found it useful to primarily analyze $\tilde{p}_s(R) - \tilde{p}(e_0 \rightarrow e_0) = \tilde{p}(e_e \rightarrow e_0)$ to determine if reverse annealing improved upon the $\tilde{p}_s$ of forward annealing.

The results from setting $e_i = e_0$ for each problem with a problem size of $n = 20$ where $m = 5$ and $w = 4$ are shown in Figure 23. Because the computation begins in the correct solution state, this test measures the probability by which reverse annealing introduces errors into the correct solution. Ideally, $\tilde{p}_s$ will remain near unity for all s_p and t_p. It was observed that reverse annealing causes the system to leave the ground state with $\tilde{p}_s$ reducing to on the order of 10^{-5} by annealing back to at least $s = .6$ and increasing $t_p \geq 200\ \mu s$.

The results from setting $e_i = e_1$ with a problem size of $n = 20$ where $m = 5$ and $w = 4$ for each problem is shown in Figure 24. A maximal value of 4.8×10^{-4} for $\tilde{p}_s$ is found with parameters $s = 0.7$ and $t_p = 800\ \mu s$. This is is a $\tilde{p}_s$ one order of magnitude higher than what is observed with forward annealing. This suggests that if e_i is very close to e_0, there may be some benefit to choosing reverse annealing over forward annealing.

When solving optimization problems for applications in practice, the ground state and excited state will be unknown. However, one approach is to use reverse annealing in addition to forward annealing by using the lowest energy state found with 1000 forward annealing samples e_f as e_i for another 1000 samples of reverse annealing. The next experiment tests whether reverse annealing used in combination with forward annealing increases $\tilde{p}_s$ with a problem size of $n = 20$, where $m = 5$ and $w = 4$. The experimental results from setting $e_i = e_f$ are shown in Figure 25. These tests were constructed to determine when combining reverse annealing with forward annealing can improve upon forward annealing. Therefore, I removed the 6 problems forward annealing provided an $e_i = e_0$, and thus $\tilde{p}_s$ for this experiment is given by $\tilde{p}_s(R) - p(e_0 \rightarrow e_0)$ in this analysis. Similar to the previous experiment in Figure 24, the $\tilde{p}_s$ is at best on the order of 10^{-4} at parameters $s = .7$ and $t_p = 400\ \mu s$, which is one order of magnitude greater than the forward annealing experiments.

Figure 25 shows a potential for reverse annealing to improve upon results found with forward annealing in $\tilde{p}_s$. Therefore, for a set of 100 problems solved with reverse annealing and forward annealing, the $\tilde{p}_s$ of forward annealing (orange) alone, the $\tilde{p}_s$ of reverse annealing alone (blue), and the $\tilde{p}_s$ with a selection of either forward annealing or reverse annealing (green) was compared. If for a problem forward annealing at least one ground state was

found, the forward annealing $\tilde{p}_s$ was plotted for that problem (6 problems); otherwise, the reverse annealing $\tilde{p}_s$ was plotted (94 problems). The $\tilde{p}_s$ is measured over n ranging from $[8, 20]$. The reverse annealing parameters are set to have an $e_i = e_f$, $s = .7$, and $t_p = 400\ \mu s$. As shown in Figure 26, it was observed that when taking the combination of best results from forward annealing and reverse annealing with $e_i = e_f$, $\tilde{p}_s$ improves by an order of magnitude over forward annealing alone for $n = [16, 20]$ with a sub-exponential decay at a rate of $-.309$. Note that although the blue reverse annealing trend appears to perform the best, this trend is artificially inflated because 6 of the problems have $e_i = e_0$, which has been demonstrated in Figure 23 to yield a $\tilde{p}_s$ on the order of 10^{-2} at $s = .7$ and $t_p = 400\ \mu s$.

Next a histogram, as seen in Figure 27, visualizes all energies recorded from 1000 samples returned for a set of 94 problems where forward annealing did not find e_0 with $n = 20$. Forward annealing is compared with reverse annealing where $e_i = e_f$. We observe even for problems where neither reverse annealing or forward annealing found e_0, reverse annealing still on average finds a lower energy solution more often than forward annealing.

5.4 Combinations Constrained to the Budget

Assuming the optimal solution lies where the total value of assets bought equals the budget, the number of solutions which need to be checked is drastically reduced. If we have 1 asset, the only solution is buying the slice equal to 1. If we have 2 assets, the slice of the 2nd asset is dictated by whichever slice is chosen from the 1st asset. If the number of slices chosen is w, then we know that the slices correspond to $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, ..., \frac{1}{2^w}$. This gives a total of $2^w + 1$ (since we can also buy 0 for all slices) which are less than or equal to the budget. Mathematically, this can be expressed as

$$\#\text{ solutions} = \sum_{a_1=0}^{2^w} \sum_{a_2=0}^{2^w - a_1} 1 = 2^w + 1. \tag{5.7}$$

This is an equivalent problem to stating how many distinct terms are in the binomial $(a_1 + a_2)^{2^w}$.

Extending this to an arbitrary amount of assets (m), this equates to finding how many distinct terms are in the multinomial expansion $(a_1 + a_2 + ... + a_m)^{2^w}$, which can be found using the following equation

$$\#\text{ solutions} = \prod_{a=1}^{m-1} \frac{2^w + a}{a} = \frac{(2^w + m - 1)!}{(2^w)!(m-1)!}. \tag{5.8}$$

5.5 Post-processing Experiments

Post-processing strategies are used to determine a logical value for a broken chain instead of discarding the sample. In theory, majority vote can be a tool to improve $\tilde{p}_s$ for problems where broken chains still retain some representation of the logical qubit. However, if noise propagates through the chain of qubits or if k is too small for a chain to be a cohesive logical qubit, then majority vote is essentially a random selection. Figure 19 shows that chains that break with high probability also have half or more intra-chain qubits that break. Therefore, majority vote is ineffective for this configuration of embedding, chain strength, and J_{ij} value range. Although it appears in Figure 28 that the majority vote $\tilde{p}_s$ has increased significantly for k with a high $\tilde{p}_b$, the $\tilde{p}_s$ does not surpass the $\tilde{p}_s$ of a random selection $\frac{1}{2^n}$. The $\tilde{p}_b$ is only marginally lower than for $k = 0$; however, a $\tilde{p}_s$ that deviates from random selection is observed. Where $n = [8, 12]$, $k = -0.25$ delivers a $\tilde{p}_s$ an order of magnitude lower than random selection and at $n = 16$, a $\tilde{p}_s$ an order of magnitude higher than random selection. From this, it can be concluded that if the chain strength is too low, majority vote could deliver a $\tilde{p}_s$ lower than random selection due to strong bias errors deviating chains to a particular value. For $k = -0.5$, a $\tilde{p}_b$ is seen that is higher than for $k < -0.5$ but delivers a $\tilde{p}_s$ that is higher than for $k < -0.5$. However, majority vote does not improve over the results found with discard for this control or any $k \leq -0.5$. From these experiments, I concluded that majority vote does not improve $\tilde{p}_s$ for sufficiently strong k.

If there is a discernible pattern for the chain breaks based on a particular embedding algorithm, more intelligent post-processing methods can be utilized which filter out the qubits which are most likely to be faulty within a broken chain. Because there is a clear pattern in the error rates that can be observe for these problems with clique embedding, I

propose a novel post-processing strategy which performs a voted selection for each logical qubit where the choice is weighted according to the probability of each qubit q_l in the chain to be incorrect $\tilde{p}_q$. The logical choice $x_i \in [-1, 1]$ with the highest weight $W_i(x_i)$ is selected for the i^{th} broken logical qubit chain of the m^{th} sample. The formulation for the weight of each logical choice is given by

$$W_i(x_i) = (1 - \prod_{l=1}^{l_c}(\sigma_l * \tilde{p}_q^l + \sigma_l')) \prod_{l=1}^{l_c}(\sigma_l' * \tilde{p}_q^l + \sigma_l) \tag{5.9}$$

where $\sigma_m = \frac{(x_i q_l + 1)}{2}$ is the variable indicating if the l^{th} qubit corresponds to the logical choice x_i and $\sigma_l' = \frac{(-x_i q_l + 1)}{2}$ indicates if if the l^{th} qubit corresponds to the opposing logical choice $|1 - x_i|$. It is observed that this method improves the $\tilde{p}_s$ by one or more orders of magnitude for problems where there is a high p_b. In particular, when $k = 0.25$ there is a high $\tilde{p}_b$ but this post-processing strategy improves the $\tilde{p}_s$ by an order of magnitude for $n = 20$. This result is far better than a random selection for samples with high $\tilde{p}_b$ which demonstrates that there is still some part of the logical problem which survives when k is too weak. By contrast, when k is too large in magnitude compared to the $J_{i,j}$ values, all post-processing strategies tested were ineffective because the non-adiabatic dynamics limiting the $\tilde{p}_s$ most commonly causes the entire system to jump to an excited state instead of breaking the chain. This can be observed in Figure 29.

Chapter 6

Conclusions

By benchmarking the performance of tuning strategies for quantum annealing, this research establishes that there are methods for improving solution quality for a set of portfolio optimization problems. Furthermore, this research demonstrates a methodology for determining an optimal set of controls and proposes a novel method for post-processing solutions that contain errors. By analyzing the performance across a wide range of hardware controls, current performance of quantum annealing hardware is evident. These experiments cover a range of pre-processing, annealing, and post-processing controls to investigate the performance of QA with Markowitz portfolio optimization. This dissertation provides a methodology for choosing controls to optimize QA performance for future problems and applications. This research also sets a benchmark on the performance of the current state of the art in QA for problems such as portfolio optimization and also informs the QA community of the quantum annealer's capability for this application. By applying these results in a meta-analysis with other problem classes, the QA community could determine the overall effectiveness of various controls and their effects on $\tilde{p}_s$ and $\tilde{p}_b$ and inform future algorithm and hardware development for tuning controls. These goals can be accomplished in part with efforts to benchmark the solution quality when solving problems that are known to be difficult for classical computational methods and when adjusting the available tuning strategies of QA hardware.

By varying the problem controls such as n and w, an increase in the density of state is seen for problems with higher problem size and an increase in number of slices. This is

most likely associated with the number of viable solutions given by Eq. 5.1. As a result, I found that the average probability of success for problems with $w = 0$ is much higher than the problems with $w = 4$. Next, various forward annealing controls were tested including the embedding algorithm, T, and g. Clique embedding was measured against CMR, and it found that there was little difference between the two in $\tilde{p}_s$, but the CMR had slightly higher $\tilde{p}_b$ and a sub-exponential decay in $\tilde{p}_s$ of $-.528$ that is slightly higher than that of clique embedding which yielded $-.523$. Thus, I chose to continue future experiments with clique embedding with a design with fully connected problems such as Markowitz portfolio selection in mind. When varying $T = [1\mu s \rightarrow 999\mu s]$, it was found that $\tilde{p}_b$ was slightly higher for $T = [1\mu s, 5\mu s]$, but increasing anneal time had very little to no improvement on $\tilde{p}_s$. For this reason, I chose to continue all future forward annealing experiments using $T = 100\ \mu s$ where the exponential decay rate in p_s was $-.528$. When varying $g = [0, 10]$, small improvements were found in $\tilde{p}_s$ between $g = 0$ and $g = 2$, where the exponential decay rate became $-.505$ without much change from increasing the value of g further, and there was no consistent difference in $\tilde{p}_b$.

By sweeping over a range of chain strengths k, I was able to determine the optimal k by problem size n. The optimal k is strongly linked to the ratio of $J_{i,j}$ to k where if the k is at or below the $J_{i,j}$ range in magnitude, the $\tilde{p}_b$ greatly increases and the $\tilde{p}_s$ decreases due to chain break errors. Conversely, if k is too strong in magnitude as compared to $J_{i,j}$, the $\tilde{p}_b$ increases slightly and the $\tilde{p}_s$ can decrease by a full order of magnitude indicating that a k too strong will increase the probability for the sample solution to be an excited state. By analyzing the probability for each physical qubit in each chain to be faulty when a sample is broken $\tilde{p}_q$, I was able to visualize which chains break with highest probability and where they are most likely to break. It was found that there are certain chains that break most frequently which indicated that $\tilde{p}_q$ is strongly tied to the embedding, in this case clique embedding. The chains typically break from the edges of the chain and the qubits that are coupled within a unit cell were statistically that least likely to be faulty. Using this information, I present a novel post-processing strategy which assigns a weight probability selection based on the the $\tilde{p}_q$. With this post-processing method, the $\tilde{p}_s$ was improved by several orders of magnitude for weak k.

I then tested reverse annealing methods with parameters e_i, s, and t_p. I began by observing the results in $\tilde{p}_s$ and $\tilde{p}_b$ at $n = 20$. I consistently observed that $\tilde{p}_b$ was the same order of magnitude as with the forward annealing experiments and $\tilde{p}_b$ was consistently highest at $s = .8$. By setting $e_i = e_0$, it was observed that the $\tilde{p}_s$ decreased exponentially as s increased. By setting $e_i = e_1$, reverse annealing had a $\tilde{p}_s$ an order of magnitude higher than forward annealing, which allowed us to conclude that when e_i is close to the ground state, reverse annealing provides some advantage over forward annealing. Because in general the ground state won't be known for a problem, I developed a heuristic which sets $e_i = e_f$ where $\tilde{p}_s$ is an order of magnitude higher than using forward annealing alone. The $\tilde{p}_s$ is then plotted as a function of n to compare reverse annealing with $e_i = e_f$, $s = .7$, and $t_p = 400\ \mu s$ to forward annealing with clique embedding, $T = 100\ \mu s$, and $g = 0$ alone. In particular, I used the $\tilde{p}_s^{(k)}$ of forward annealing for the 6 problems where $e_i = e_0$ and the $\tilde{p}_s^{(k)}$ of reverse annealing for the 94 problems where $e_i \neq e_0$. Reverse annealing continued demonstrating an order of magnitude increase in p_s over forward annealing alone. Lastly, by creating an histogram which plots the lowest energies found across 1000 samples for the 94 problems where $e_i \neq e_0$, I found that reverse annealing($e_i = e_f$) on average finds lower energy solutions as compared with forward annealing.

Many prominent questions remain open about the expected physical and computational behavior of AQC, which includes clarification about how nonadiabatic effects impact the performance of QA devices as well as how the optimal run-time can be realized without prior knowledge of the underlying energy landscape [145, 108]. The design, development, and demonstration of reliable and programmable adiabatic quantum computers also remain an open endeavor. The idealized setting for satisfying the adiabatic condition exactly has yet to be realized in practice, and robust models for describing nonadiabatic effects will require better characterization about the underlying physical systems. Existing demonstrations that relax the adiabatic condition have made remarkable progress in controllability of QA, but experimental evidence remains mixed on the computational performance, largely because of the relatively small amount of data that can be processed by these devices, which is insufficient for best-in-class comparisons.

Further experimental investigations into computational scaling using larger-capacity devices will help identify the significance of nonadiabatic effects. However, the scaling of QA to arbitrarily large capacities will likely require methods for managing and correcting faults from noise in the devices and errors in the controls. The principles of fault-tolerant operation are well defined within the context of the circuit model of quantum computation, where redundantly encoded quantum states are actively corrected in the presence of noise. These principles may also apply to AQC, but a complete theory of fault-tolerant adiabatic quantum computation has yet to be developed.

Assuming the engineering of scalable quantum devices is achieved, QA/ AQC can be expected to have a significant effect on computational science. Already, the formal reduction of many combinatorial optimization problems to the Ising problem has made QA/AQC an attractive model for numerous known applications. However, AQC supports an even broader class of Hamiltonians, including those that are complete for BQP and QMA, and how this can be leveraged for new methods of quantum computation remains to be seen.

Among this wide range of open questions in the field of adiabatic quantum computation and QA, this research focuses on developing a benchmarking strategy for QA, using QA for combinatorial optimization, and understanding how tuning strategies/controls can be used to mitigate nonadiabtic dynamics and improve the quality of solutions. This dissertation involved developing a framework for solving problems using the D-Wave 2000Q. This framework takes any QUBO or Ising Hamiltonian, converts it to quantum Ising, prepares a set of controls to be applied to the D-Wave hardware or Hamiltonian, embeds the Hamiltonian onto the hardware, runs a certain number of samples with specified controls, receives the solution state back from the D-Wave, detects errors, interprets those results with post-processing, and benchmarks the solution states for quality against a brute force solver. A formulation was then developed for Markowitz portfolio optimization that can be solved using QA. This formulation was implemented with random data (modeled after the stock market) to investigate the performance of the quantum annealer and how QA controls affect performance according to our benchmarks.

