# Leveraging Artificial Intelligence to Advance Problem-Solving with Quantum Annealers

by

Ramin Ayanzadeh

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County in partial fulfillment
of the requirements for the degree of Doctorate
of Philosophy in Computer Science
2020

Advisory Committee:
Research Professor Milton Halem, Co-Chair/Advisor
Professor Tim Finin, Co-Chair/Advisor
Research Associate Professor John Dorband
Professor Samuel Lomonaco
Scientist Daniel O'Malley

# ABSTRACT


Title of dissertation: Leveraging Artificial Intelligence to Advance Problem-Solving with Quantum Annealers

Ramin Ayanzadeh, Doctor of Philosophy, 2020

Dissertation directed by: Professor Milton Halem
Department of Computer Science
Professor Tim Finin
Department of Computer Science



We show how to advance quantum information processing, specifically problem-solving with quantum annealers, in the realm of artificial intelligence. We introduce SAT++, as a novel quantum programming paradigm, that can compile classical algorithms (implemented in classical programming languages) and execute them on quantum annealers. Moreover, we introduce a post-quantum error correction method that can find samples with significantly lower energy values, compared to the state-of-the-art techniques in quantum annealing. We also demonstrate that performing post-quantum error correction methods can make results of quantum annealers reproducible—i.e., more robust behavior of quantum annealers, compared to recent software and hardware advancements in quantum annealing.

In addition, we offer to conjugate both optimization and sampling aspects of the quantum annealers and introduce two AI hybrid approaches. In reinforcement quantum annealing (RQA) scheme, an intelligent agent interacts with a quantum annealer that plays the stochastic environment role of learning automata and searches in the space of Hamiltonians, rather than exploring the Hilbert space of a given Ising model. In the same manner, greedy quantum annealing (GQA) is a novel model that utilizes quantum an-

nealers to better select candidates in greedy algorithms. We experimentally demonstrate that our proposed AI hybrid approaches outperform the best-known techniques in quantum annealing, specifically when problems require longer chains for forming virtual qubits with higher connectivity. Furthermore, we introduce the theory of ensemble quantum annealing (EQA) that generates multiple (distinct) Ising Hamiltonians whose ground states are all identical to a solution to the problem of interest. Our experimental results reveal that applying EQA can significantly boost the performance of the quantum annealers.

After benchmarking the D-Wave 2000Q quantum processors, as a proof-of-concept, we apply our proposed models on two real-world problems. As our first study case, we use SAT++ for factoring pseudo-prime numbers on a D-Wave quantum processor that can jeopardize the security of modern public-key cryptography systems. As our second case of study, we address the NP-hard problem of compressive sensing (i.e., the $\ell_0$-norm problem of sparse recovery) through reducing the original problem of compressive sensing to Weighted-MAX-SAT instances, casting the $\ell_0$-norm problem of binary compressive sensing to quadratic unconstrained binary optimization (QUBO), and proposing to solve the original problems of binary compressive sensing and binary compressive sensing with matrix uncertainty on quantum annealers.

# DEDICATION

For Zahra.

# ACKNOWLEDGEMENTS

I extend my sincere appreciations to Profs. Milton Halem and Tim Finin for exemplifying intellectual integrity and being patience during this precious mentorship. Also, many thanks to my committee members who accepted to support me through their valuable comments.

This research has been supported by NASA grant (#NNH16ZDA001N-AIST 16-0091), NIH-NIGMS Initiative for Maximizing Student Development Grant (2 R25-GM55036), and the Google Lime scholarship. I would like to thank the D-Wave Systems management team for access to the 2000Q quantum computer at Burnaby, Canada. I also would like to thank the eBiquity research group and the NSF funded center for accelerated real-time analytics (CARTA) for access support to computational resources.

# TABLE OF CONTENTS

# LIST OF FIGURES

List of Abbreviations

| | |
|---|---|
| ADC | Analog to Digital Convertor |
| AI | Artificial Intelligence |
| AMSU | Microwave Sounding Unit |
| ANN | Artificial Neural Networks |
| APU | Annealing Processing Unit |
| ASIC | Application-Specific Integrated Circuits |
| BCS | Binary Compressive Sensing |
| CARTA | Center for Accelerated Real Time Analytics |
| CDCL | conflict-driven clause learning |
| CMOS | Complementary Metal Oxide Semiconductor |
| CNF | Conjunctive Normal Form |
| CNOT | Conditional Not |
| CPU | Central Processing Unit |
| CS | Compressive Sensing |
| CT | Computed Tomography |
| DAC | Digital to Analog Convertor |
| DPLL | Davis—Putnam—Lagemann—Loveland |
| EA | Evolutionary Algorithms |
| EDA | Electronic Design Automation |
| EQA | Ensemble Quantum Annealing |
| FFT | Fast Fourier Transform |
| FMRI | Functional Magnetic resonance imaging |
| FPGA | Field-Programmable Gate Array |
| GQA | Greedy Quantum Annealing |
| GPR | ground penetrating radar |
| GPU | Graphical Processing Unit |
| HDL | Hardware Description Languages |
| IBM | International Business Machines |
| IP | Integer Programming |
| IPU | Ising Processing Unit |
| ISAR | Inverse Synthetic Aperture Radar |
| LA | Learning Automata |
| LANL | Los Alamos National Laboratory |