Bibliography

[1] (2020). Top500 list - november 2019. 1

[2] Adachi, S. H. and Henderson, M. P. (2015a). Application of quantum annealing to training of deep neural networks. *arXiv preprint arXiv:1510.06356*. 33

[3] Adachi, S. H. and Henderson, M. P. (2015b). Application of quantum annealing to training of deep neural networks. *arXiv preprint arXiv:1510.06356*. 33

[4] Adami, C. and Cerf, N. J. (1999). Quantum computation with linear optics. In *Quantum Computing and Quantum Communications*, pages 391–401. Springer. 16

[5] Aharonov, D., Van Dam, W., Kempe, J., Landau, Z., Lloyd, S., and Regev, O. (2008). Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM review*, 50(4):755–787. 17

[6] Ajagekar, A., Humble, T., and You, F. (2020). Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems. *Computers & Chemical Engineering*, 132:106630. 40

[7] Albash, T. and Lidar, D. A. (2015). Decoherence in adiabatic quantum computation. *Physical Review A*, 91(6):062320. 30, 45

[8] Albash, T. and Lidar, D. A. (2018a). Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1):015002. 18, 21, 35

[9] Albash, T. and Lidar, D. A. (2018b). Demonstration of a scaling advantage for a quantum annealer over simulated annealing. *Physical Review X*, 8(3):031016. 31, 40

[10] Appenzeller, J., Lin, Y.-M., Knoch, J., and Avouris, P. (2004). Band-to-band tunneling in carbon nanotube field-effect transistors. *Physical review letters*, 93(19):196805. 3

[11] Association, S. I. et al. (2015). Rebooting the it revolution. In *based on Rebooting the IT Revolution Workshop, Washignton*. 1

[12] Bengtsson, I. and Życzkowski, K. (2017). *Geometry of quantum states: an introduction to quantum entanglement*. Cambridge university press. 14

[13] Bennett, C. H. (2003). Notes on landauer's principle, reversible computation, and maxwell's demon. *Studies In History and Philosophy of Science Part B: Studies In History and Philosophy of Modern Physics*, 34(3):501–510. 4

[14] Bernstein, E. and Vazirani, U. (1997). Quantum complexity theory. *SIAM Journal on computing*, 26(5):1411–1473. 15

[15] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., and Lloyd, S. (2017). Quantum machine learning. *Nature*, 549(7671):195. 36

[16] Bian, Z., Chudak, F., Israel, R. B., Lackey, B., Macready, W. G., and Roy, A. (2016). Mapping constrained optimization problems to quantum annealing with application to fault diagnosis. *Frontiers in ICT*, 3:14. 33

[17] Bian, Z., Chudak, F., Macready, W. G., Clark, L., and Gaitan, F. (2013). Experimental determination of ramsey numbers. *Phys. Rev. Lett.*, 111:130505. 33

[18] Biere, A., Heule, M., and van Maaren, H. (2009). *Handbook of satisfiability*, volume 185. IOS press. 35

[19] Bohr, M. (2007). A 30 year retrospective on dennard's mosfet scaling paper. *IEEE Solid-State Circuits Society Newsletter*, 12(1):11–13. 5

[20] Boixo, S., Rnnow, T. F., Isakov, S. V., Wang, Z., Wecker, D., Lidar, D. A., Martinis, J. M., and Troyer, M. (2014). Evidence for quantum annealing with more than one hundred qubits. *Nature Physics*, 10(3):218224. 30, 31

[21] Boixo, S., Smelyanskiy, V. N., Shabani, A., Isakov, S. V., Dykman, M., Denchev, V. S., Amin, M. H., Smirnov, A. Y., Mohseni, M., and Neven, H. (2016). Computational multiqubit tunnelling in programmable quantum annealers. *Nature communications*, 7:10327. 21

[22] Boothby, T., King, A. D., and Roy, A. (2016). Fast clique minor generation in chimera qubit connectivity graphs. *Quantum Information Processing*, 15(1):495–508. 27, 29

[23] Born, M. and Fock, V. (1928). Beweis des adiabatensatzes. *Zeitschrift für Physik*, 51(3):165–180. 17

[24] Bravyi, S., Divincenzo, D. P., Oliveira, R. I., and Terhal, B. M. (2006). The complexity of stoquastic local hamiltonian problems. *arXiv preprint quant-ph/0606140*. 19, 21

[25] Bunyk, P. I., Hoskinson, E. M., Johnson, M. W., Tolkacheva, E., Altomare, F., Berkley, A. J., Harris, R., Hilton, J. P., Lanting, T., Przybysz, A. J., et al. (2014a). Architectural considerations in the design of a superconducting quantum annealing processor. *IEEE Transactions on Applied Superconductivity*, 24(4):1–10. 22

[26] Bunyk, P. I., Hoskinson, E. M., Johnson, M. W., Tolkacheva, E., Altomare, F., Berkley, A. J., Harris, R., Hilton, J. P., Lanting, T., Przybysz, A. J., and et al. (2014b). Architectural considerations in the design of a superconducting quantum annealing processor. *IEEE Transactions on Applied Superconductivity*, 24(4):110. 43

[27] Cai, J., Macready, W. G., and Roy, A. (2014). A practical heuristic for finding graph minors. 27

[28] Cao, Y., Jiang, S., Perouli, D., and Kais, S. (2016). Solving set cover with pairs problem using quantum annealing. *Scientific reports*, 6:33957. 36

[29] Cesarone, F., Scozzari, A., and Tardella, F. (2014). Linear vs. quadratic portfolio selection models with hard real-world constraints. *Computational Management Science*, 12(3):345370. 50

[30] Chancellor, N. (2017). Modernizing quantum annealing using local searches. *New Journal of Physics*, 19(2):023024. 31

[31] Chapuis, G., Djidjev, H., Hahn, G., and Rizk, G. (2019). Finding maximum cliques on the d-wave quantum annealer. *Journal of Signal Processing Systems*, 91(3):363–377. 33

[32] Childs, A. M., Farhi, E., Goldstone, J., and Gutmann, S. (2000). Finding cliques by quantum adiabatic evolution. *arXiv preprint quant-ph/0012104*. 15

[33] Childs, A. M., Farhi, E., and Preskill, J. (2001). Robustness of adiabatic quantum computation. *Physical Review A*, 65(1):012322. 45

[34] Choi, V. (2008a). Minor-embedding in adiabatic quantum computation: I. the parameter setting problem. *Quantum Information Processing*, 7(5):193–209. 22, 23, 28

[35] Choi, V. (2008b). Minor-embedding in adiabatic quantum computation: I. the parameter setting problem. 26, 27

[36] Choi, V. (2010). Adiabatic quantum algorithms for the np-complete maximum-weight independent set, exact cover and 3sat problems. *arXiv preprint arXiv:1004.2226*. 15

[37] Choi, V. (2011). Minor-embedding in adiabatic quantum computation: Ii. minor-universal graph design. *Quantum Information Processing*, 10(3):343–353. 28

[38] Cook, S. A. (1971). The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM. 35

[39] Crispin, A. and Syrichas, A. (2013). Quantum annealing algorithm for vehicle scheduling. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3523–3528. IEEE. 33

[40] Cubitt, T. S., Perez-Garcia, D., and Wolf, M. M. (2015). Undecidability of the spectral gap. *Nature*, 528(7581):207. 18

[41] D-Wave Systems, I. Ocean tools library. https://docs.ocean.dwavesys.com/en/stable/. 12, 32, 39, 42, 43, 44, 89

[42] D-Wave Systems, I. Ocean tools library embedding documentation. https://docs.ocean.dwavesys.com/projects/system/en/latest/reference/embedding.html. 43

[43] Danowitz, A., Kelley, K., Mao, J., Stevenson, J. P., and Horowitz, M. (2012). Cpu db: recording microprocessor history. *Queue*, 10(4):10–27. 3

[44] De las Cuevas, G. and Cubitt, T. S. (2016). Simple universal models capture all classical spin physics. *Science*, 351(6278):1180–1183. 34

[45] Denchev, V. S., Boixo, S., Isakov, S. V., Ding, N., Babbush, R., Smelyanskiy, V., Martinis, J., and Neven, H. (2016). What is the computational value of finite-range tunneling? *Phys. Rev. X*, 6:031015. 21

[46] Deng, L., Yu, D., et al. (2014). Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387. 33

[47] Dennard, R. H., Gaensslen, F. H., Rideout, V. L., Bassous, E., and LeBlanc, A. R. (1974). Design of ion-implanted mosfet's with very small physical dimensions. *IEEE Journal of Solid-State Circuits*, 9(5):256–268. 5

[48] Deutsch, D. (1985). Quantum theory, the church–turing principle and the universal quantum computer. *Proc. R. Soc. Lond. A*, 400(1818):97–117. 15

[49] Devoret, M. H. and Martinis, J. M. (2005). Implementing qubits with superconducting integrated circuits. In *Experimental aspects of quantum computing*, pages 163–203. Springer. 16

[50] Dinneen, M. J., Mahasinghe, A., and Liu, K. (2019). Finding the chromatic sums of graphs using a d-wave quantum computer. *The Journal of Supercomputing*. 33

[51] DiVincenzo, D. P. (2000). The physical implementation of quantum computation. *Fortschritte der Physik: Progress of Physics*, 48(9-11):771–783. 15

[52] Djidjev, H. N., Chapuis, G., Hahn, G., and Rizk, G. (2018). Efficient combinatorial optimization using quantum annealing. *arXiv preprint arXiv:1801.08653*. 40

[53] Dridi, R. and Alghassi, H. (2017). Prime factorization using quantum annealing and computational algebraic geometry. *Scientific reports*, 7:43048. 33

[54] Dziarmaga, J. (2005). Dynamics of a quantum phase transition: Exact solution of the quantum ising model. *Physical Review Letters*, 95(24). 29

[55] Elgart, A. and Hagedorn, G. A. (2012). A note on the switching adiabatic theorem. *Journal of Mathematical Physics*, 53(10):102202. 18

[56] Farhi, E., Goldstone, J., Gosset, D., Gutmann, S., Meyer, H. B., and Shor, P. (2009). Quantum adiabatic algorithms, small gaps, and different paths. *arXiv preprint arXiv:0909.4766*. 25

[57] Farhi, E., Goldstone, J., Gutmann, S., and Sipser, M. (2000). Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*. 17, 19, 23, 25, 35, 40, 41, 45

[58] Feynman, R. P. (1982). Simulating physics with computers. *International journal of theoretical physics*, 21(6-7):467–488. 15, 37

[59] Flynn, L. J. (2004). Intel halts development of 2 new microprocessors. 4

[60] Fortnow, L. (2009). The status of the p versus np problem. *Communications of the ACM*, 52(9):78–86. 7

[61] Fu, Y. and Anderson, P. W. (1986). Application of statistical mechanics to np-complete problems in combinatorial optimisation. *Journal of Physics A: Mathematical and General*, 19(9):1605. 36

[62] Goodrich, T. D., Sullivan, B. D., and Humble, T. S. (2018). Optimizing adiabatic quantum program compilation using a graph-theoretic framework. *Quantum Information Processing*, 17(5):118. 23

[Grant and Humble] Grant, E. and Humble, T. Qa controls for portfolio optimization. https://code.ornl.gov/egy/qa-controls-for-portfolio-optimization/. Data is available upon request. 42, 89

[64] Hamerly, R., Inagaki, T., McMahon, P. L., Venturelli, D., Marandi, A., Onodera, T., Ng, E., Langrock, C., Inaba, K., Honjo, T., et al. (2018). Scaling advantages of all-to-all connectivity in physical annealers: the coherent ising machine vs. d-wave 2000q. 29

[65] Harris, R., Johansson, J., Berkley, A. J., Johnson, M. W., Lanting, T., Han, S., Bunyk, P., Ladizinsky, E., Oh, T., Perminov, I., Tolkacheva, E., Uchaikin, S., Chapple, E. M., Enderud, C., Rich, C., Thom, M., Wang, J., Wilson, B., and Rose, G. (2010).

Experimental demonstration of a robust and scalable flux qubit. *Phys. Rev. B*, 81:134510. 23

[66] Harris, R., Sato, Y., Berkley, A., Reis, M., Altomare, F., Amin, M., Boothby, K., Bunyk, P., Deng, C., Enderud, C., et al. (2018). Phase transitions in a programmable quantum spin glass simulator. *Science*, 361(6398):162–165. 33, 37

[67] Hauke, P., Katzgraber, H. G., Lechner, W., Nishimori, H., and Oliver, W. D. (2019). Perspectives of quantum annealing: Methods and implementations. *arXiv preprint arXiv:1903.06559*. 21

[68] Heim, B., Rønnow, T. F., Isakov, S. V., and Troyer, M. (2015). Quantum versus classical annealing of ising spin glasses. *Science*, 348(6231):215–217. 20

[69] Hen, I. and Spedalieri, F. M. (2016). Quantum annealing for constrained optimization. *Physical Review Applied*, 5(3):034007. 33

[70] Hinton, G. (2014). Boltzmann machines. *Encyclopedia of Machine Learning and Data Mining*, pages 1–7. 36

[71] Hogg, T. (2003). Adiabatic quantum computing for random satisfiability problems. *Physical Review A*, 67(2):022314. 15

[72] Horodecki, R., Horodecki, P., Horodecki, M., and Horodecki, K. (2009). Quantum entanglement. *Rev. Mod. Phys.*, 81:865–942. 14

[73] Huang, X. (2012). Mean–variance models for portfolio selection subject to experts estimations. *Expert Systems with Applications*, 39(5):5887–5893. 50

[74] Jaklevic, R., Lambe, J., Silver, A., and Mercereau, J. (1964). Quantum interference effects in josephson tunneling. *Physical Review Letters*, 12(7):159. 22

[75] Jansen, S., Ruskai, M.-B., and Seiler, R. (2007). Bounds for the adiabatic approximation with applications to quantum computation. *Journal of Mathematical Physics*, 48(10):102111. 18

[76] Jarret, M., Jordan, S. P., and Lackey, B. (2016). Adiabatic optimization versus diffusion monte carlo methods. *Physical Review A*, 94(4):042318. 40

[77] Jensen, T. R. and Toft, B. (2011). *Graph coloring problems*, volume 39. John Wiley & Sons. 36

[78] Jiang, S., Britt, K. A., McCaskey, A. J., Humble, T. S., and Kais, S. (2018). Quantum annealing for prime factorization. *Scientific Reports (Nature Publisher Group)*, 8:1–9. 33

[79] Johnson, M. W., Amin, M. H., Gildert, S., Lanting, T., Hamze, F., Dickson, N., Harris, R., Berkley, A. J., Johansson, J., Bunyk, P., et al. (2011a). Quantum annealing with manufactured spins. *Nature*, 473(7346):194. x, 22, 40, 98

[80] Johnson, M. W., Amin, M. H., Gildert, S., Lanting, T., Hamze, F., Dickson, N., Harris, R., Berkley, A. J., Johansson, J., Bunyk, P., et al. (2011b). Quantum annealing with manufactured spins. *Nature*, 473(7346):194. 22

[81] Jordan, S. P., Gosset, D., and Love, P. J. (2010). Quantum-merlin-arthur–complete problems for stoquastic hamiltonians and markov matrices. *Physical Review A*, 81(3):032331. 21

[82] Kadowaki, T. and Nishimori, H. (1998). Quantum annealing in the transverse ising model. *Physical Review E*, 58(5):5355. 20

[83] Kaeslin, H. (2008). *Digital integrated circuit design: from VLSI architectures to CMOS fabrication*. Cambridge University Press. 2

[84] Kalinin, K. P. and Berloff, N. G. (2018). Blockchain platform with proof-of-work based on analog hamiltonian optimisers. *arXiv preprint arXiv:1802.10091*. 33

[85] Kaplan, D. M. and White, C. G. (2003). *Hands-On Electronics: A practical introduction to analog and digital circuits*. Cambridge University Press. 2

[86] Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer. 35, 36

[87] Katzgraber, H. G., Hamze, F., and Andrist, R. S. (2014). Glassy chimeras could be blind to quantum speedup: Designing better benchmarks for quantum annealing machines. *Physical Review X*, 4(2):021008. 40

[88] Katzgraber, H. G., Hamze, F., Zhu, Z., Ochoa, A. J., and Munoz-Bauza, H. (2015). Seeking quantum speedup through spin glasses: The good, the bad, and the ugly. *Physical Review X*, 5(3):031026. 21

[89] Kempe, J., Kitaev, A., and Regev, O. (2006). The complexity of the local hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097. 19

[90] King, A. D., Carrasquilla, J., Ozfidan, I., Raymond, J., Andriyash, E., Berkley, A., Reis, M., Lanting, T. M., Harris, R., Poulin-Lamarre, G., et al. (2018a). Observation of topological phenomena in a programmable lattice of 1,800 qubits. *Nature*. 33, 37

[91] King, A. D., Carrasquilla, J., Raymond, J., Ozfidan, I., Andriyash, E., Berkley, A., Reis, M., Lanting, T., Harris, R., Altomare, F., and et al. (2018b). Observation of topological phenomena in a programmable lattice of 1,800 qubits. *Nature*, 560(7719). 32

[92] King, A. D. and McGeoch, C. C. (2014). Algorithm engineering for a quantum annealing platform. *arXiv preprint arXiv:1410.2628*. 25, 26, 29, 30, 31, 32, 33, 44

[93] King, J., Yarkoni, S., Nevisi, M. M., Hilton, J. P., and McGeoch, C. C. (2015). Benchmarking a quantum annealing processor with the time-to-target metric. *arXiv preprint arXiv:1508.05087*. 21, 40

[94] Klymko, C., Sullivan, B. D., and Humble, T. S. (2012). Adiabatic quantum programming: Minor embedding with hard faults. 27

[95] Klymko, C., Sullivan, B. D., and Humble, T. S. (2014). Adiabatic quantum programming: minor embedding with hard faults. *Quantum information processing*, 13(3):709–729. 23

[96] Krentel, M. W. (1986). The complexity of optimization problems. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 69–76. 40

[97] Kumar, S. (2015). Fundamental limits to moore's law. *arXiv preprint arXiv:1511.05956*. 3

[98] Lanting, T., Przybysz, A. J., Smirnov, A. Y., Spedalieri, F. M., Amin, M. H., Berkley, A. J., Harris, R., Altomare, F., Boixo, S., Bunyk, P., et al. (2014). Entanglement in a quantum annealing processor. *Physical Review X*, 4(2):021041. 40

[99] Lawler, E. L. (1985). The traveling salesman problem: a guided tour of combinatorial optimization. *Wiley-Interscience Series in Discrete Mathematics*. 36

[100] Lenstra, J. K. and Kan, A. R. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227. 15

[101] Levin, L. A. (1973). Universal sequential search problems. *Problemy Peredachi Informatsii*, 9(3):115–116. 35

[102] Liu, J., Spedalieri, F., Yao, K.-T., Potok, T., Schuman, C., Young, S., Patton, R., Rose, G., and Chamka, G. (2018). Adiabatic quantum computation applied to deep learning networks. *Entropy*, 20(5):380. 36

[103] Lloyd, S., Mohseni, M., and Rebentrost, P. (2013). Quantum algorithms for supervised and unsupervised machine learning. *arXiv preprint arXiv:1307.0411*. 36

[104] Lucas, A. (2014a). Ising formulations of many NP problems. *Frontiers in Physics*, 2:5. 15, 17, 19, 34, 36

[105] Lucas, A. (2014b). Ising formulations of many np problems. *Frontiers in Physics*, 2:5. 40

[106] Markov, I. L. (2014). Limits on fundamental limits to computation. *Nature*, 512(7513):147–154. 4, 5

[107] Markowitz, H. (1952). Portfolio selection. *The journal of finance*, 7(1):77–91. 49

[108] Marshall, J., Venturelli, D., Hen, I., and Rieffel, E. G. (2019). Power of pausing: Advancing understanding of thermalization in experimental quantum annealers. *Physical Review Applied*, 11(4):044083. 32, 38, 68

[109] Martin, I. (2017). What is the expected return on the market? *The Quarterly Journal of Economics*, 132(1):367–433. 50

[110] Martinis, J. M., Nam, S., Aumentado, J., Lang, K., and Urbina, C. (2003). Decoherence of a superconducting qubit due to bias noise. *Physical Review B*, 67(9):094510. 25

[111] Martoňák, R., Santoro, G. E., and Tosatti, E. (2004a). Quantum annealing of the traveling-salesman problem. *Physical Review E*, 70(5):057701. 36

[112] Martoňák, R., Santoro, G. E., and Tosatti, E. (2004b). Quantum annealing of the traveling-salesman problem. *Physical Review E*, 70(5):057701. 40

[113] Marzec, M. (2016). Portfolio optimization: applications in quantum computing. *Handbook of High-Frequency Trading and Modeling in Finance (John Wiley & Sons, Inc., 2016) pp*, pages 73–106. 33, 50

[Mass] Mass, S. S. I. F. Production of system-on-chip with 10-nanometer finfet technology. 2

[115] McGeoch, C. C. and Wang, C. (2013). Experimental evaluation of an adiabatic quantum system for combinatorial optimization. In *Proceedings of the ACM International Conference on Computing Frontiers*, page 23. ACM. 33

[116] Montanaro, A. (2016). Quantum algorithms: an overview. *npj Quantum Information*, 2:15023. 15

[117] Moore, G. E. (1998). Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86(1):82–85. 2

[118] Morita, S. and Nishimori, H. (2008). Mathematical foundation of quantum annealing. *Journal of Mathematical Physics*, 49(12):125210. 40

[119] Mott, A., Job, J., Vlimant, J.-R., Lidar, D., and Spiropulu, M. (2017). Solving a higgs optimization problem with quantum annealing for machine learning. *Nature*, 550(7676):375. 33

[120] Neukart, F., Compostella, G., Seidel, C., von Dollen, D., Yarkoni, S., and Parney, B. (2017a). Traffic flow optimization using a quantum annealer. *Frontiers in ICT*, 4:29. 33

[121] Neukart, F., Compostella, G., Seidel, C., Von Dollen, D., Yarkoni, S., and Parney, B. (2017b). Traffic flow optimization using a quantum annealer. *Frontiers in ICT*, 4:29. 35, 40

[122] Neven, H., Rose, G., and Macready, W. G. (2008). Image recognition with an adiabatic quantum computer i. mapping to quadratic unconstrained binary optimization. *arXiv preprint arXiv:0804.4457*. 33

[123] Nielsen, M. A. and Chuang, I. (2002). Quantum computation and quantum information. 7

[124] Nielsen, M. A. and Chuang, I. L. (2010). *Quantum Computation and Quantum Information*. Cambridge University Press, 2nd edition. 14, 15, 16, 24

[125] Nishimori, H. and Takada, K. (2017). Exponential enhancement of the efficiency of quantum annealing by non-stoquastic hamiltonians. *Frontiers in ICT*, 4:2. 21

[126] Novikov, S., Hinkey, R., Disseler, S., Basham, J. I., Albash, T., Risinger, A., Ferguson, D., Lidar, D. A., and Zick, K. M. (2018). Exploring more-coherent quantum annealing. In *2018 IEEE International Conference on Rebooting Computing (ICRC)*, pages 1–7. IEEE. 25

[127] Ohkuwa, M., Nishimori, H., and Lidar, D. A. (2018). Reverse annealing for the fully connected p -spin model. *Physical Review A*, 98(2). 32

[128] O'Malley, D., Vesselinov, V. V., Alexandrov, B. S., and Alexandrov, L. B. (2017). Nonnegative/binary matrix factorization with a d-wave quantum annealer. *arXiv preprint arXiv:1704.01605*. 36

[129] OMalley, D., Vesselinov, V. V., Alexandrov, B. S., and Alexandrov, L. B. (2018). Nonnegative/binary matrix factorization with a d-wave quantum annealer. *PloS one*, 13(12). 40

[130] Packan, P. A. (1999). Pushing the limits. *Science*, 285(5436):2079–2081. 3

[131] Pardalos, P. M. and Rosen, J. B. (1987). *Constrained global optimization: algorithms and applications*, volume 268. Springer. 39

[132] Passarelli, G., Yip, K.-W., Lidar, D. A., Nishimori, H., and Lucignano, P. (2020). Reverse quantum annealing of the p -spin model with relaxation. *Physical Review A*, 101(2). 41

[133] Pearson, A., Mishra, A., Hen, I., and Lidar, D. (2019). Analog errors in quantum annealing: Doom and hope. 25

[134] Pelofske, E., Hahn, G., and Djidjev, H. (2019). Optimizing the spin reversal transform on the d-wave 2000q. In *2019 IEEE International Conference on Rebooting Computing (ICRC)*, pages 1–8. IEEE. 30

[135] Pendry, J. (1983). Quantum limits to the flow of information and entropy. *Journal of Physics A: Mathematical and General*, 16(10):2161. 4

[136] Perdomo-Ortiz, A., Dickson, N., Drew-Brook, M., Rose, G., and Aspuru-Guzik, A. (2012). Finding low-energy conformations of lattice protein models by quantum annealing. *Scientific reports*, 2:571. 33

[137] Perdomo-Ortiz, A., Venegas-Andraca, S. E., and Aspuru-Guzik, A. (2011). A study of heuristic guesses for adiabatic quantum computation. *Quantum Information Processing*, 10(1):33–52. 25, 31

[138] Potok, T. E., Schuman, C., Young, S., Patton, R., Spedalieri, F., Liu, J., Yao, K.-T., Rose, G., and Chakma, G. (2018). A study of complex deep learning networks on high-performance, neuromorphic, and quantum computers. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 14(2):19. 33, 36

[139] Powell, J. R. (2008). The quantum limit to moore's law. *Proceedings of the IEEE*, 96(8):1247–1248. 3

[140] Preskill, J. (2018). Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79. 16

[141] Pudenz, K. L. (2016). Parameter setting for quantum annealers. In *2016 IEEE high performance extreme computing conference (HPEC)*, pages 1–6. IEEE. 23, 30

[142] Pudenz, K. L., Albash, T., and Lidar, D. A. (2014). Error-corrected quantum annealing with hundreds of qubits. *Nature communications*, 5(1):1–10. 30, 45

[143] Pugh, E. W. (1995). *Building IBM: shaping an industry and its technology*. MIT press. 1

[144] Quintana, C., Chen, Y., Sank, D., Petukhov, A., White, T., Kafri, D., Chiaro, B., Megrant, A., Barends, R., Campbell, B., et al. (2017). Observation of classical-quantum crossover of 1/f flux noise and its paramagnetic temperature dependence. *Physical review letters*, 118(5):057702. 23

[145] Quiroz, G. (2019). Robust quantum control for adiabatic quantum computation. *Physical Review A*, 99(6):062306. 38, 68

[146] Rado, T. (1962). On non-computable functions. *Bell System Technical Journal*, 41(3):877–884. 6

[147] Raymond, J., Ndiaye, N., Rayaprolu, G., and King, A. (2020). Improving performance of logical qubits by parameter tuning and topology compensation. *arXiv preprint arXiv:2006.04913*. 29

[148] Roland, J. and Cerf, N. J. (2002). Quantum search by local adiabatic evolution. *Physical Review A*, 65(4):042308. x, 12, 19, 24, 99

[149] Rønnow, T. F., Wang, Z., Job, J., Boixo, S., Isakov, S. V., Wecker, D., Martinis, J. M., Lidar, D. A., and Troyer, M. (2014). Defining and detecting quantum speedup. *Science*, 345(6195):420–424. 30, 31

[150] Rosenberg, G., Haghnegahdar, P., Goddard, P., Carr, P., Wu, K., and De Prado, M. L. (2016). Solving the optimal trading trajectory problem using a quantum annealer. *IEEE Journal of Selected Topics in Signal Processing*, 10(6):1053–1060. 33, 35, 50

[151] S Humble, T., J McCaskey, A., S Bennink, R., J Billings, J., F DAzevedo, E., D Sullivan, B., F Klymko, C., and Seddiqi, H. (2014). An integrated programming and development environment for adiabatic quantum optimization. *Computational Science Discovery*, 7(1):015006. 28

[152] Sandberg, A. (1999). The physics of information processing superobjects: daily life among the jupiter brains. *Journal of Evolution and Technology*, 5(1):1–34. 3, 4

[153] Santoro, G. E. and Tosatti, E. (2006). Optimization using quantum mechanics: quantum annealing through adiabatic evolution. *Journal of Physics A: Mathematical and General*, 39(36):R393. 33

[Schilling] Schilling, A. From 2022 in 3 nm: Samsung's multi bridge channel fet as the future of semiconductor technology. 2

[155] Schrock, J., McCaskey, A., Hamilton, K., Humble, T., and Imam, N. (2017). Recall performance for content-addressable memory using adiabatic quantum optimization. *Entropy*, 19(9):500. 33

[156] Sipser, M. (1996). Introduction to the theory of computation. *ACM Sigact News*, 27(1):27–29. 6

[157] Stollenwerk, T., OGorman, B., Venturelli, D., Mandrà, S., Rodionova, O., Ng, H., Sridhar, B., Rieffel, E. G., and Biswas, R. (2019). Quantum annealing applied to de-conflicting optimal trajectories for air traffic management. *IEEE transactions on intelligent transportation systems*, 21(1):285–297. 40

[158] Struck, P. R. and Burkard, G. (2016). *Spin Quantum Computing*, pages 71–103. Springer Netherlands, Dordrecht. 16

[159] Tichy, W. (2017). Is quantum computing for real? an interview with catherine mcgeoch of d-wave systems. *Ubiquity*, 2017(July):1–20. 43

[160] Titiloye, O. and Crispin, A. (2011). Quantum annealing of the graph coloring problem. *Discrete Optimization*, 8(2):376–384. 36

[161] Tsai, J.-F., Carlsson, J. G., Ge, D., Hu, Y.-C., and Shi, J. (2014). Optimization theory, methods, and applications in engineering 2013. *Mathematical Problems in Engineering*, 2014. 39

[162] (US), E. I. A. and Office, G. P. (2016). *International Energy Outlook 2016, with Projections to 2040*. Government Printing Office. 5

[163] Ushijima-Mwesigwa, H., Negre, C. F. A., and Mniszewski, S. M. (2017). Graph partitioning using quantum annealing on the d-wave system. In *Proceedings of the Second International Workshop on Post Moores Era Supercomputing*, PMES'17, pages 22–29, New York, NY, USA. ACM. 33

[164] van der Ploeg, S. H. W., Izmalkov, A., Grajcar, M., Hubner, U., Linzen, S., Uchaikin, S., Wagner, T., Smirnov, A. Y., van den Brink, A. M., Amin, M. H. S., and et al. (2007). Adiabatic quantum computation with flux qubits, first experimental results. *IEEE Transactions on Applied Superconductivity*, 17(2):113119. 40

[165] Venturelli, D. and Kondratyev, A. (2019). Reverse quantum annealing approach to portfolio optimization problems. *Quantum Machine Intelligence*, 1(1-2):17–30. 32, 35, 50

[166] Venturelli, D., Mandra, S., Knysh, S., OGorman, B., Biswas, R., and Smelyanskiy, V. (2015). Quantum optimization of fully connected spin glasses. *Physical Review X*, 5(3):031040. 23, 27, 29, 33

[167] Vinci, W., Albash, T., Paz-Silva, G., Hen, I., and Lidar, D. A. (2015). Quantum annealing correction with minor embedding. *Physical Review A*, 92(4):042310. 25, 26

[168] Weigel, M., Katzgraber, H. G., Machta, J., Hamze, F., and Andrist, R. S. (2015). Glassy chimeras could be blind to quantum speedup: Designing better benchmarks for quantum annealing machines. *PHYSICAL REVIEW X*, 5(1). 21

[169] Weik, M. H. (1961). The eniac story. *ORDNANCE, January-February*, 1. 1

[170] Wendin, G. (2017). Quantum information processing with superconducting circuits: a review. *Reports on Progress in Physics*, 80(10):106001. 22

[171] Whitepaper, D.-W. T. (2018). Reverse quantum annealing for local refinement of solutions. https://www.dwavesys.com/sites/default/files/14-1018A-A_Reverse_Quantum_Annealing_for_Local_Refinement_of_Solutions.pdf. 32

[172] Wieschebrink, C. (2006). Two np-complete problems in coding theory with an application in code based cryptography. In *2006 IEEE International Symposium on Information Theory*, pages 1733–1737. IEEE. 15

[173] Xia, R., Bian, T., and Kais, S. (2017). Electronic structure calculations and the ising hamiltonian. *The Journal of Physical Chemistry B*, 122(13):3384–3395. 33

[174] Yamashiro, Y., Ohkuwa, M., Nishimori, H., and Lidar, D. A. (2019). Dynamics of reverse annealing for the fully connected p -spin model. *Physical Review A*, 100(5). 26, 41

[175] Young, K. C., Blume-Kohout, R., and Lidar, D. A. (2013). Adiabatic quantum optimization with the wrong hamiltonian. *Physical Review A*, 88(6). 27, 29

[176] Zener, C. (1932). Non-adiabatic crossing of energy levels. *Proc. R. Soc. Lond. A*, 137(833):696–702. 18

[177] Zhang, J., Li, F.-g., Xie, Y., Wu, C.-w., Ou, B.-q., Wu, W., and Chen, P.-x. (2018). Realizing an adiabatic quantum search algorithm with shortcuts to adiabaticity in an ion-trap system. *Physical Review A*, 98(5):052323. 22

[178] Zhu, Z., Ochoa, A. J., Schnabel, S., Hamze, F., and Katzgraber, H. G. (2016). Best-case performance of quantum annealers on native spin-glass benchmarks: How chaos can affect success probabilities. *Physical Review A*, 93(1):012317. 40

Appendices

A Solving QUBO problems on the D-Wave 2000Q

The following sections display the python 3 code that I used to develop the framework for solving optimization problems including the Markowitz portfolio optimization problem on the D-Wave 2000Q quantum annealer with their ocean tools [41]. The entire python project can also be view in my git repository [Grant and Humble].

A.1 Framework for Solving QUBO on the Quantum Annealer

A framework to solve QUBO problems on the D-Wave 2000Q quantum annealer was built using the ocean tools available in Python 3.5. This project started with the construction of a Python class that takes a QUBO problem, converts it to a quantum Ising Hamiltonian, embeds that Ising Hamiltonian onto the D-Wave hardware, anneals for a specified set of controls, unembeds the problem, and retrieves the solution state.

Listing 1: Framework for solving QUBO as quantum Ising with the D-Wave 2000Q in Python 3. The inputs include the QUBO matrix (*qubo*), number of logical spins (n), number of samples (*num_samples*), number of spin reversal transforms (*spin*), chain strength (*chain_strength*), annealing schedule (*anneal*), and *embedding*. We also need the D-Wave *url*, *token*, and *solver* name for remote access. Additionally, for reverse annealing, we provide the initial state (*initial*).

```
import numpy as np
from dwave.cloud import Client
import dwave
import dwave.embedding
import dimod
from dwave.system.samplers import DWaveSampler

# Create remote connection
conn = Client(endpoint=url, token=token, solver=solver, proxy=None,
    ↪ permissive_ssl=False, request_timeout=60, polling_timeout=None,
    ↪ connection_close=False, headers=None)
# Set parameters
```

```
self.params = {"answer_mode": "raw", "auto_scale": True, "postprocess": "", "
    ↪ num_reads": num_samples, "num_spin_reversal_transforms": spin_reversal,
    ↪ "anneal_schedule": anneal_schedule, "initial_state": initial_state, "
    ↪ reinitialize_state": reinitialize}
# Convert QUBO matrix into a QUBO dictionary
qubo_dict = {}
qubo_dict.update({(i, j): qubo[i][j] for i in range(n) for j in range(n)})
# Get the solver
solver = DWaveSampler(endpoint=url, self.token, solver=solver)
# Set the range of the Ising h and J values.
h_range = [-2, 2]
J_range = [-1, 1]

# convert qubo to ising and retrieve the h, J, and constant (ising_offset).
(h, J, ising_offset) = dimod.utilities.qubo_to_ising(qubo_dict)

# Normalize h and J to be $-1 \leq h,J \leq 1$.
h_max = max(list(map(abs,h.values())))
j_max = max([abs(x) for x in list(J.values())])
if h_max > j_max:
    j_max = self.h_max
rescale = j_scale / j_max
h1 = dict((k, (v * rescale)) for k, v in h.items())
if len(list(J.values())) > 0:
    J1 = {key: rescale*val for key, val in list(J.items())}
else:
    J1 = J

# Embed the Ising Hamiltonian onto the D-Wave hardware
Adjacency = self.solver.adjacency  #hardware adjacency
[h0, j0] = dwave.embedding.embed_ising(h1, J1, embedding, Adjacency,
    ↪ chain_strength)

# Solve Ising Hamiltonian on the D-Wave 2000Q
samples = solver.sample_ising(h0, j0, **self.params)
```

```
#Unembed samples with post-processing method (Majority Vote shown here)
ising_solutions, idx = dwave.embedding.majority_vote(samples, embedding)
qubo_solutions = np.zeros((len(ising_solutions), len(ising_solutions[0])))
for z in range(len(ising_solutions)):
    qubo_solutions[z] = [int(x + 1) / 2 for x in qubo_solutions[z]]

# Find the Ising energies and convert to the QUBO energies
ising_energies = np.zeros(len(ising_solutions))
qubo_energies = np.zeros(len(qubo_solutions))
qubo_arrays = np.zeros(n, len(qubo_solutions))
for i in range(len(qubo_solutions)):
    print('h vals', list(h1.values()))
    solution = ising_solutions[i]
    h_energy = sum(list(h1.values())[v] * val for v, val in enumerate(solution
    ↪ ))
    J_energy = sum(J1[(u, v)] * solution[u] * solution[v] for u, v in J1)
    ising_energies[i] = (self.h_energy + self.J_energy)
    qubo_energies[i] = ((ising_energies[i] / rescale) + ising_offset)

return qubo_solutions
return qubo_energies
```