| | |
|---|---|
| LP | Linear Programming |
| MAX-SAT | Maximum Satisfiability |
| ML | Machine Learning |
| MRI | Magnetic resonance imaging |
| NASA | National Aeronautics and Space Administration |
| NIH | National Institutes of Health |
| NISQ | Noisy Intermediate Scale Quantum |
| NP | Nondeterministic Polynomial |
| NSF | National Science Foundation |
| OIM | Optical Ising Machines |
| OpenQASM | Open Quantum Assembly |
| ORNL | Oak Ridge National Laboratory |
| PCA | Principle Component Analysis |
| QA | Quantum Annealing |
| QABCS | Quantum Annealing based Binary Compressive Sensing |
| QAI | Quantum Artificial Intelligence |
| QAIL | Quantum Artificial Intelligence Laboratory |
| Qubit | Quantum Bit |
| QC | Quantum Computing |
| QCL | Quantum Computation Language |
| QDK | Quantum Development Kit |
| QISKit | Quantum information science kit |
| QMASM | Quantum macro assembler |
| QMI | Quantum Machine Intelligence |
| QPU | Quantum Processing Unit |
| QUBO | Quadratic Unconstrained Binary Optimization |
| RA | Reverse Annealing |
| RL | Reinforcement Learning |
| RQA | Reinforcement Quantum Annealing |
| RSA | Rivest–Shamir–Adleman |
| SAPI | Solver Application Program Interface |
| SAR | Synthetic Pperture Radar |
| SAT | Satisfiability (or Boolean Satisfiability) |
| SDK | Software Development Kit |
| SMT | Satisfiability Modulo Theory |
| SVM | Support Vector Machine |

TPU Tensor Processing Unit
TWR Through the Wall imaging Radar

# CHAPTER I

# Introduction

## 1.1 Motivation

Transitioning from vector-based computation, in central processing units (CPUs), to matrix-based computation resulted in emerging graphical processing units (GPUs) that has reshaped the landscape for accelerated computing, namely in the realms of scientific computing, high-performance computing (HPC), machine learning and big data analytics. In the same manner, custom design application-specific integrated circuits (ASICs) for extending computations (from vectors and matrices) to tensors (i.e., complex and higher-order objects), can repeat providing such a disruptive capability[76, 85]. As an example, the tensor processing unit (TPU), by Google, is an accelerator for near-real-time deep learning applications with low latency that has demonstrated throughput improvements of over 15–30 and power efficiency improvement of 30–70 over current CPUs and Kepler generation of GPUs, albeit lower precision computations [85]. After decades of predomination, nevertheless, the era of Moore's law draws to a close.

In post-Moore era, therefore, the supercomputing community is exploring non-Von-Neumann computing architectures for emerging next generation of processors. Neural processing units (NPUs), a.k.a. neuromorphic chips—namely the Neurosynaptic System by IBM, the SpiNNaker System by the University of Manchester, Intel's Loihi chip and memristors based systems—are (human) brain-inspired accelerators that have demonstrated a dramatic speedup in implementing artificial intelligence (AI) models, more specifically artificial neural networks (ANN) and deep networks. Although NPUs have demonstrated a notable speedup in terms of training deep networks and performing inference, compared to their CPU/GPU based counterparts, fixed connectivity of (artificial) neurons and low precision representation of synapses (e.g., 2-bit precision in IBM's TrueNorth system) restricts the applicability of a specific neuromorphic chip to a particular deep network architecture and precludes the use of well-nourished training methods (like stochastic gradient descent), respectively.

Quantum computing is a different information processing paradigm that takes the advantage of quantum mechanics—namely superposition and entanglement of quantum bits (qubits)—and promises new capabilities in performing computationally intensive tasks that are intractable in the realm of classical computing, ranging from problem-solving to sensing and communication. Owing to recent advances in the physical realization of quantum processing units (QPUs)—namely superconducting gate models, trapped ions, cold atoms and quantum annealers—quantum processors can play a crucial role in post-Moore era. There are several models for the physical realization of quantum computers—including, but not limited to, gate (or circuit) models and adiabatic quantum computers [101]. In gate model quantum computing, in addition to sparse connectivity of quantum circuits (at the device level) that limits the problem size and the gates that

can be implemented on a QPU, for noisy intermediate-scale quantum (NISQ) devices, noise restricts the size of circuits that can be implemented and executed on a physical QPU. Gate model quantum computing is well-nourished and there is a growing consensus of opinion that we can demonstrate the supremacy of quantum computing through the physical realization of the quantum circuits on (useful) applications. Adiabatic quantum computers, nevertheless, are currently more readily available at user sites due to recent advancements in commercializing programmable quantum annealers by D-Wave Systems [94].

Ising processing units (IPUs) or annealing processing units (APUs) are single instruction computing machines that can only search the Hilbert space of a given Hamiltonian to find the ground state (i.e., a configuration with the lowest energy value) of an Ising model. In the Ising model (a.k.a. spin glass model), pairwise correlations between interacting variables (or spins) can model complex probability distributions or represent computationally hard problems. Similar to NPUs and QPUs, IPUs—namely digital annealers by Fujitsu (based on FPGA and recently CMOS technologies), optical Ising machines (OIMs) at the University of Rome La Sapienza (based on photons) and D-Wave quantum annealers (based on superconducting qubits)—need to bypass low precision and noise barriers. In addition, the D-Wave quantum annealers are locally dense but globally sparse that limits their performance.

Thermal annealing (a.k.a. simulated annealing or classical annealing), that has to be implemented on CPUs/GPUs, can be very ineffective, compared to quantum annealing, because: (1) the landscape of the given Hamiltonian can be too glassy and there exist high energy barriers around local minimums that can trap the system for a very long time; and (2) a classical (or nonquantum) system can only assume one configuration at a time

while the number of configurations in discrete optimization problems (i.e., combinatorial optimization problems) can grow exponentially with the number of variables [49]. Quantum annealing is a meta-heuristic that (instead of thermal fluctuations) applies adjustable quantum fluctuations into a problem [4, 72, 95] and can bypass very high energy barriers, when they are narrow enough, which can address the ergodicity problem to some extent [154, 49, 161, 116, 162]. Moreover, at some stage of annealing, quantum annealing can see the whole landscape simultaneously (i.e., qubits are in a superposition of all eigen states) that can provide much faster relaxation to the ground state of the given Hamiltonian [49].