A.2 Generating Data for Markowitz Portfolio Optimization

Listing 2: Methodology for generating random price data modeled after stock market in Python 3.

```
import numpy as np
import random

price_data = np.zeros((num_prices, num_assets))
price = random.Random()
price.seed(budget / 5)
percent = random.Random()
percent.seed(.15)
for i in range(assets):
```

```
        starting_price = price.uniform(budget/10, budget)
        for j in range(num_prices * problem_number):
            percent_change = percent.uniform(-.25, .25)
            price_data[j, i] = starting_price + starting_price*percent_change

return price_data
```

A.3 Implementation of the Binary Slicing

Listing 3: Algorithm for normalizing the price data to represent fractions of the budget and then expanding the price data in slices according to the binary slicing method in Python 3.

```
import numpy as np

# The binary slicing series expanded to the number of slices specified.
slice_list = np.zeros(num_slices)
for i in range(0, num_slices):
    slice_list[i] = float(1) / float(2 ** i)
# Expand and normalize the raw price data (rpd_expanded) where each row is the
# normalized price point and each column is a sliced asset.
self.rpd_expanded = np.zeros(num_prices, num_assets * num_slices)
l = 0
for i in range(num_assets):
    # The new asset prices will be a matrix where each column represents a
    ↪ slice
    # and each row is the normalized price.
    new_asset_prices = np.zeros((num_prices, num_slices))
    # Normalize the price data for each asset to the budget so that the
    # purchasing price is 1.
    norm_price_factor = budget / raw_price_data[num_prices - 1, i]
    # Create new asset prices for each slice according to the binary slicing
    # series.
    for j in range(num_slices):
        for k in range(num_prices):
            new_asset_prices[k, j] = rpd[k, i] * slice_list[j] *
            ↪ norm_price_factor
```

```
    # Expand the raw price data to include the new normalized and sliced price
    # data arrays for each asset.
    rpd_expanded[num_prices, l:l+num_slices] = new_asset_prices
    l = l + num_slices

return rpd_expanded
```

A.4 Building Portfolio Optimization QUBO

Listing 4: Methodology for generating QUBO from the Markowitz formulation in Python 3.

```
import numpy as np

exp_returns = np.zeros(num_assets) # expected returns
# Generate expected returns using a simple moving average.
for i in range(num_assets):
    for j in range(num_prices - 1):
        historical_returns[j, i] = price_data[j + 1, i] - price_data[j, i]
    exp_returns[i] = np.mean(historical returns[:, i])

# Calculate the covariance matrix.
qubo_covariance = np.zeros((num_assets, num_assets))
 for i in range(num_cols):
    for j in range(num_cols):
        qubo_covariance[i, j] = ((price_data[num_prices - 1, i] -  np.mean(
            ↪ price_data[:, i])) * (price_data[num_rows - 1, j] -    np.mean(
            ↪ price_data[:, j]))) / (num_assets - 1)

# The purchasing price for each asset.
prices = self.price_data[self.num_rows - 1, :].tolist()
# A diagonal matrix with the vector of expected returns
QUBO_returns = np.diag(expected_returns)
# The linear/ diagonal matrix with price penalties according to the budget(b).
QUBO_prices_linear = np.diag([x * (2 * b) for x in prices])
# The quadratic/ off-diagonal portion of the price penalty term.
QUBO_prices_quadratic = np.outer(prices, prices)
```

```
# The linear QUBO term
qi = -(theta_one * QUBO_returns) - (theta_two * QUBO_prices_linear)
# the quadratic QUBO term
qij = (theta_two * QUBO_prices_quadratic) + (theta_three * QUBO_covariance)

# Build the nxn QUBO matrix where n is the number of assets
qubo = qi + qij
# Form QUBO as an upper triangular nxn matrix
for col in range(0, n-1):
    for row in range(col+1, self.n):
        qubo[row, col] = 0
for row in range(0, n-1):
    for col in range(row+1, n):
        qubo[row, col] = 2 * self.qubo[row, col]

return qubo
```

B Mathematical Derivations

B.1 Binary Slicing Expansion

Slicing allows purchasing an asset with a percentage of the budget instead of allocating the budget toward a single asset. If a naive approach is used for slicing, where, for example, the slices include buying an asset with 25%, 50%, and/or 75% of the budget, the same effective problem will be considered multiple times (i.e., $25\% + 50\% = 75\%$). The maximum number of assets that can be bought will be $\frac{w*(w-1)}{2}$. Therefore, to consider a single asset with up to $\frac{w*(w-1)}{2}$ assets being bought, ($\frac{w*(w-1)}{2} \approx \frac{(N+1)^2}{2}$ for large N). The total considered combinations is 2^{w-1}.

B.2 Number of Combinations Constrained to the Budget

The binary slicing method was developed to eliminate redundancy in the solution space because all solutions are considered at once on a quantum computer. If only powers of 2 are considered (because each "asset" can either be bought or not bought) slides would include buying 1 asset, 50% asset, 25% asset. . . down to $\frac{1}{2^N}$ assets. This expansion eliminates all redundancy and allows a maximum of $2^w - 1$ slices of a single asset to be bought ($2^w - 1 \approx 2^{N+1}$ for large w). Because the total number of considered combinations remains the same, a dramatic increase in diversity of the search space is achieved at no additional expense. Thus, the following binary slicing series is used $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}..\frac{1}{2^{w-1}}$ where each value is a fraction of the budget. Assuming the optimal solution lies where the total value of assets bought equals the budget, the number of solutions that need to be checked is drastically reduced. If we have 1 asset, the only solution is buying the slice equal to 1. If we have 2 assets, the slice of the second asset is dictated by whichever slice is chosen from the first asset. If the number of slices chosen is w, we know that the slices correspond to $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}..\frac{1}{2^{w-1}}$, which gives a total of $2^{w-1}+1$ (because we can also buy 0 for all slices), which are less than or equal to the budget. Mathematically, this can be expressed as

$$\text{solutions} = \sum_{a_1=0}^{2^{w-1}} \sum_{a_2=0}^{2^{w-1}-a_1} 1 = 2^{w-1} + 1 \tag{1}$$

This equation is an equivalent problem to stating how many distinct terms are in the binomial $(a_1 + a_2)^{2^{w-1}}$.

By extending to an arbitrary amount of assets A, this formulation equates to finding how many distinct terms are in the multinomial expansion $(a_1 + a_2 + ... + a_A)^{2^{w-1}}$, which can be found using Eq. 2 :

$$\text{solutions} = \prod_{a=1}^{A-1} \frac{2^{w-1} + a}{a} = \frac{(2^{w-1} + A - 1)!}{(2^{w-1})!(A-1)!} \tag{2}$$

C Figures

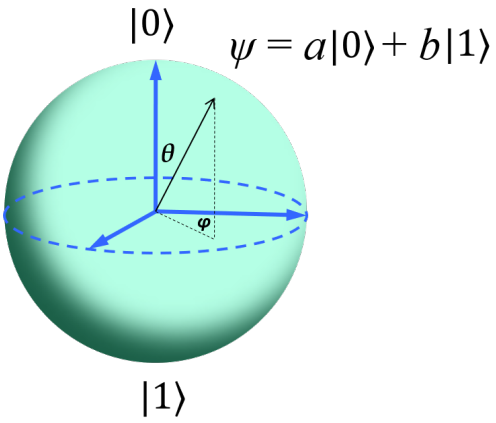

Figure 1: The Bloch sphere is the geometric representation of a qubit ψ as a superposition of two orthogonal states. Every point on the surface of the sphere corresponds to valid qubit, whereas the states of a classical bit correspond only to the the north and south poles designed as 0 and 1, respectively. A qubit is specified by the complex-valued coefficients a and b, which may be defined in terms of the spherical coordinates θ and ϕ.

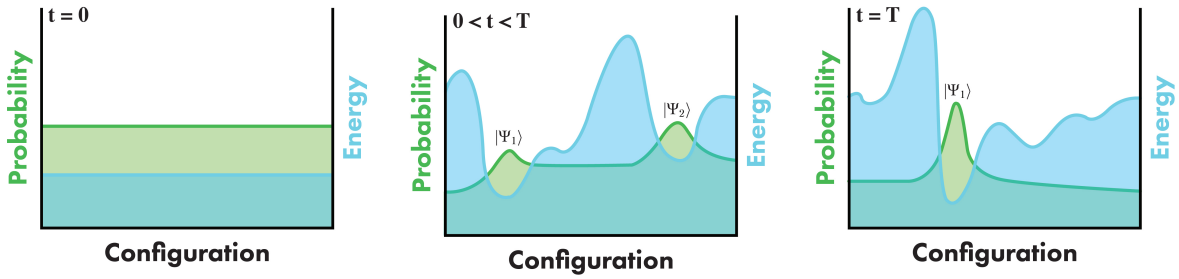

Figure 2: At $t = 0$, QA begins with a prepared state with uniform probability. During the annealing steps, the probability begins to concentrate at the minimums. The dynamics drive the probability toward the global minimum by final time T.

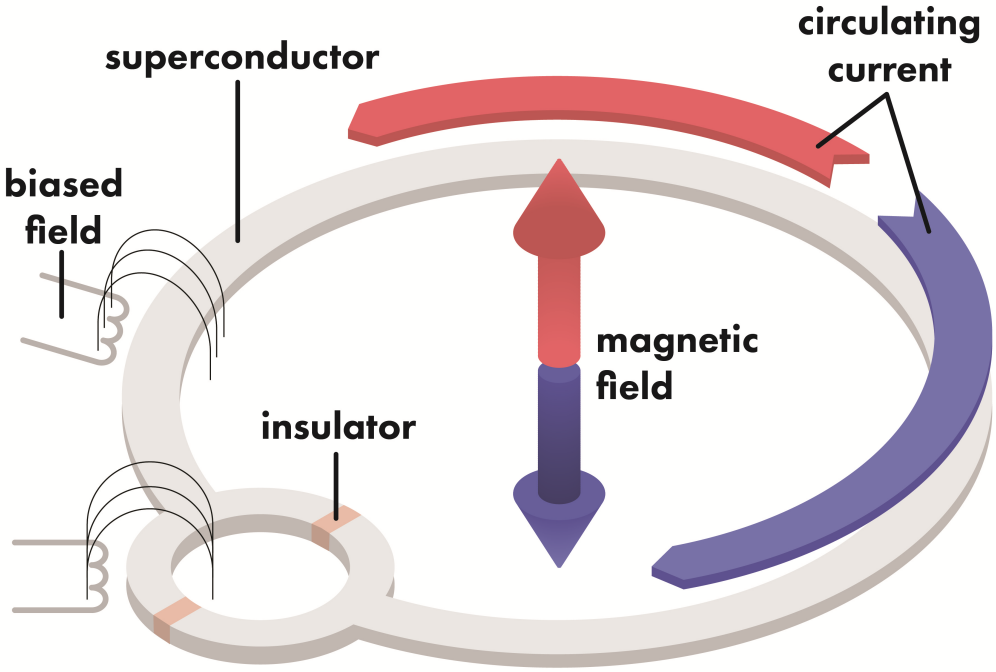

Figure 3: A flux-qubit design based on a compound Josephson junction in which counter-propagating currents induce a magnetic field. The flux qubit is encoded within the resulting magnetic flux while externally applied control biases tune the current [79].

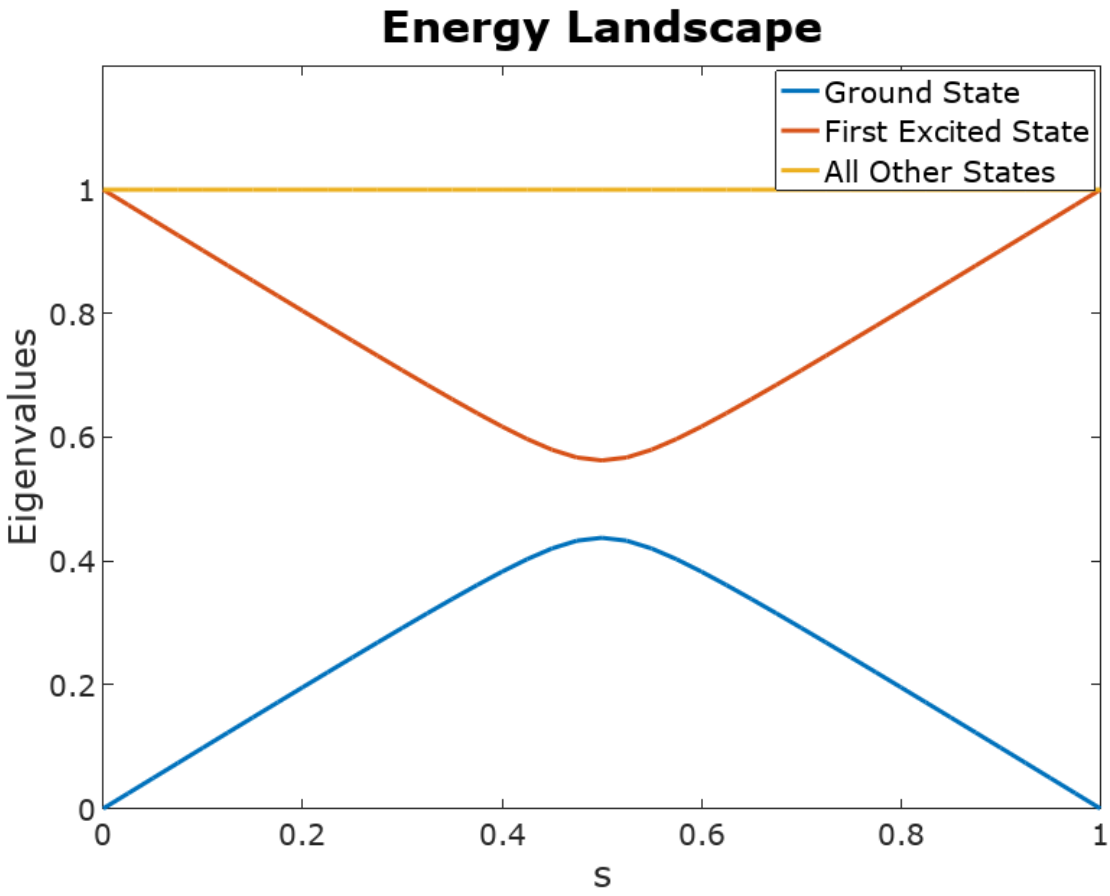

Figure 4: The time-dependent energy eigenspectrum for the time-dependent Hamiltonian used an example of Grover's search algorithm for AQC where $s = \frac{t}{T}$ is the position in the anneal schedule. The spectral gap between the ground state and first excited state changes with time [148].

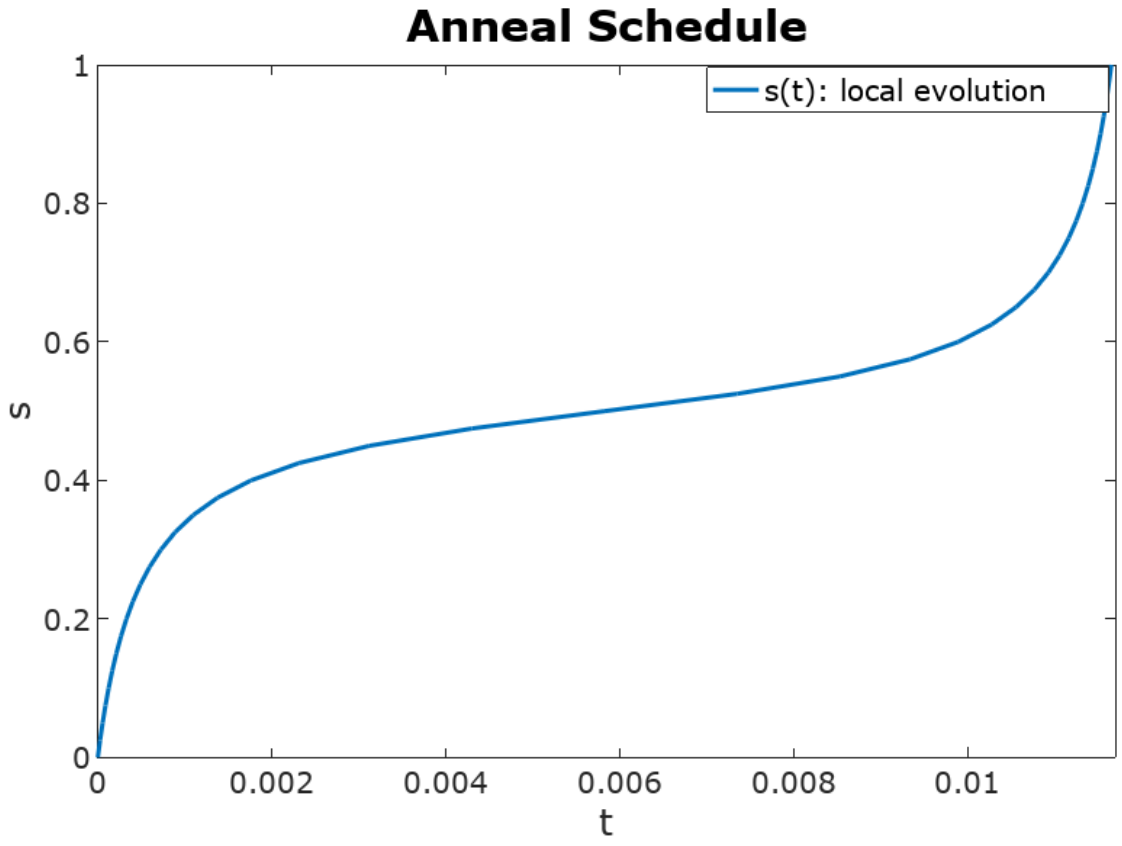

Figure 5: The optimal local schedule for implementing Grover's search using AQC accounts for the time-dependent behavior of the spectral gap shown in Figure 4 where $s = \frac{t}{T}$ is the position in the anneal schedule. This schedule tailors the dynamics to evolve more slowly near the minimum spectral gap at $s = .5$ and faster outside of this region.

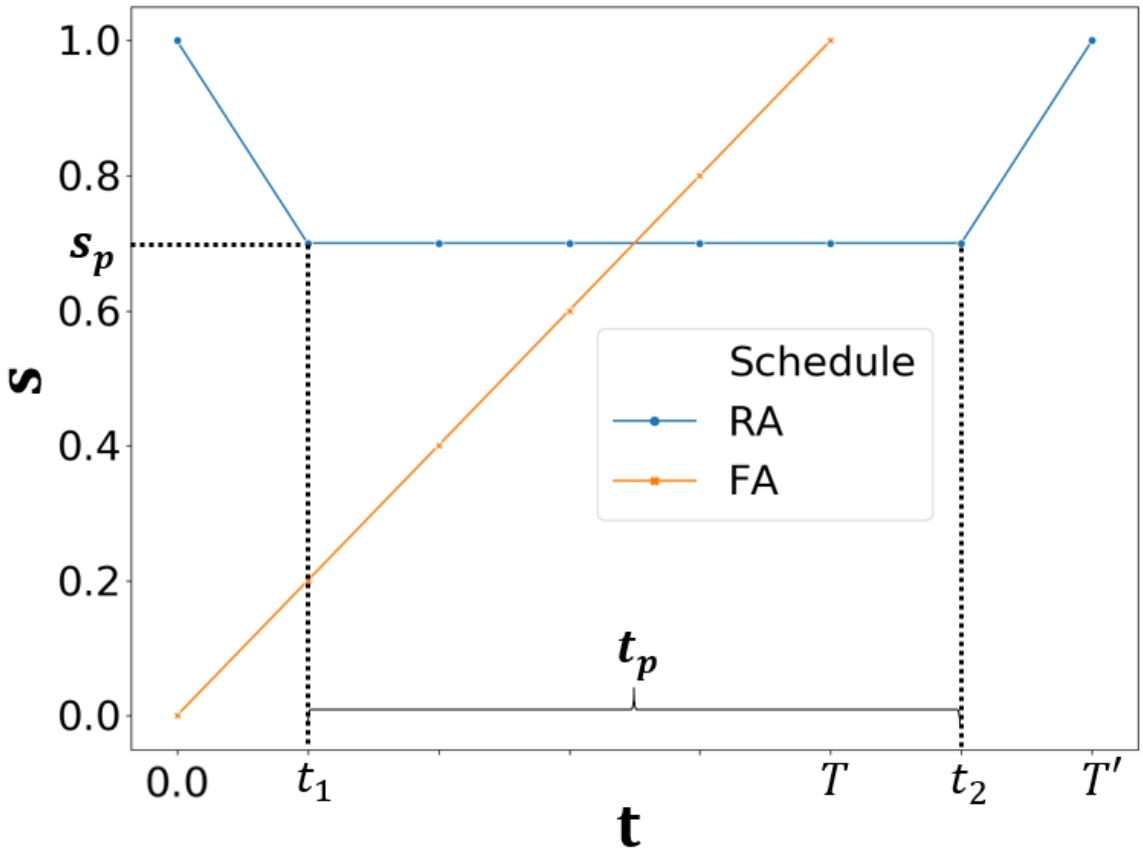

Figure 6: The control schedule for reverse annealing (RA) compared with forward annealing (FA) plotted with respect to time. The control schedule for forward annealing starts at $t = 0, s = 0$ and anneals at a constant rate to $t = T, s = 1$, whereas the control schedule for reverse annealing starts at $t = 0$ with $s = 1$, decreases to a value s_p at time t_1, pauses for time $t_p = t_2 - t_1$, and then increases to $s = 1$ at time T'.

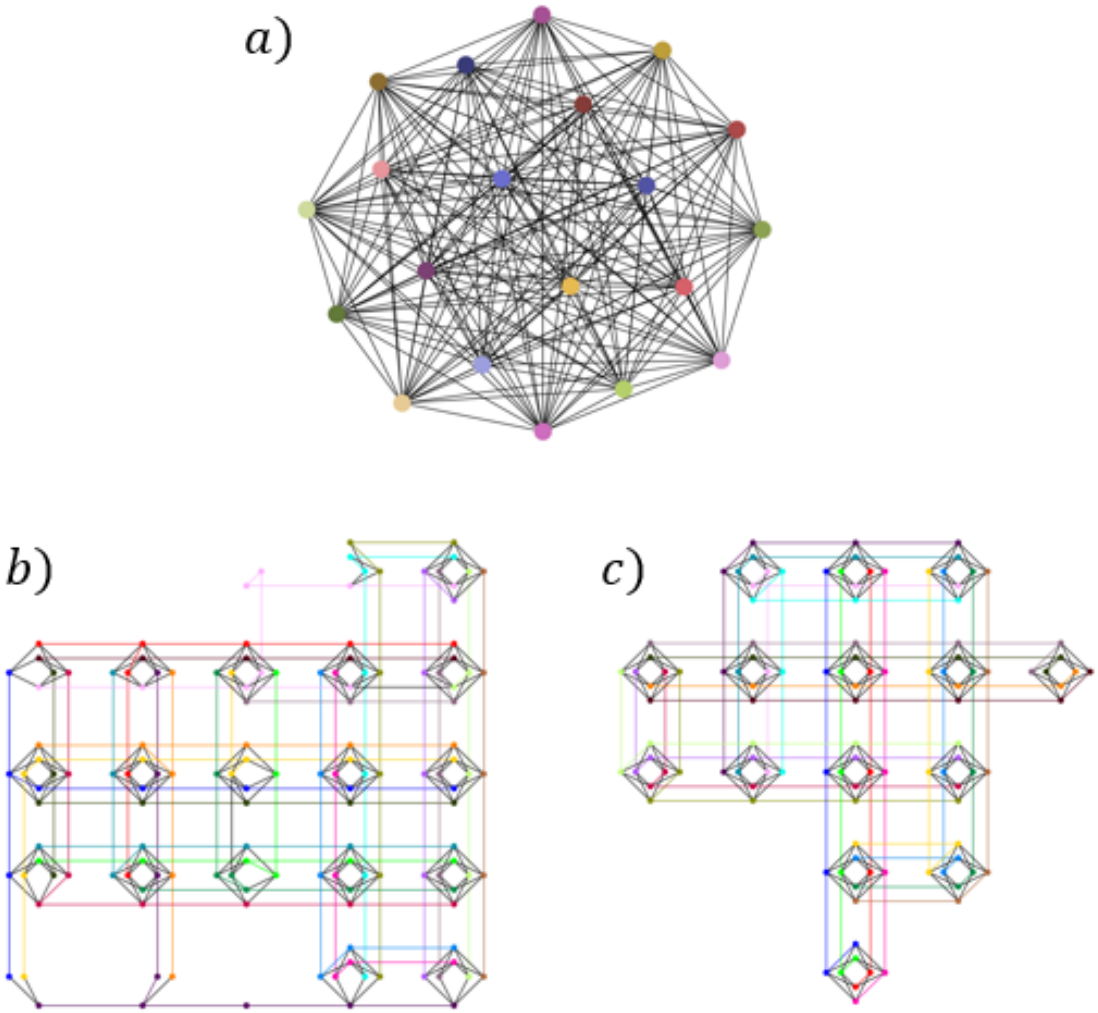

Figure 7: The embedding of a 20 logical spin complete graph onto a Chimera graph structure. Figure (a) is a complete K_{20} graph, which is fully connected with 20 nodes and 190 edges where each node represents a logical spin and each edge is a coupling between spins. Figure (b) is the CMR algorithm, which requires the allocation of 23 unit cells. Figure (c) is the clique embedding algorithm, which requires the allocation of 15 unit cells. The nodes represent physical qubits, lines are the couplings between physical qubits, and each color is a different physical spin chain corresponding to a logic spin.

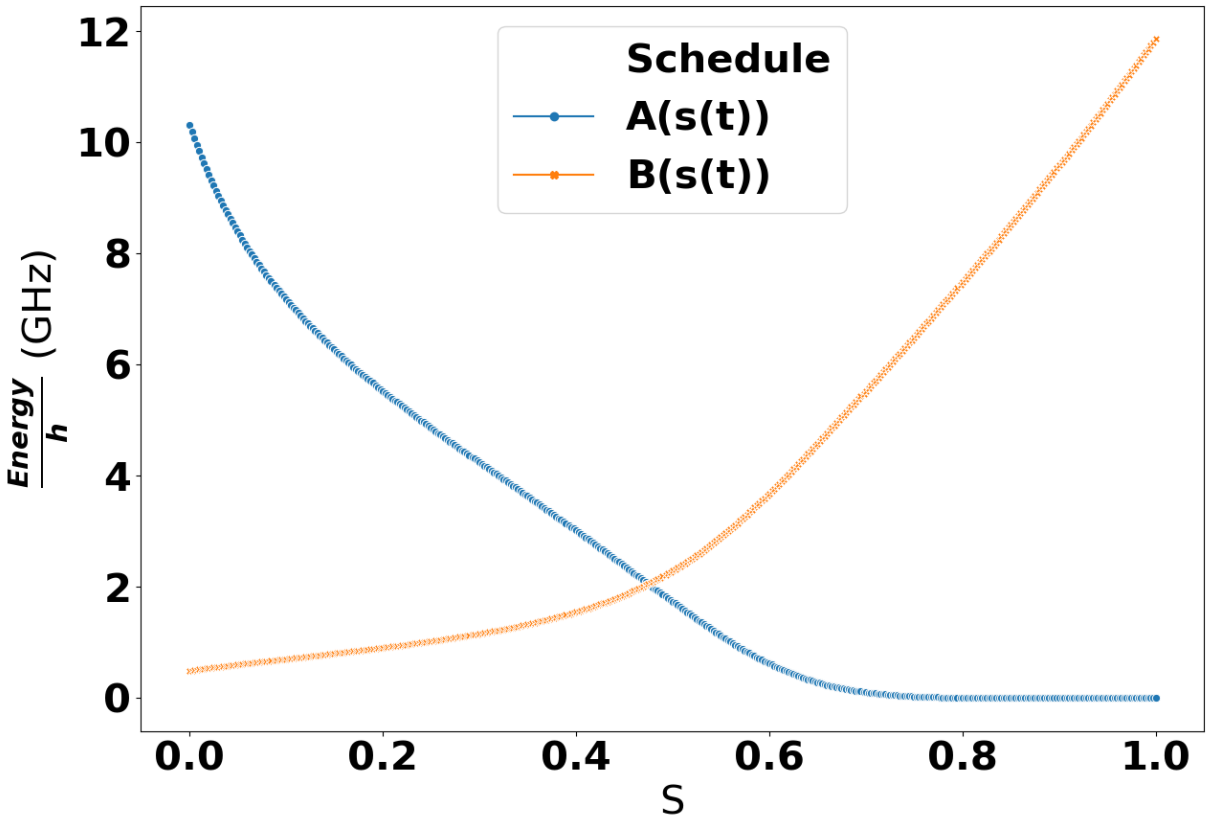

Figure 8: Actual D-Wave schedule for the particular D-Wave 2000Q_5 machine measured from $s = 0$ to $s = 1$ in increments of 0.001.

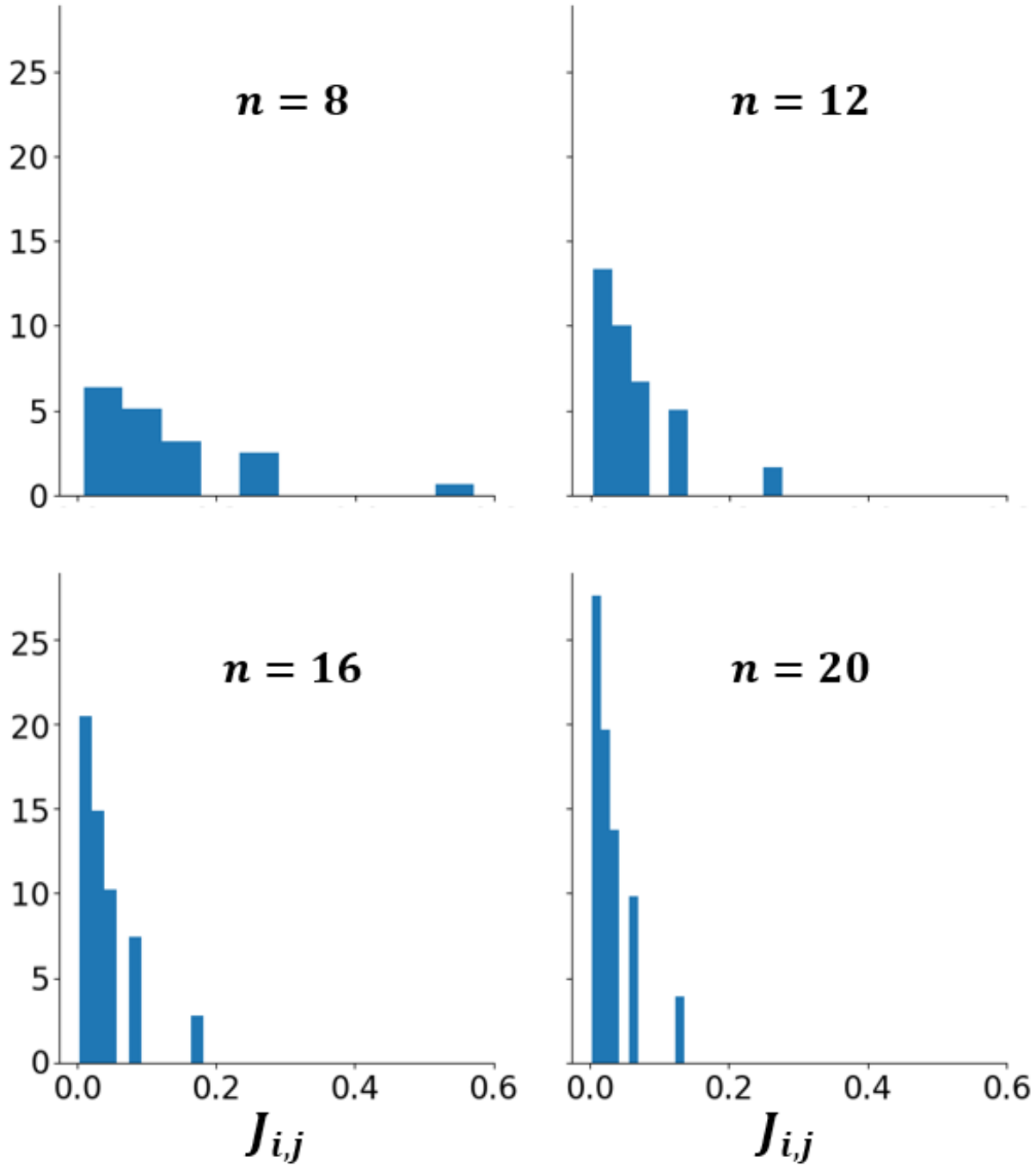

Figure 9: A histogram of all $J_{i,j}$ values for 1000 portfolio optimization problems for each problem size n. This graph is normalized to the probability density.

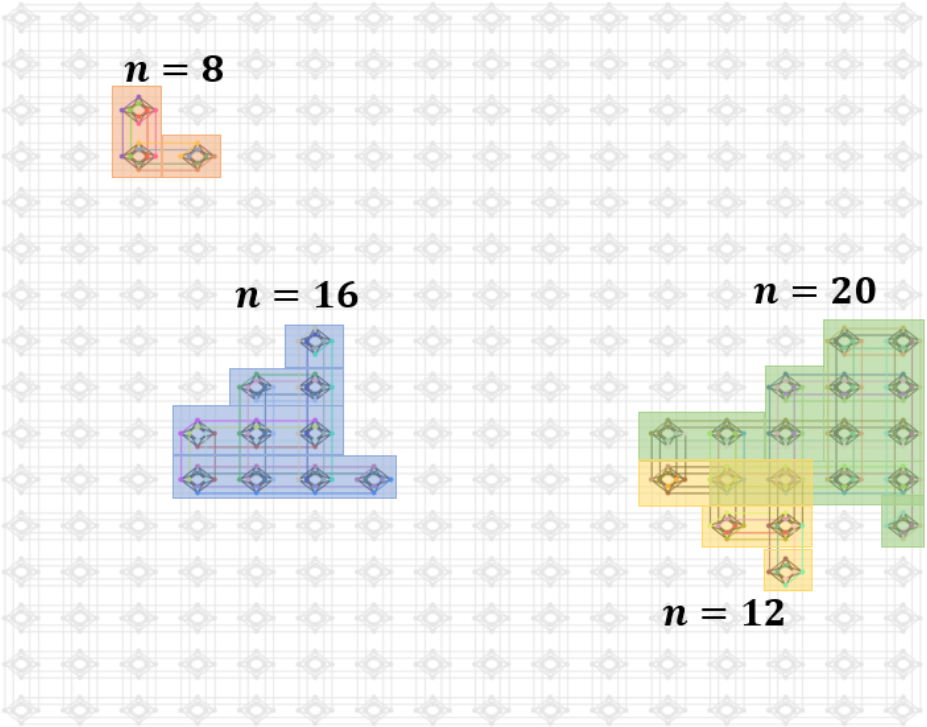

Figure 10: The clique embedding graphs used for all 1000 problems on the D-Wave 2000Q hardware for problem sizes n.

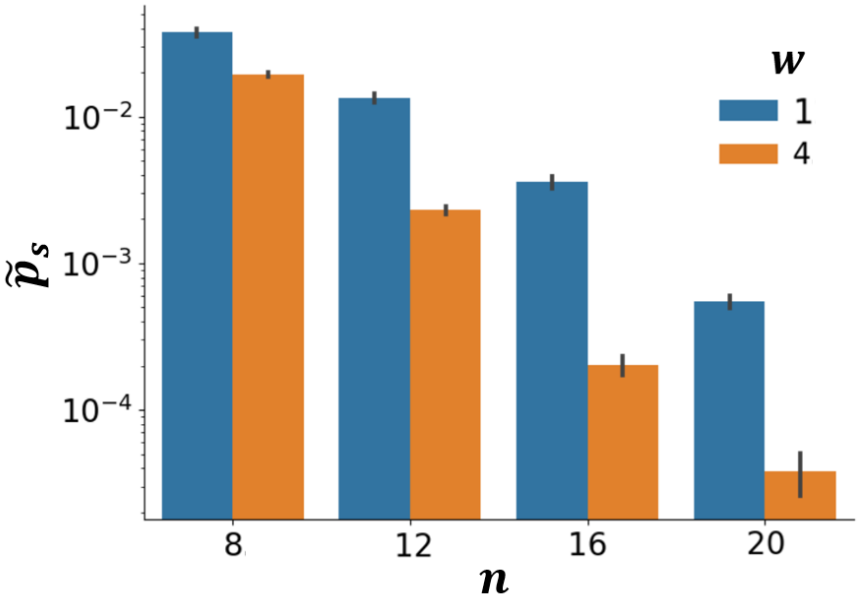

Figure 11: The average probability of success over 1000 problems each with 1000 samples using CMR, $g = 0$, and $T = 15$ μs. The comparison is between a set of problems from problem sizes 8 to 20 for $w = 1$ (blue) and $w = 4$ (orange) The problems set to slices $w = 1$ are much less complex and therefore have a much higher probability of success.