Quantum annealers, as the physical realization of quantum annealing, appear in the intersection of IPUs (i.e., they are a single-instruction machine that can only sample from the ground state of a given Ising Hamiltonian) and QPUs (i.e., they are a type of adiabatic quantum computer). From the quantum information processing viewpoint, although current generations of the quantum annealers, namely the D-Wave 2000Q QPUs, are not fully universal yet (i.e., quantum annealers are not fully resemblance of adiabatic quantum computers)—which can make them polynomially equivalent to other quantum computing models—several studies have demonstrated significant potentials in addressing real-world applications in the realm of quantum annealing. It is worth noting that, recent studies have offered to make quantum annealers universal [138].

Despite all aforementioned classical near-term accelerators, quantum annealers are potentially capable of providing the next disruption in post-Moore era, based on favorable aspects from both IPUs and QPUs. From an application perspective, while NPUs are designed for a specific AI model, quantum annealers can serve as a general-purpose computing paradigm through addressing combinatorial optimization problems and sampling from complex high-dimensional probability distributions in near-constant time. In the

same manner, in contrast to the limited size of quantum circuits, quantum annealers are easier to scale and we can employ them on real-world applications in the near-term. In this study, therefore, we aim to improve problem-solving with quantum annealers.

Quantum artificial intelligence and quantum machine learning are emerging fields that leverage quantum information processing to address certain types of AI problems that are intractable in the realm of classical computing [102, 20, 64]. Recent studies have revealed the potential of quantum annealing to address certain types of hard AI problems that are intractable in the realm of classical computing—including, but not limited to, planning [155], scheduling [179, 177], image registration [144, 143], discrete optimization problems [21, 117], constraint satisfaction problems [22], Boolean satisfiability [174, 11, 16], natural language processing [9] and training ANNs [29, 2, 171].

While most current research in Quantum artificial intelligence (and quantum machine learning) applies quantum computing models to hard AI problems, in this study, we hypothesized that we might apply AI techniques to improve the performance of quantum annealers. Current generations of the D-Wave quantum processors have some technological barriers—e.g., sparse connectivity between qubits, confined annealing schedule, coefficients' range and precision limitations, noise and decoherence—that lower the quality of results (i.e., energy value of the drawn samples is higher than the energy value of the ground state). To this end, we adopt ideas of reinforcement learning, ensemble techniques in machine learning, greedy algorithms and search heuristics/meta-heuristics to propose different models for improving the performance of the quantum annealers (in terms of finding samples with lower energy and making results of the quantum annealers reproduceable) in addressing problems with glassy landscapes.

Unlike conventional computing machines (namely classical and gate model quantum computers) that have a rich set of machine instructions, quantum annealers are single-instruction processors that can only sample from the ground state(s) of a given (Ising) Hamiltonian. More precisely, a D-Wave quantum processor receives coefficients of an Ising Hamiltonian (i.e., a quadratic objective function) as an executable quantum machine instruction (QMI) and samples from its ground state(s) at cryogenic temperatures. To solve a problem on a quantum annealer, therefore, one needs to cast the original problem of interest to a spin glass problem that is nontrivial for real-world applications. We offer to leverage Boolean satisfiability (SAT), as an intermediate representation of problems, and propose a systematic model that compiles classical algorithms (implemented in classical programming languages) and execute them on a quantum annealer. From a programming point of view, our model bridges the gap between classical algorithms and quantum programming. We demonstrate our approaches through two case studies, namely the problem of prime factoring and the $\ell_0$-norm problem of compressive sensing, both of which have been shown to be intractable in the realm of classical computing.

## 1.2 Thesis Statement

*Artificial intelligence can compile classical programs for Ising processing units, find samples with lower energy and improve the reproducibility of results, attained by the quantum annealers.*

## 1.3 Summary of Contributions

Major results and contributions made in this thesis are as follows:

(i) extended post-quantum error correction methods for quantum annealers and introduced a novel scheme that enhances the reproducibility of results and notably improves the quality of samples (i.e., finds samples with lower energy values), attained by quantum annealers;

(ii) proposed SAT++ as a novel quantum programming paradigm that bridges the gap between classical and quantum programming through compiling classical algorithms (implemented in classical programming languages) to quantum machine instructions that are executable by quantum annealers;

(iii) presented two AI hybrid quantum annealing approaches—so-called reinforcement quantum annealing and greedy quantum annealing—that conjugate both optimization and sampling aspects of the quantum annealers, and demonstrated that they can notably improve the probability of achieving the global optimum;

(iv) offered to address the $\ell_0$-norm problem of sparse recovery via reducing the original problem of compressive sensing to Weighted-MAX-SAT instances, mapping the $\ell_0$-norm problem of binary compressive sensing to quadratic unconstrained binary optimization (QUBO), and proposing to solve the original problems of binary compressive sensing and binary compressive sensing with matrix uncertainty with quantum annealers;

(v) introduced the theory of ensemble quantum annealing that can boost the performance of the quantum annealers in finding the ground state(s) of Ising Hamiltonians.

## 1.4 Thesis Organization

This dissertation shows how to leverage different aspects of artificial intelligence to not only improve the performance of the physical quantum annealers in addressing computationally intensive problems but also run classical algorithms on IPUs, namely the D-Wave quantum processors. The organization of this thesis is as follows. Chapter II briefly overviews the concepts of quantum computing, quantum annealers by D-Wave Systems and the problem of Boolean satisfiability (SAT). Chapter III presents novel error correction schemes for quantum annealers, in both qubit and problem variable domains, where we classically exploit samples (drawn by quantum annealers) to find a sample with lower Ising energy value. In this chapter, we experimentally demonstrate that applying the proposed post-quantum error correction techniques notably improves the probability of finding the ground state of Ising Hamiltonians—although we cannot guarantee that we can achieve the global optimum, specifically when the problems require dense connectivity (i.e., clique like problems). In addition, we show that applying the proposed post-quantum error correction techniques can significantly enhance the reproducibility of results.

In chapter IV, we introduce SAT++, a novel quantum programming paradigm, that enables us to execute classical algorithms on IPUs. As a proof-of-concept, we show how one can employ SAT++ to run classical prime factoring approach(s) on a D-Wave 2000Q quantum processor. In chapter V, we offer to conjugate both optimization and sampling aspects of the quantum annealers, and we introduce two novel hybrid quantum annealing approaches. we adopt the idea of reinforcement learning to introduce a novel model, called reinforcement quantum annealing (RQA). In the same manner, we introduce greedy quantum annealing (GQA) that conjugates both sampling and optimization aspects of

the quantum annealers. We then demonstrate that both RQA and GQA can find samples with notably lower energy values, compared to the best-known techniques in the realm of quantum annealing, specifically when the problem requires a dense connectivity.

In chapter VI, as the second study case, we address the NP-Hard problem of compressive sensing (i.e., the $\ell_0$-norm problem of sparse recovery). We first introduce novel methods for reducing the original problem of compressive sensing to Weighted-MAX-SAT, defining QUBO form of the $\ell_0$-norm problem of binary compressive sensing, and solving problems of binary compressive sensing and binary compressive sensing with matrix uncertainty on a quantum annealer. Afterward, we demonstrate that applying our proposed methods in this thesis results in a better sparse recovery. Finally, chapter VII presents conclusions and future works of this study.

# CHAPTER II

# Background and Related Work

## 2.1 Quantum Computing

In conventional computing, data are encoded as a sequence of (classical) bits (a.k.a. Shannon bit or binary digit) that take their values from $\{0, 1\}$ and a register with $n$ bits at time $t$ can be in only one of the $2^n$ possible states. Solving a problem on a classical computer (i.e., any model of Turing machine) requires a sequence of instructions that follows the control flow of a classical algorithm. When the number of required instructions to finish a computation grows exponentially with the problem size, the corresponding problem becomes intractable in the realm of classical computing.

Quantum computing is a novel approach for tackling computationally intensive problems that are intractable in the realm of classical computing [136, 107]. A quantum bit (qubit) is the fundamental object of quantum information processing. Qubits are two-level quantum systems and we define the state of a qubit in a two-dimensional Hilbert space as follows:

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \alpha|0\rangle + \beta|1\rangle \tag{2.1}$$

where $\alpha, \beta \in \mathbb{C}$ are coefficients of the computational basis (a.k.a. eigen states), denoted by $|0\rangle$ and $|1\rangle$. State of a qubit can be $|0\rangle$, $|1\rangle$—analogous to "0" and "1" in classical computing, respectively—or any superposition of the computational basis such that

$$|\alpha|^2 + |\beta|^2 = 1. \tag{2.2}$$

Quantum gates (a.k.a. quantum operations) are (square) unitary matrices that can emulate any rotation of the quantum state. Quantum computers apply quantum gates on qubits—i.e., multiplying the transform matrix of the corresponding quantum gate by the state vector of qubits—for quantum information processing. Tensor product of $n$ qubits forms a quantum register, denoted as:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle, \tag{2.3}$$

which includes $2^n$ eigen states. Similarly, tensor product of single-qubit quantum gates can result in multi-qubit quantum operations. After performing the measurement on qubits of a quantum system, the state of the quantum register is collapsed to one of the computational basis, with the corresponding probability. As an example, measuring a single qubit results in $|0\rangle$ and $|1\rangle$ with the probability of $|\alpha|^2$ and $|\beta|2$, respectively. Having qubits in their superposition and taking the advantage of their entanglement, we can apply single-qubit quantum operations (e.g., Hadamard, Phase and rotation) and two-qubit quantum operations (like conditional NOT or CNOT) to implement quantum algorithms and achieve the quantum supremacy [108].

### 2.1.1 Quantum Computing Models

After introducing the concept of quantum computing by Richard Feynman in 1959, several models have been proposed for the realization of quantum information processing. Although these models are theoretically equivalent, their underlying concepts, as well as realization requirements, are significantly different from each other. Circuit models in quantum computing are closely analogous to classical (i.e., transistor-based) computers in which, a sequence of quantum operations (or gates) on quantum registers process the quantum information and emulates the quantum algorithms [108]. Unlike the majority of the classical gates that are irreversible, quantum gates (except the measurement operation) are reversible. In gate model quantum computing, the circuit must have enough coherence time for running the quantum algorithms [108].

In measurement-based quantum computing (also known as one-way quantum computing), rounds of measurements on qubits—prepared in a highly entangled state (cluster or graph state)—performs the quantum information processing. In this model, results from previous measurements define the basis for current measurements to guarantee that the computation always succeeds [184, 33].

Adiabatic quantum computers rely on *Quantum Adiabatic Evolution* for quantum information processing. According to the *Adiabatic Theorem*, adiabatic quantum computers are polynomially equivalent to the circuit models. Unlike previous models in quantum information processing, qubits of adiabatic quantum computers do not perform discrete operations. Indeed, adiabatic quantum computers receive a Hamiltonian (also called energy function) as input whose ground state represents the solution for the problem that we

are trying to solve. Afterward, qubits are adiabatically evolved from some initial unknown state to a final state that minimizes the energy function [70].