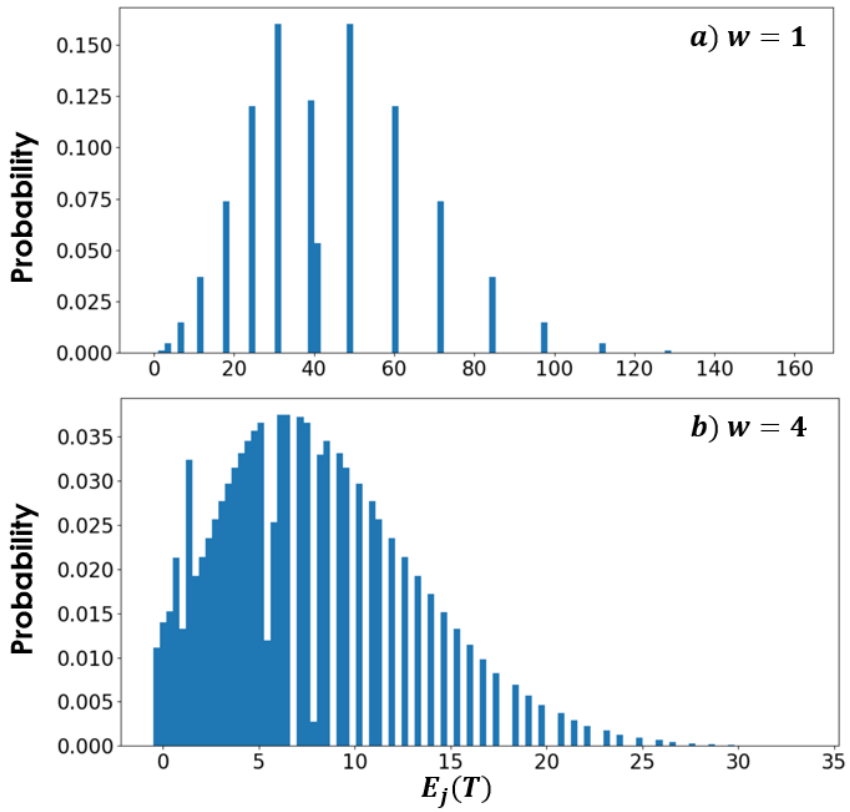

Figure 12: Probability histogram (100 bins) of all possible energies for problem of size 20 where a) is of $w = 1$ and b) is of $w = 4$. There is a higher density of states close to e_0 in figure b) and therefore more opportunities to jump to an excited state throughout the sample.

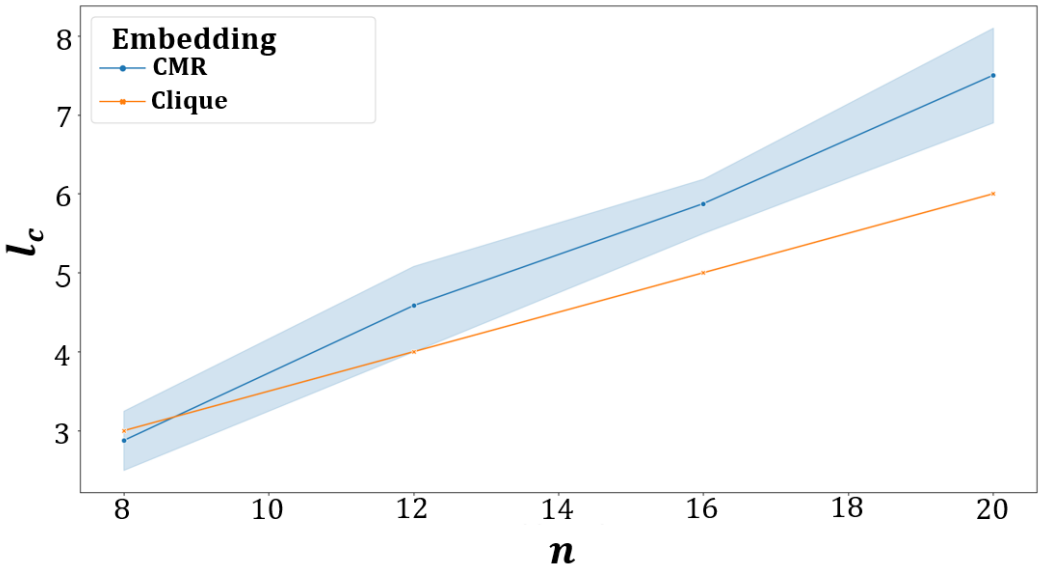

Figure 13: The average chain length over all chains for a given embedding clique and CMR embedding as n increases.

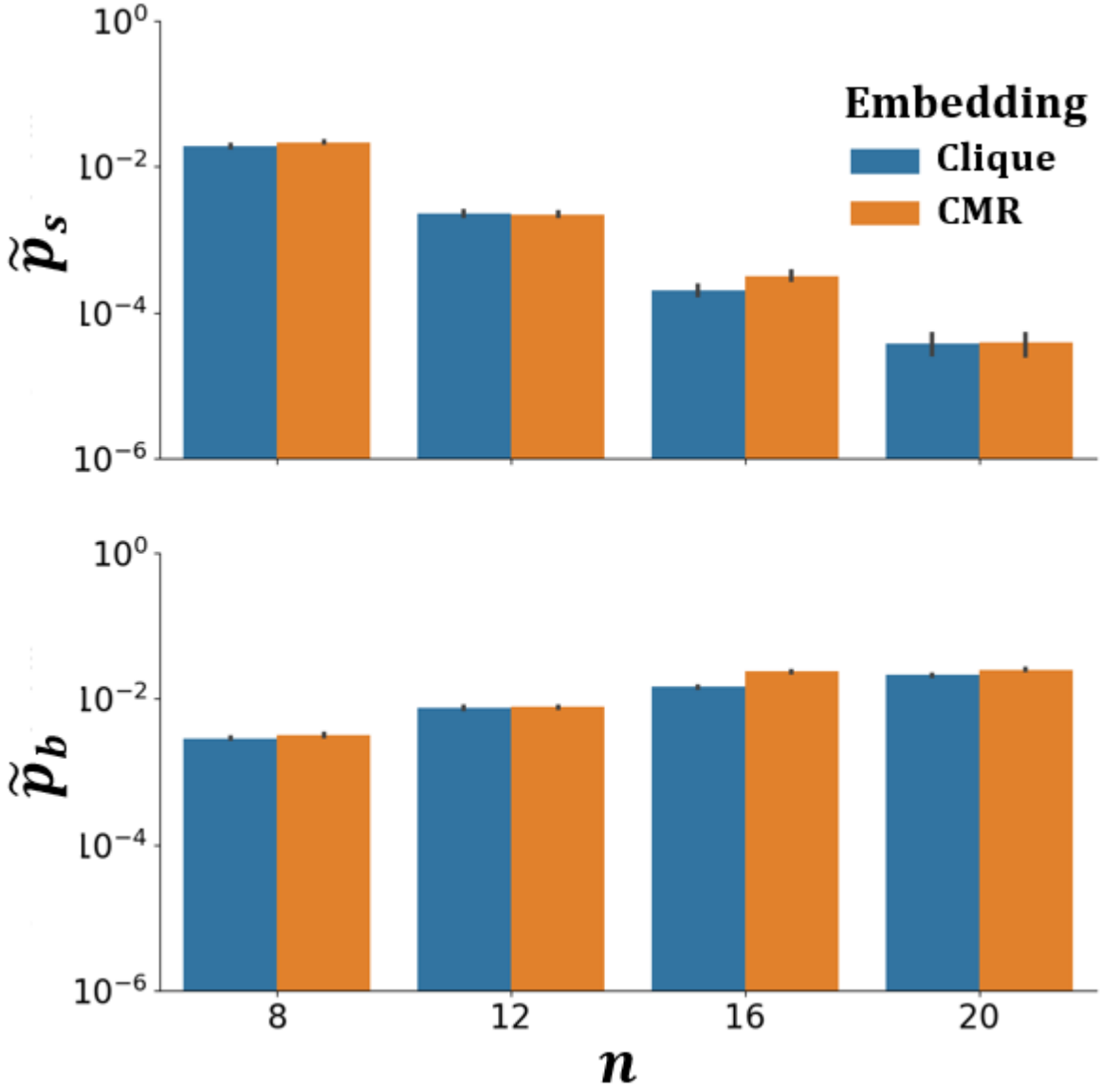

Figure 14: The $\tilde{p}_s$ (top) and $\tilde{p}_b$(bottom) on a log scale over 1000 samples for 1000 problems comparing CMR to clique embedding for parameter settings of $g = 0$ and $T = 100$ μs.

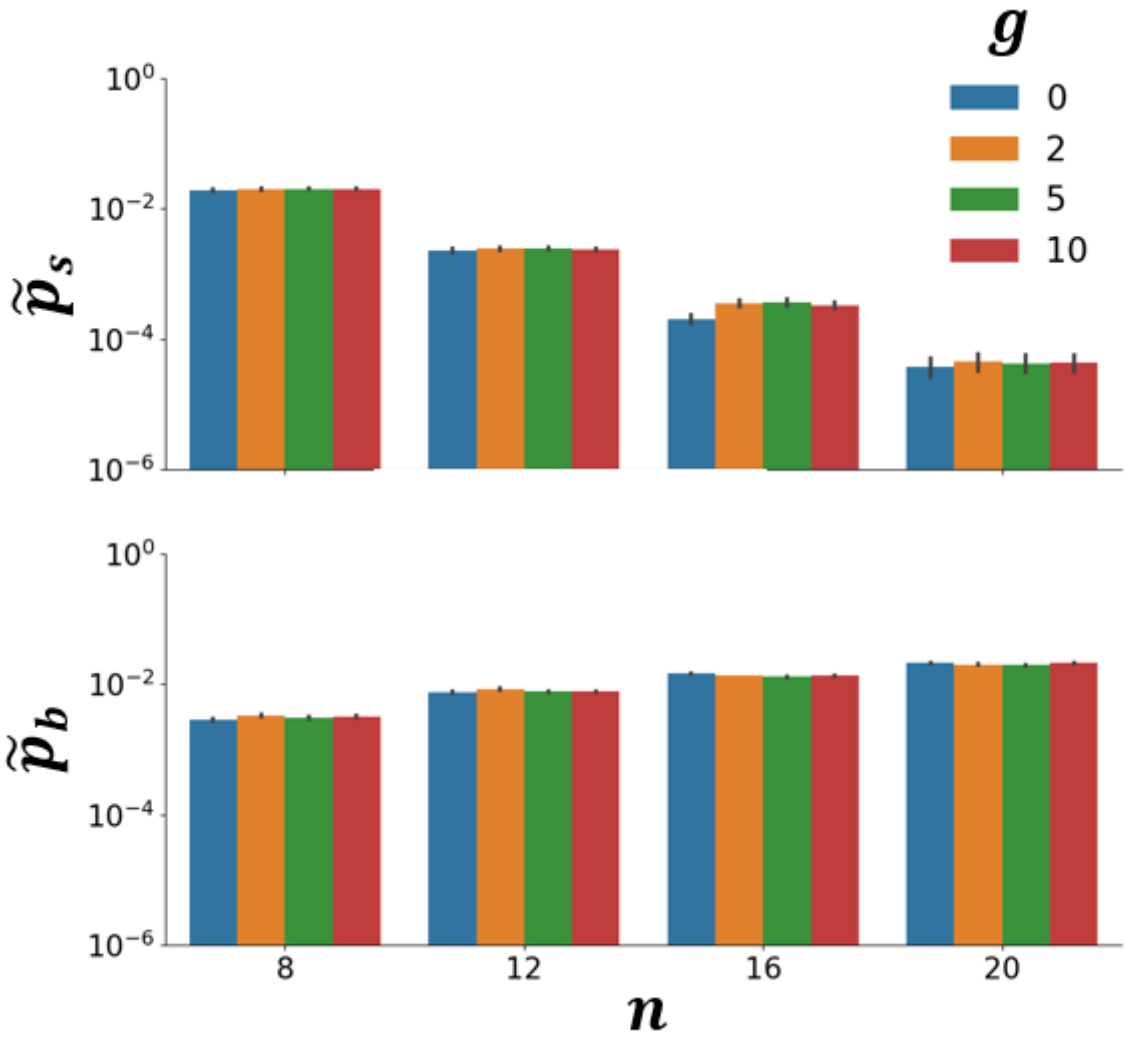

Figure 15: The average p_s (top) and p_b(bottom) on a log scale over $N_s = 1000$ samples for $N_p = 1000$ problems at $g = 0 \rightarrow 10$ for parameter setting of $T = 100\mu s$ and clique embedding.

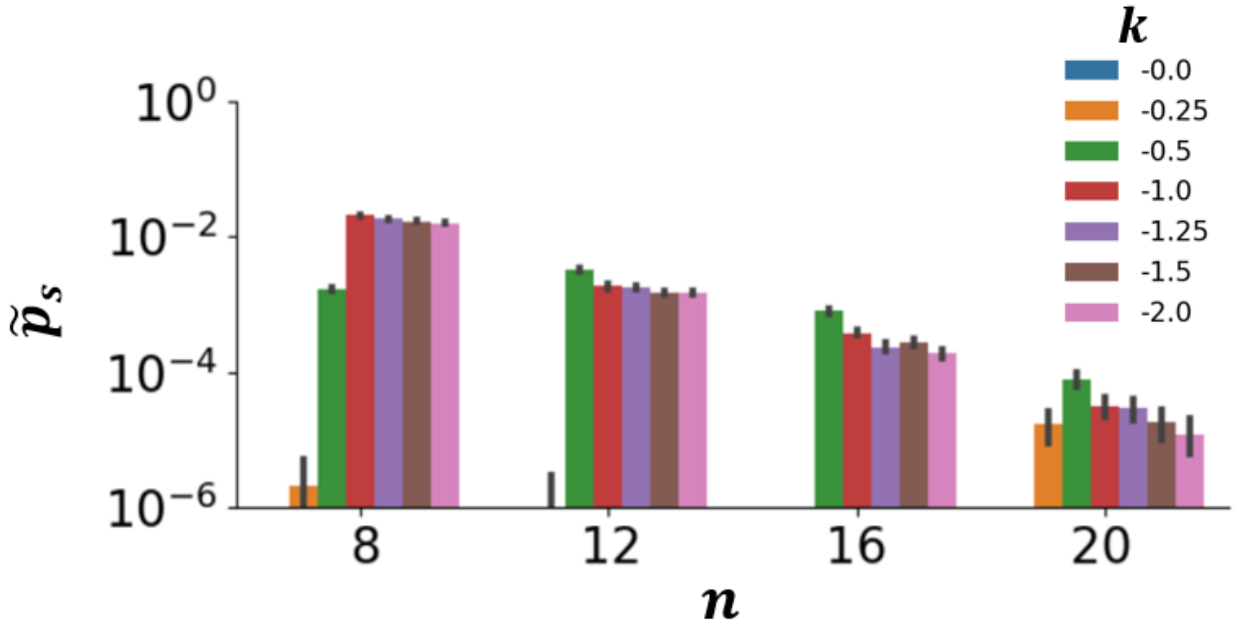

Figure 16: The average probability of success $\tilde{p}_s$ comparing intra-chain strengths $k = [0 \rightarrow -2]$ for 1000 samples of 1000 problems for each problem size n. All samples with one or more broken chains are discarded (no post-processing) and counted as incorrect.

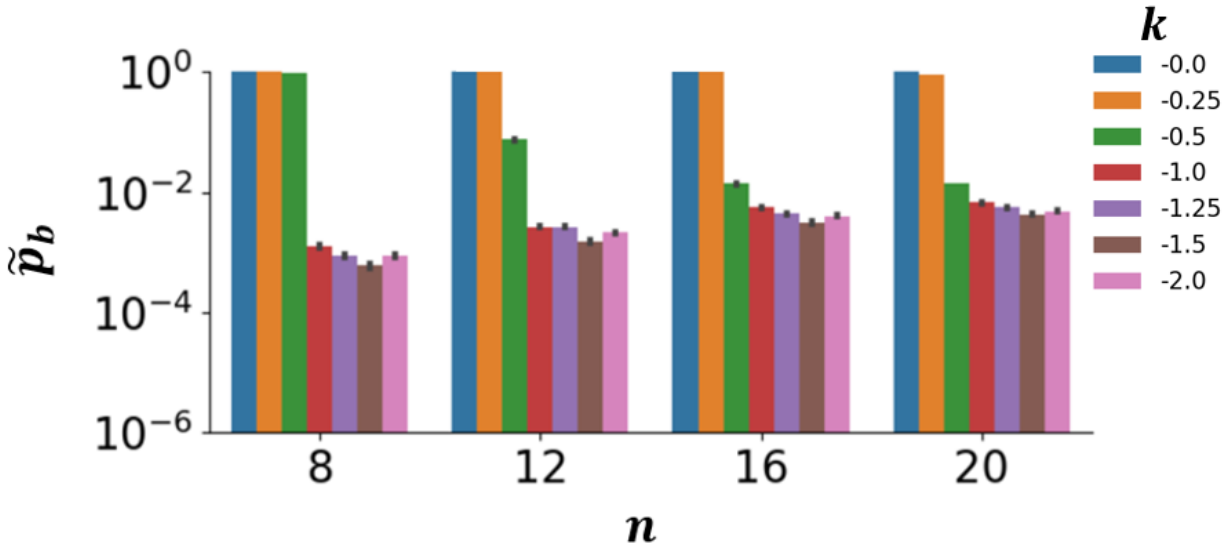

Figure 17: The average probability that a sample has at least one broken chain $\tilde{p}_b$ comparing intra-chain strengths $k = [0 \rightarrow -2]$ for 1000 samples of 1000 problems for each problem size n.