## 2.2 D-Wave Quantum Annealers

The quantum processing unit (QPU), by D-Wave Systems, is a programmable quantum annealer that samples from the ground state(s) of a given Ising Hamiltonian at cryogenic temperatures [94]. From a problem-solving point of view, the D-Wave quantum annealers receive coefficients of an Ising Hamiltonian as an executable quantum machine instruction (QMI), here $\mathbf{h}$ and $J$, and return the ground state of the following quadratic energy function:

$$E_{\text{Ising}}(\mathbf{z}) = \sum_{i=1}^{N} \mathbf{h}_i \mathbf{z}_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} J_{ij} \mathbf{z}_i \mathbf{z}_j, \tag{2.4}$$

where $N$ denotes the number of quantum bits (qubits). In this representation, $\mathbf{z}_i \in \{-1, +1\}$. One can apply a linear transform to map Eq. (2.4) to its equivalent quadratic unconstraint binary optimization (QUBO) form, and vice versa, as follows:

$$E_{\text{QUBO}}(\mathbf{x}) = \sum_{i \leq j}^{N} \mathbf{x}_i Q_{ij} \mathbf{x}_j, \tag{2.5}$$

where $\mathbf{x} \in \{0, 1\}^N$ and $i, j \in \{1, 2, \cdots, N\}$. In this representation, $Q$ includes both linear biases and quadratic couplers—analogous to $\mathbf{h}$ and $J$ in Eq. (2.4), respectively.

The current generation of the D-Wave quantum annealers (a.k.a. the Chimera architecture) includes more than 2,000 qubits and about 6,000 couplers, while the next generation (the Pegasus topology of the Advantage) will include more than 5,000 qubits and about 40,000 couplers [30]. Unlike conventional computing machines (namely classical and gate model quantum computers) that have a rich set of machine instructions, quantum

annealers are single-instruction computing machines that can only minimize Eq. (2.4) or (2.5). To solve a problem on a quantum annealer, therefore, one needs to define coefficients of an Ising Hamiltonian whose ground state represents a solution of the original problem of interest.

### 2.2.1 Trends in Applying Quantum Annealers

One can reduce any problem of class NP to an NP-complete problem in polynomial-time [79, 15]. Thus, we can employ quantum annealers to address (significant) real-world problems that are intractable in the realm of classical computing. Recent studies have revealed the potential of quantum annealing to address certain classes of real-world problems that are intractable in the realm of classical computing [29]. In particular, the D-Wave quantum annealer has been applied to, planning [155], scheduling [179, 177], image registration [144, 143], data assimilation and processing [133, 84], discrete optimization problems [21, 117], constraint satisfaction problems [22], Boolean satisfiability [174, 11, 16], matrix factorization [141], linear least squares problems [31], cryptography [146, 91], fault detection and system diagnosis [148], compressive sensing [17, 13], natural language processing [9], protein folding [147] and material design [100]. In addition, by sampling from high-dimensional probability distributions, one can use the D-Wave quantum annealer for many applications in artificial intelligence, machine learning and signal processing [29, 2, 171].

### 2.2.2 Limitations

Beside all aforementioned applications, the D-Wave quantum annealer architecture has limitations that not only restrict the process of mapping problems into an executable

QMI—embedding Ising Hamiltonians to the working graph of a D-Wave QPU is non-trivial—but also lower the quality of results—i.e., the energy value of resulting samples (attained by the quantum annealer) is higher than the ground state of the given Ising Hamiltonian. From another perspective, for a given QMI, the D-Wave QPU draws samples from a problem-dependent pseudo-Boltzmann distribution at cryogenic temperatures [29]. The energy values of samples from the D-Wave QPU follow a Gaussian distribution. Thus, when we increase the number of reads/samples, we expect that the average parameter in the corresponding Gaussian distribution approaches the ground state energy of the corresponding Ising Hamiltonian—i.e., the probability of finding the global minimum approaches one. There are several drawbacks, nevertheless, that prevent quantum annealers from attaining a global minimum.

From a quantum computing perspective, an adiabatic quantum computer needs to minimize a non-stoquastic Hamiltonian in order to be universal (which would make them equivalent to gate models); nevertheless, the D-Wave QPU minimizes an Ising Hamiltonian which is stoquastic [138, 180, 118]. The required anneal time in a quantum annealer to keep the process adiabatic has a reverse exponential relation to the energy gap between the ground state (global minimum) and the first excited state (a state right above the global minimum) [138]. In current generations of the D-Wave QPUs, any executable QMI obeys:

$$-2 \leq \mathbf{h}_i \leq +2 \tag{2.6}$$

and

$$-1 \leq J_{ij} \leq +1, \tag{2.7}$$

where $i, j \in \{1, 2, \ldots, N\}$ and $i < j$. Thus, one needs to scale the resulting Ising Hamiltonian, shown in Eq. (2.4), by dividing all coefficients with a large-enough positive number to satisfy the QPU hardware constraints. Scaling the Ising model, however, reduces the energy gap between the ground and the first excited states. As a result, the required annealing time can quickly exceed the maximum possible anneal time on a physical quantum annealer (for example 2,000 micro-seconds on the D-Wave QPUs) and makes the process diabatic, which exponentially reduces the probability of getting to the ground state [138, 118].

Besides, the current generation of the D-Wave QPUs uses 8–9 bits for representing coefficients in Eq. (2.4). Hence, the D-Wave QPU truncates coefficients of a QMI prior to putting qubits in their superposition, which can result in the Ising model having a different ground state—compared to the original QMI. Consequently, the D-Wave QPU may solve a different problem whose result is either infeasible or less accurate than the original problem of interest [150, 59]. Dorband (2018) has proposed a heuristic that addresses the precision issue on the D-Wave QPUs, albeit executing many QMIs for one problem [59].

Since coupling every qubit to every other qubit in a quantum annealer is impractical, the D-Wave QPU has a sparse connectivity structure/topology. Figures 2.1 and 2.2 illustrate the Chimera (in D-Wave 2000Q system) and Pegasus (in D-Wave Advantage system) topologies, respectively.

Hence, we entangle multiple qubits to represent virtual qubits with higher connectivity. Chaining physical qubits substantially reduces the capacity of QPUs—e.g., 2,048 qubits in the Chimera architecture is equivalent to a clique of size 64. It is possible to implicitly leverage the capacity of the current D-Wave QPUs [140], albeit executing multiple QMIs. In addition, virtual qubits are vulnerable to breaking—the longer the chains,

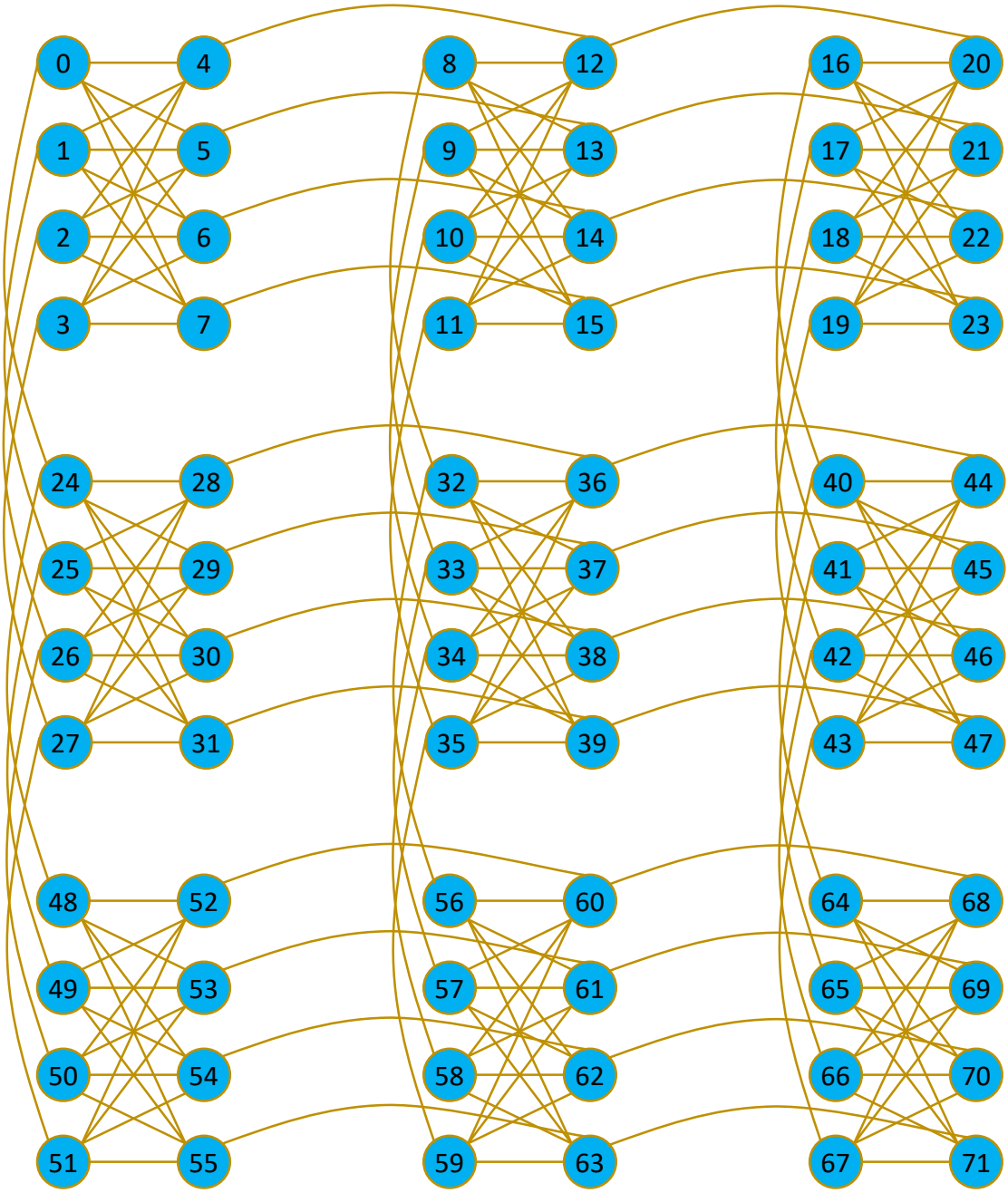


Figure 2.1: Chimera topology of the D-Wave 2000Q.

the higher the probability they break during the annealing process. Although we can remediate broken chains by applying postprocessing methods on classical computers (e.g., voting among the physical qubits on a chain), some chains break because they represent a state with lower energy.