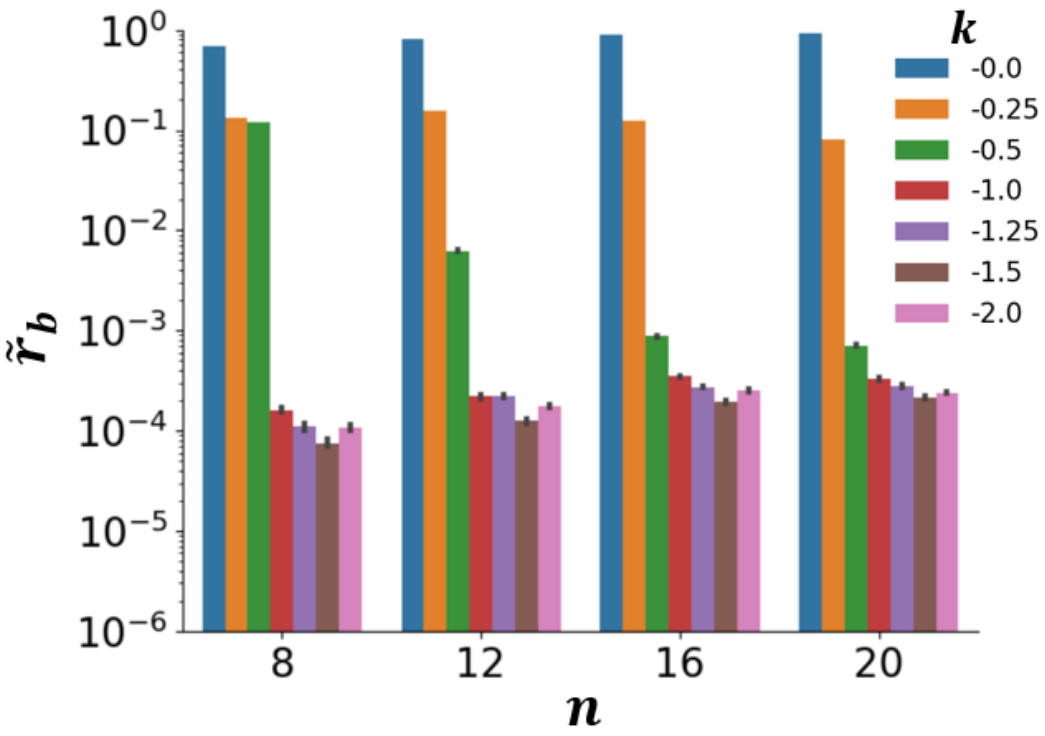

Figure 18: The average ratio of broken chains in a sample $\tilde{r}_b$ comparing intra-chain strengths $k = [0 \rightarrow -2]$ for 1000 samples of 1000 problems for each problem size n.

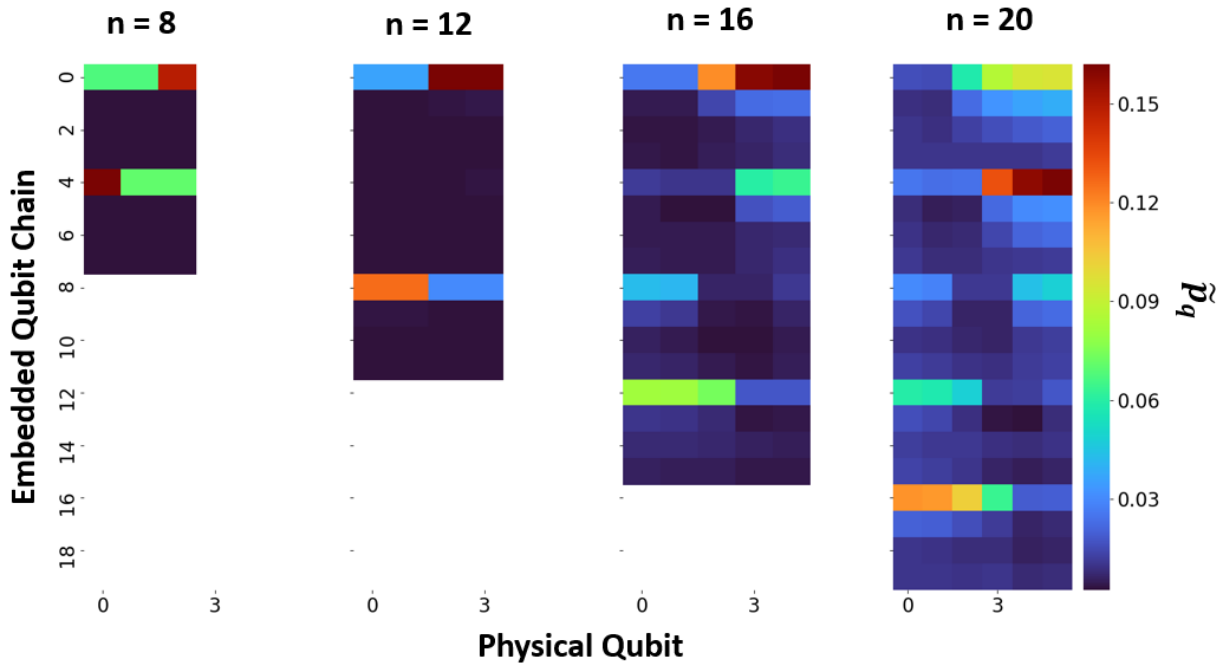

Figure 19: A heat map showing average probability for each physical qubit in a chain to break for a sample over 1000 problems each with all broken samples where $k = -0.5$. The comparison is between a set of problems from problem sizes n.

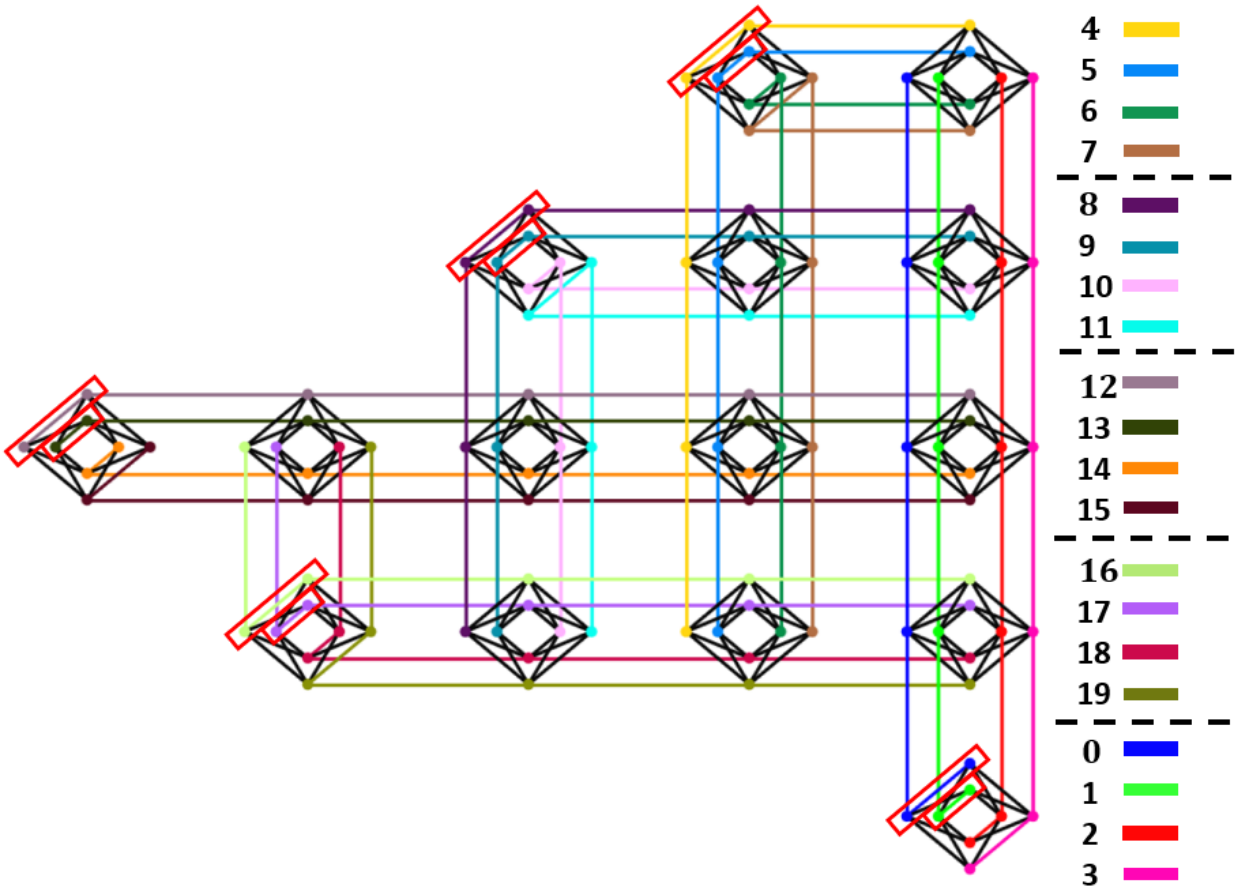

Figure 20: The clique embedding graphs for problem size $n = 20$ where the intra-unit cell coupling for qubit chains $0, 4, 8, 12, 16$ are boxed and correspond to the qubits which have the lowest probability of being faulty for those chains.

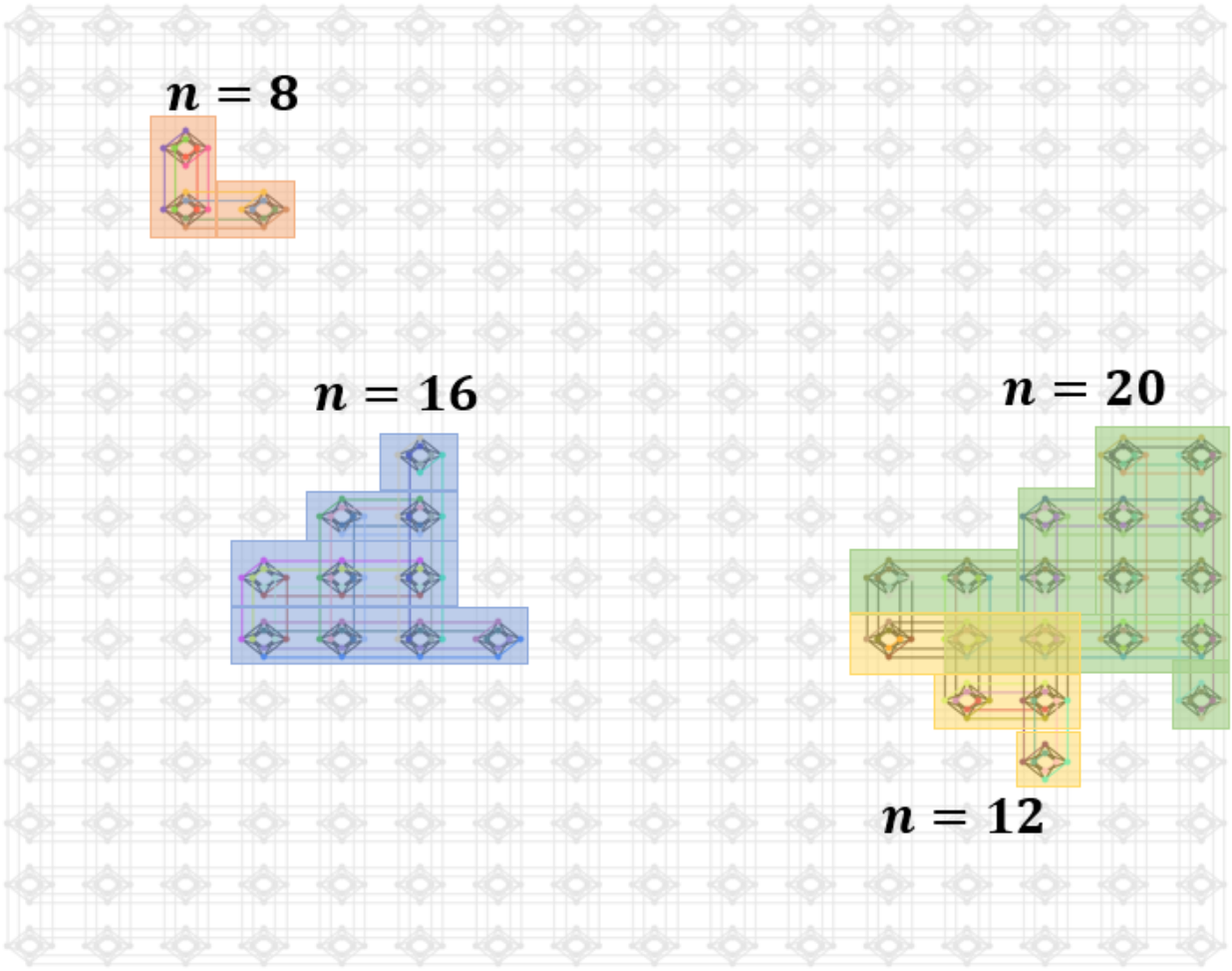

Figure 21: The clique embedding graphs used for all 1000 problems on the D-Wave 2000Q hardware for problem sizes n.

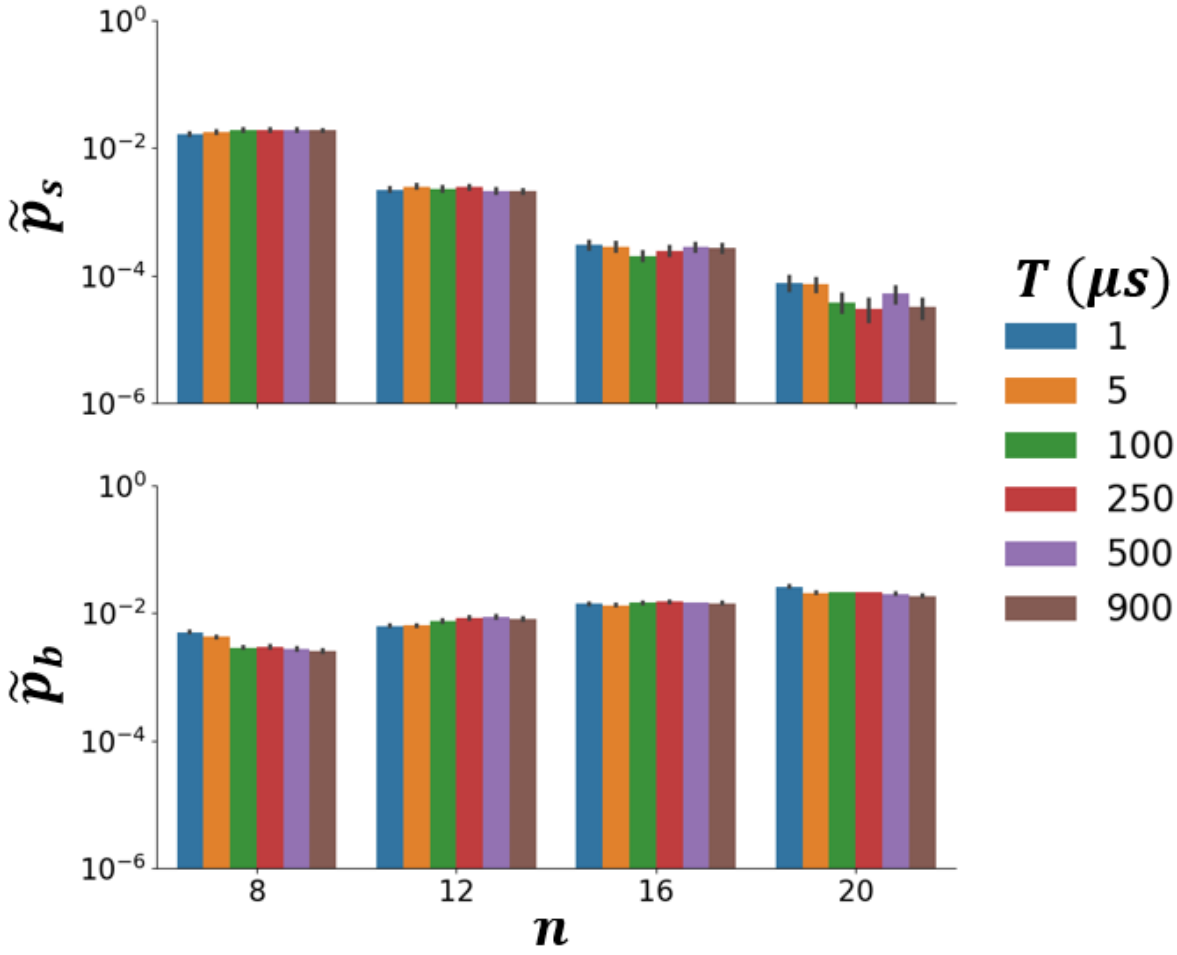

Figure 22: The average $\tilde{p}_s$ (top) and $\tilde{p}_b$(bottom) on a log scale over 1000 samples for 1000 problems at various annealing times for parameter settings of $g = 0$ and clique embedding.

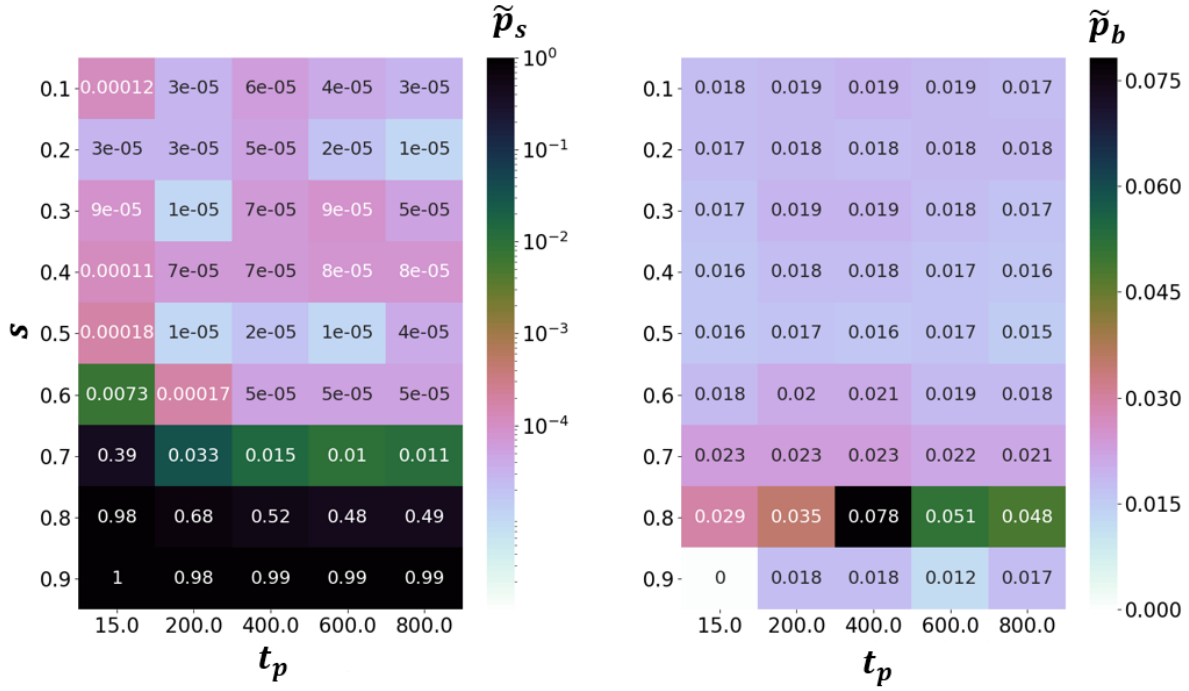

Figure 23: The $\tilde{p}_s$ (left) and $\tilde{p}_b$ (right) for reverse annealing where $e_i = e_0$ and as $s = [0.1 \to 0.9]$ and $t_p = [15\mu s \to 800\mu s]$ for $n = 20$ with $m = 5$ assets and $w = 4$.