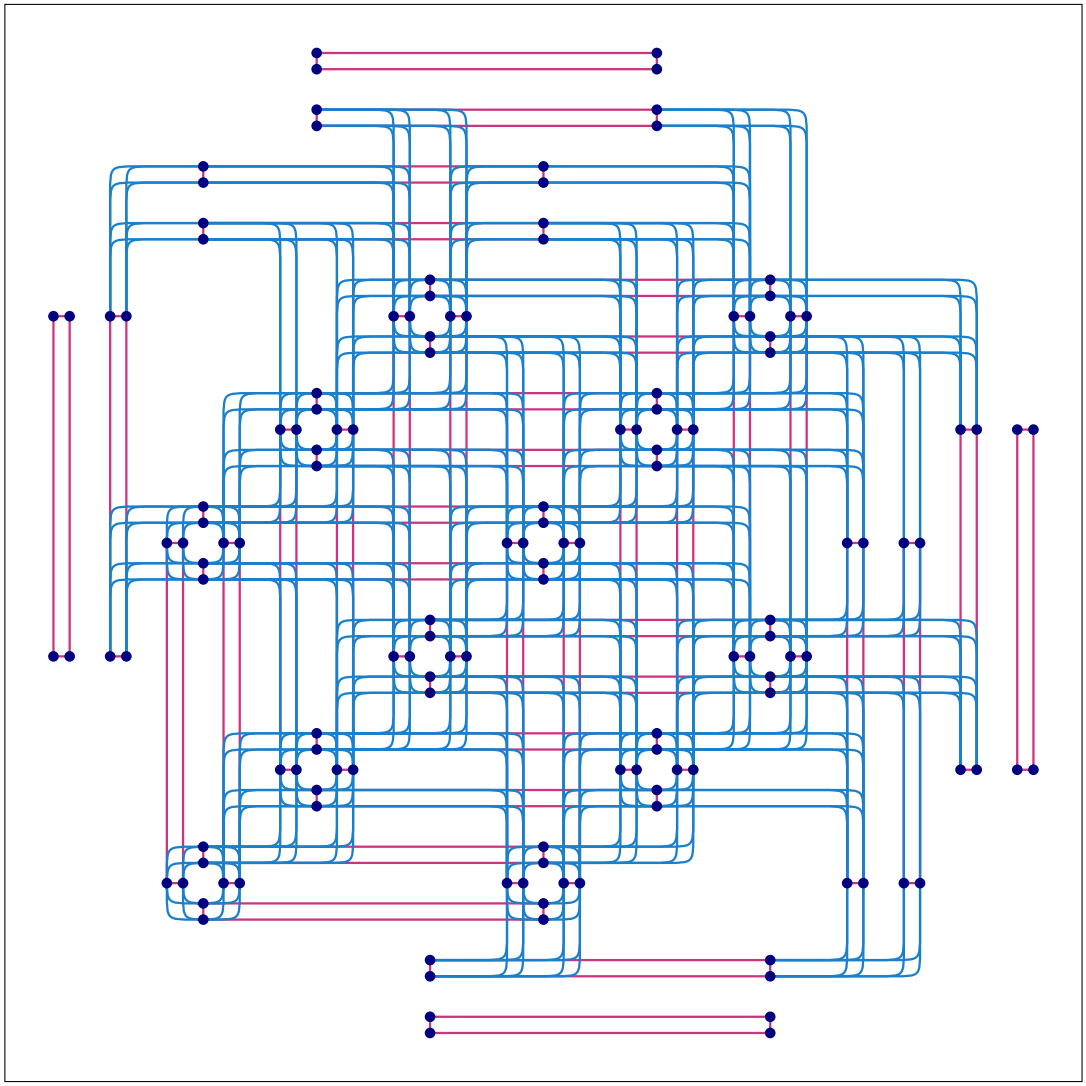

Figure 2.2: Pegasus topology of the D-Wave Advantage where curved blue lines, long red lines and short red lines are internal, external and odd couplers, respectively [30].

Applying preprocessing techniques [145] and classical postprocessing heuristics [60] can remarkably enhance the performance of the D-Wave QPU; however, from a problem-solving viewpoint, the D-Wave quantum annealer cannot guarantee to achieve a global optimum.

## 2.3 Boolean Satisfiability (SAT)

Let $f(\mathbf{x})$ be a Boolean formula (a.k.a. Boolean expression or propositional logic formula) over the binary variables (atoms) $\mathbf{x}_i \in \{\top, \bot\}$—denoting "True" (or "1") and "False" (or "0"), respectively—with logical operators—including, but not limited to, "AND" ($\wedge$), "OR" ($\vee$) and "NOT" ($\neg$). The problem of Boolean satisfiability (a.k.a. propositional satisfiability, satisfiability or SAT) aims to find $\mathbf{x}$, as a model for $f$, such that:

$$\mathbf{x} \models f.$$

From a problem-solving viewpoint, SAT is a decision problem that determines whether a constant replacement of values for all Boolean variables can interpret $f$ as $\top$ [27]. The Boolean formula $f$ is in conjunctive normal form (CNF) if:

$$f(\mathbf{x}) = C_1 \wedge C_2 \wedge \cdots \wedge C_M,$$

where the clause $C_i$ (for $i = 1, 2, \ldots, M$) is a disjunction of literals as follows:

$$C_i = \mathbf{l}_j \vee \mathbf{l}_k \vee \ldots,$$

and the literal $\mathbf{l}_i$ represents the corresponding Boolean variable ($\mathbf{x}_i$) or its negation ($\neg\mathbf{x}_i$) [27].

### 2.3.1 Extensions of SAT

From a complexity perspective, SAT is in class NP-complete [46]—one of the Karp's 21 NP-complete problems [97]; therefore, we can reduce any NP problem to SAT in

polynomial-time, and vice versa [79]. Real-world problems, nevertheless, have complex features that make the use of modern SAT solvers challenging. Therefore, various extensions of SAT have been introduced to provide different perspectives to the search-oriented nature of the original SAT (e.g., optimization and model counting) [27, 115].

The maximum satisfiability problem (MAX-SAT) is an NP-hard extension of the SAT problem that determines the maximum number of satisfiable clauses. From a problem formulation attitude, the MAX-SAT adds an optimization perspective to the original SAT that has a searching nature. Weighted-MAX-SAT is an extended version of the MAX-SAT in which, each clause has an associated weight that specifies the penalty of not satisfying the corresponding clause [27].

The #SAT problem (a.k.a. Sharp-SAT) is the problem of counting the number of satisfying assignments of a given Boolean formula [93]. The Parity-SAT problem is a special extension of the #SAT in which, we are trying to determine whether a given Boolean expression has odd numbers of satisfying assignments. In Unique-SAT problem we aim to determine whether a given Boolean formula has exactly one satisfying assignment. On the other side, the Tautology problem determines whether a given logic formula is satisfiable for all possible assignments [181].

Similar to the propositional logic (zeroth-order logic), we can extend the satisfiability problem to satisfy expressions in higher-order logics—including, but not limited to, the first-order logic (i.e., variables are quantified over the domain) and second-order logic (i.e., both variables and operators are quantified) [75]. Constraint satisfaction problems (CSP) represent problems as a set of constraints, and CSP solvers search the state space to find a feasible solution [157]. Although SAT can represent all NP problems, CSP provides a straighter forward perspective for formulating real-world problems.

Integer programming (IP) restricts variables' domain in CSP to integer numbers. In a same manner, 0-1 integer programming restricts variables to take their values from $\{0, 1\}$ [164]. From a problem formulation perspective, 01- integer programming generalizes the SAT to include complicated constraints, and we can reduce problems in 0-1 integer programming to SAT in polynomial time, and vice versa.