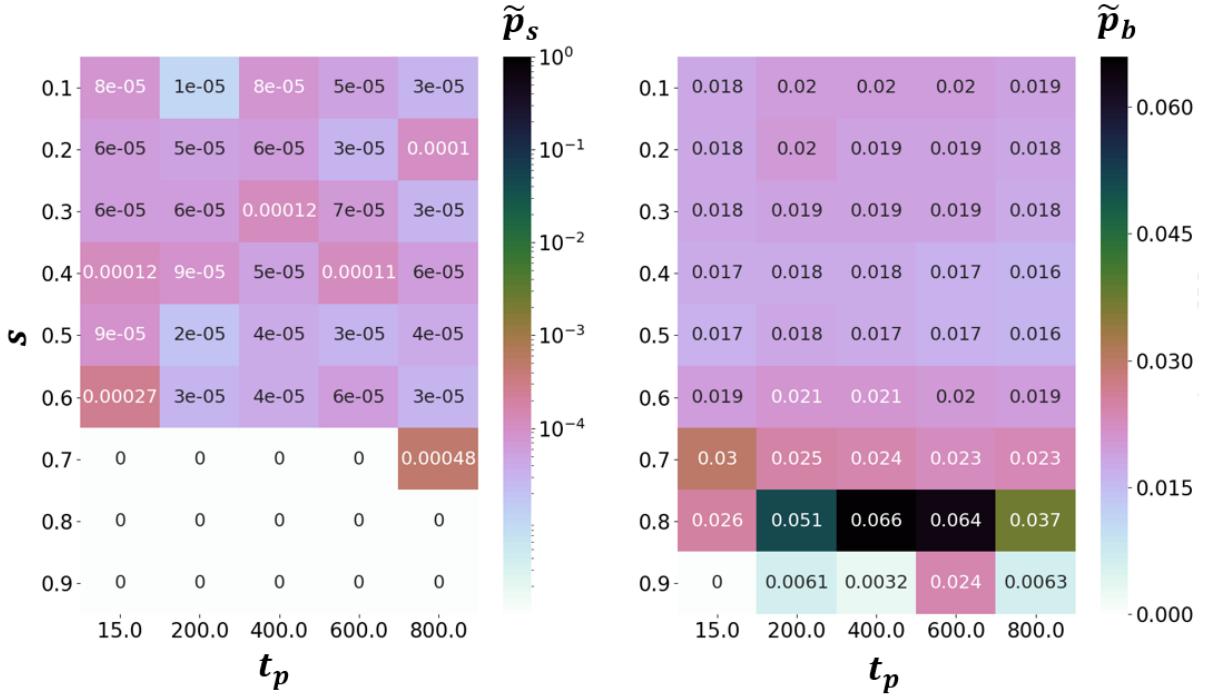

Figure 24: The $\tilde{p}_s$ (left) and $\tilde{p}_b$ (right) for reverse annealing where $e_i = e_1$ for each problem, $s = [0.1 \to 0.9]$, and $t_p = [15\mu s \to 800\mu s]$ for problem size 20 with 5 assets and 4 slices.

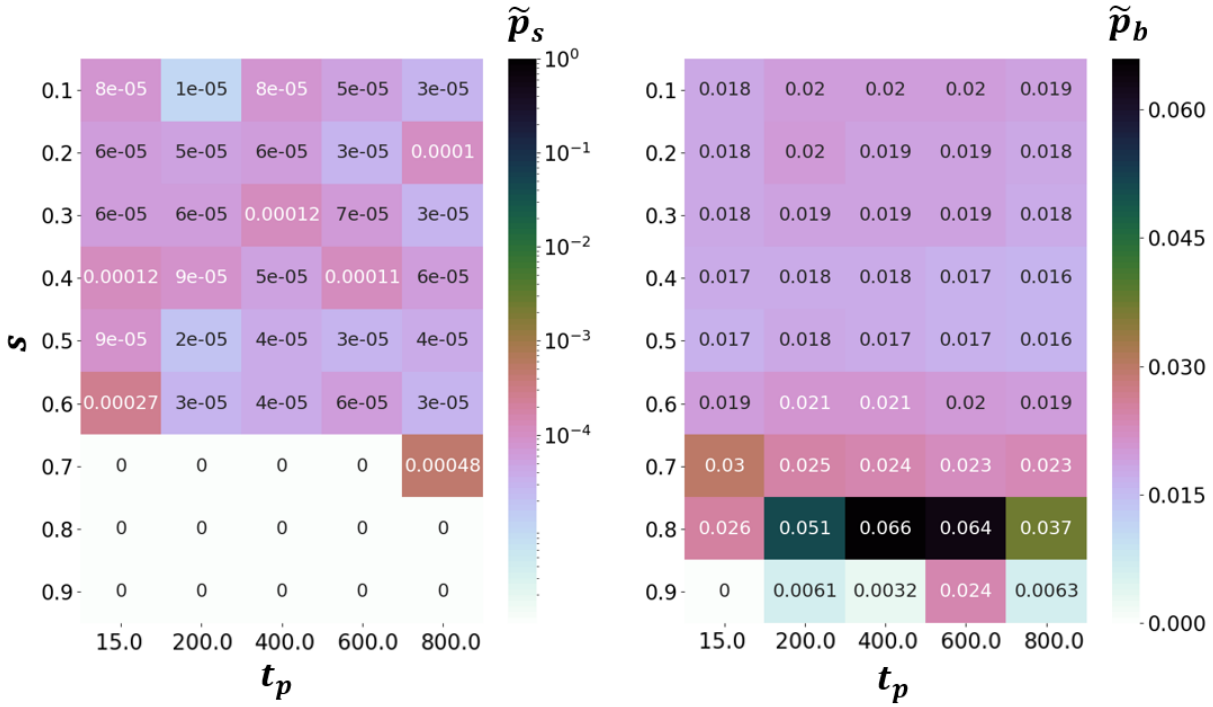

Figure 25: The $\tilde{p}_s$ (left) and $\tilde{p}_b$ (right) for reverse annealing where $e_i = e_f$ for each problem, $s = [0.1 \rightarrow 0.9]$, and $t_p = [15\ \mu s \rightarrow 800\ \mu s]$ for problem size 20 with 5 assets and 4 slices. The 6 problems where $e_f = e_g$ were excluded. Thus, $\tilde{p}_s = p(e_e \rightarrow e_0)$.

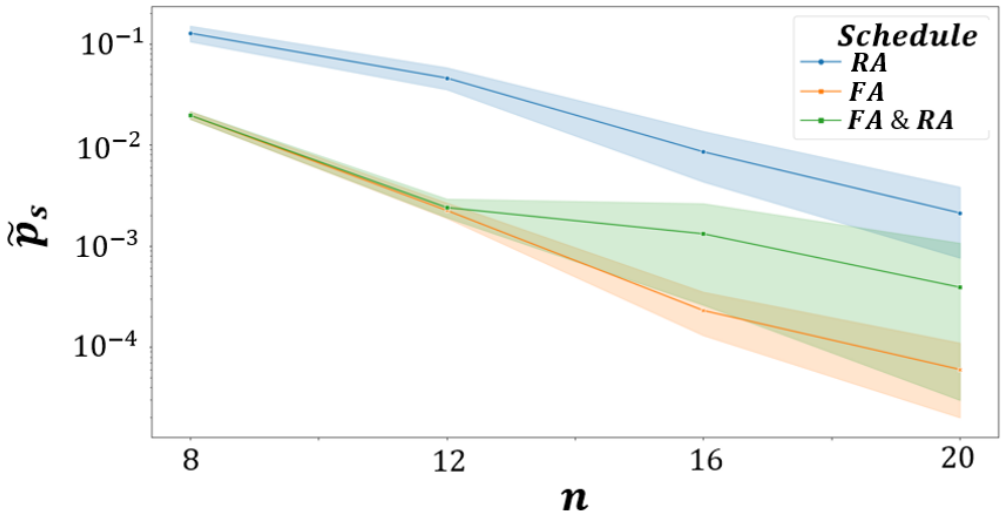

Figure 26: The $\tilde{p}_s$ as a function of n over a set of 100 problems each with 1000 samples. Reverse annealing (blue) with $e_i = e_f$, $s = .7$, and $t_p = 400\mu s$ is compared to forward annealing (orange) with clique embedding, $g = 0$, and annealing time = 100 μs. The combination of forward annealing and reverse annealing is also compared where the $\tilde{p}_s$ is chosen by problem (green). In this green trend, the $\tilde{p}_s$ is calculated using the forward annealing $\tilde{p}_s^{(k)}$ for the 6 problems where forward annealing would have provided reverse annealing with an $e_i = e_0$ and the reverse annealing $\tilde{p}_s^{(k)}$ for the 94 problems where $e_i \neq e_0$.

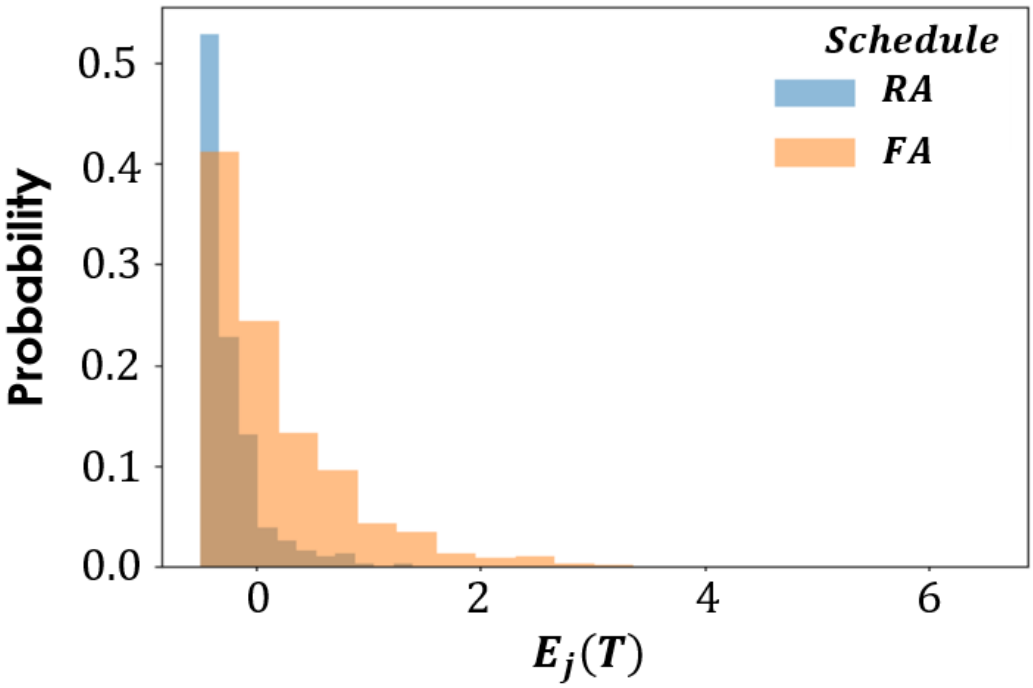

Figure 27: A probability histogram (20 bins) comparing all energies found with forward annealing and reverse annealing from all 1000 samples for the 94 problems where ($e_i \neq e_0$) with problem size 20 (5 assets and 4 slices).

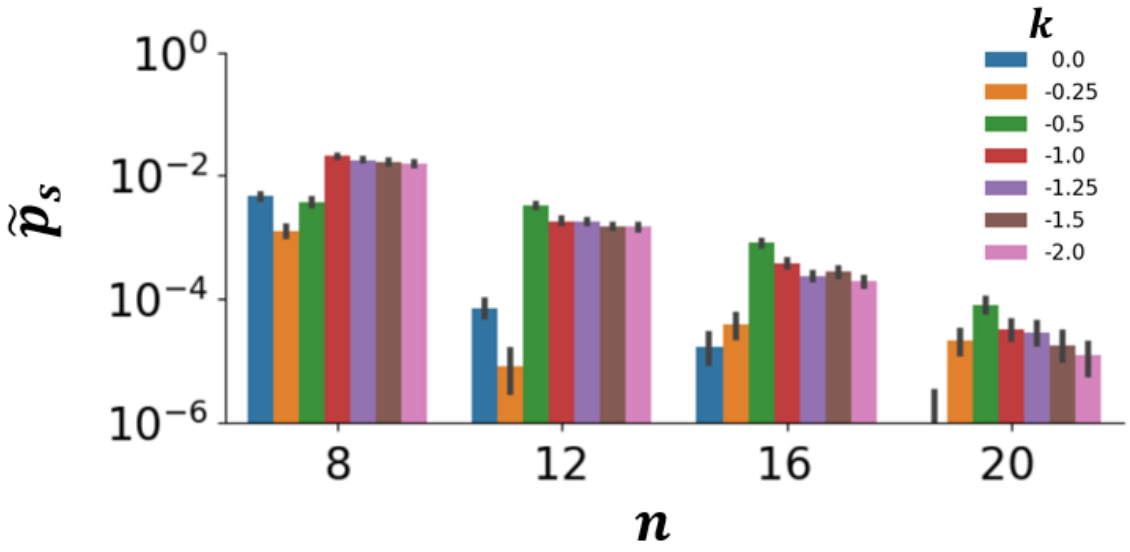

Figure 28: The $\tilde{p}_s$ over 1000 problems each with 1000 samples where k is varied. The comparison is between a set of problems from problem sizes 8 to 20. The post-processing method used is majority vote.

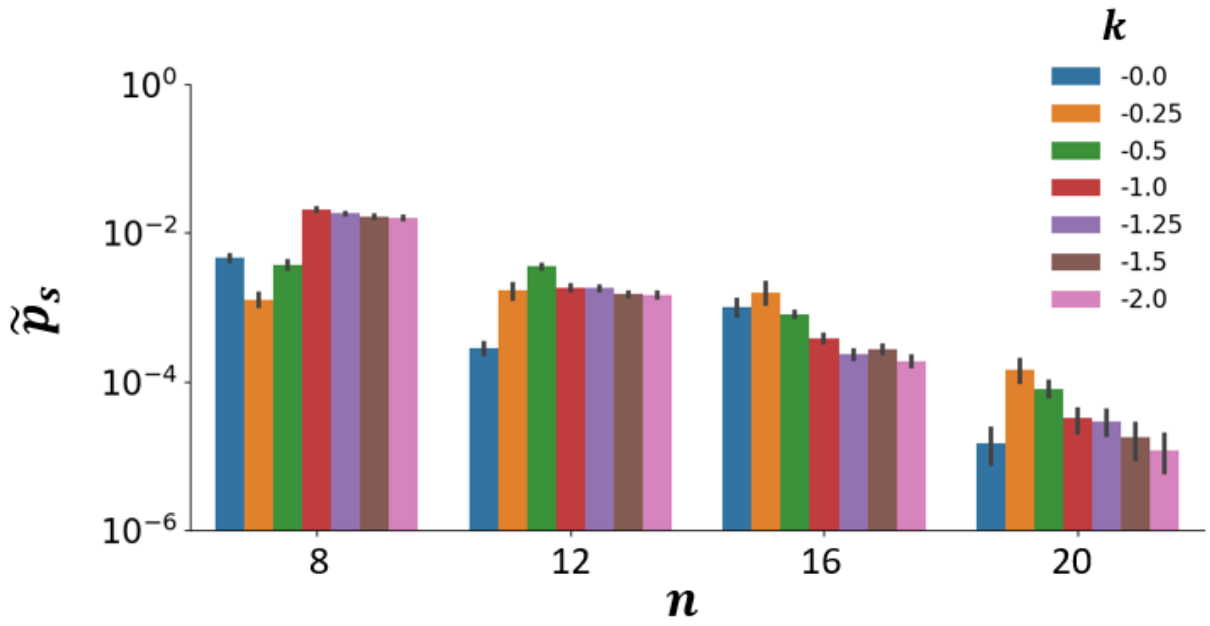

Figure 29: The $\tilde{p}_s$ over 1000 problems each with 1000 samples where $k = [0 \rightarrow -2]$. The comparison is between a set of problems from problem size n. The post-processing method used is our custom weighted random technique incorporating the probabilities of faulty qubits.

Vita

Erica Grant grew up in Richmond, Virginia. After high school, she attended Virginia Tech and received a Bachelor of Science degree in Physics with a minor in Nanoscience. After interning at Oak Ridge National Laboratory for two summers of undergrad, she knew that she wanted to attend graduate school and study quantum computing. She chose to attend the University of Tennessee, Knoxville to pursue a Doctor of Philosophy degree in Energy Science and Engineering in the Bredesen Center with a cross-cutting sciences focus in quantum computing. Her research interests include studying the tuning strategies of quantum computers and real-world quantum computing applications such as optimization. During graduate school, Erica also independently pursued the invention of a novel security system which leverages quantum information technology, Quantum Lock. After graduation, she will begin her new position as founder of Quantum Lock Technologies LLC which received funding from the Department of Energy's Innovation Crossroads program for product development. She is incredibly grateful for all the support from her partner, family, and friends as she begins her new career.