### 2.3.2 SAT Solvers

From a problem-solving perspective, the search space of SAT (which is in class NP-complete) can grow exponentially. The majority of the modern SAT solvers, therefore, have adopted back-tracking techniques, like Davis—Putnam—Lagemann—Loveland (DPLL) algorithm, for pruning the state space of given problems [83, 137]. Conflict-driven clause learning (CDCL) [65] is a modern variation of the DPLL method that serves the core functionality in developing recent modern SAT solvers—including (but not limited to) MiniSAT [173, 172], PMSat (parallel version of the MiniSAT) [80], Glucose [7, 8], Lingeling and Plingeling [26]. Modern SAT solvers have started to leverage AI and machine learning to enhance their performance. As an example, classifiers and deep neural architectures have shown that they can recognize unsatisfiable clauses (and sub-formulas) which can provide notable enhancements in the performance of the back-tracking based SAT solvers [54, 34, 185].

We can look at the satisfying a Boolean expression as a search (or optimization) problem and employ local and global search/optimization heuristics and meta-heuristics for SAT-based applications [73]. As an illustration, "Walk SAT" is a stochastic local search heuristic that toggles the value(s) of randomly selected variable(s) from unsatisfied clauses in each iteration. Although "Walk SAT" seems to be a naïve solver, it has outper-

formed many high-performance SAT solvers [166]. In the same way, we can use simulated annealing for solving MAX-SAT instances [32]. Evolutionary algorithms (EA) and swarm intelligence methods have also demonstrated that they can address SAT and MAX-SAT problems efficiently [82, 57, 56].

Several studies have addressed SAT (and its extensions) in quantum regime. In the realm of gate model quantum computing, Grover's algorithm and its generalization (Amplitude amplification) are oracular quantum algorithms that can provide quadratic speed up for any problem in class NP [124]. Grover's algorithm solves the SAT in $O\left(\sqrt{2^n}\right)$ which is trivial because the best classical algorithm for 3-SAT runs more than quadratically faster than brute force search. Amplitude amplification, nevertheless, provides quadratic speedup over the best classical 3-SAT algorithm [5]. A similar study has proposed a quantum algorithm that solves the exact SAT problem in $O(2^{n-M'})$ (where $M'$ is the number of linearly independent clauses) which is polynomially faster than the best known classical SAT algorithm. [113]. A nested quantum search algorithm (analogous to the classical tree-based structural search) has shown to outperform the original Grover's algorithm in CSP and speeds up the best corresponding classical algorithms polynomially [39]. A generalized method has also demonstrated to obtain quantum speed-up in addressing CSP over backtracking based classical algorithms [123].

On the other side of the quantum world, adiabatic quantum computing model has demonstrated a notable potential in solving SAT instances [88, 191]. Similarly, several studies have employed quantum annealers for addressing SAT (and its extensions). For $k \geq 3$, $k$-SAT is NP-complete; however, weighted-Max-SAT with positive waits is NP-hard for $k = 2$. We can directly map the weighted-MAX-2-SAT instances to a QUBO representation [24]; however, for $k > 2$ these reductions define additional ancillary variables

that quickly exceed the capacity of current physical quantum annealers. In addition to adopting the idea of penalty methods for solving SAT and MAX-SAT instances on quantum annealers [23, 40], we can employ Ising primitives to reduce SAT to an Ising Hamiltonian whose ground state represents a satisfying solution for the original SAT [174].

### 2.3.3 SAT Applications

Solving satisfiability instances, in the worst-case scenario, can require exponentially large computational resources; nevertheless, worst cases are less likely to appear in the majority of the real-world applications. In addition, owing to the recent advances in developing efficient SAT solving heuristics, modern SAT solvers can handle SAT instances with thousands of variables and millions of clauses [80]. Hence, several real-world applications leverage modern SAT solvers to address significant problems [114].

Modern SAT solvers and their extensions (like MAX-SAT, SMT and CSP solvers) play crucial functionality in some commercial planning applications [98, 156]. Similarly, SAT solvers have shown desirable performance in practical scheduling applications [90, 45]. Since we can transfer first-order logic expressions to propositional logic formulas [158], SAT serves as the cornerstone for a vast range of AI applications—more precisely, reasoning and inference systems [165]. As an example, Z3 is an efficient SMT solver that can satisfy higher-order expressions [51]. Furthermore, many applications in electronic design automation (EDA) have adopted SAT solvers to tackle significant problems in the industry—including (but not limited to) combinational equivalence checking, logic optimization, functional test vector generation, test pattern generation, circuit delay computation and bounded model checking [115]. Satisfiability has also started to address various problems

in cryptography. As an illustration, in 2005, high-performance SAT-solvers were able to break Several standard cryptographic hash functions [120]. Similarly, parallel SAT solvers have tried to tackle the problem of prime factorization that plays crucial functionality in modern public-key cryptography systems—namely the RSA [110].

# CHAPTER III

# Post-Quantum Error Correction for Quantum Annealers

## 3.1 Introduction

We can form an Ising Hamiltonian whose ground state represents the optimum solution of any given problem of interest—which can be nontrivial in many real-world applications. In practice, however, executing the corresponding QMI on a physical quantum annealer does not guarantee to achieve the global optimum [16]. Current generations of the D-Wave quantum processors, as an example, have some technological barriers—including, but not limited to, sparse connectivity between qubits [35], confined annealing schedule [138], coefficients' range and precision limitations [150, 59], noise and decoherence [53, 78, 77]—that lower the quality of results (i.e., the energy value of the drawn samples is higher than the energy value of the ground state).

Applying post-processing heuristics can improve the quality of samples—i.e., we can obtain a sample(s) with lower energy value [60]. As an example, after drawing samples by a quantum annealer, we can apply local search heuristics (on classical computers) to exploit the neighborhood of samples. Adopting the hill-climbing method [68, 158] for

addressing the spin glass problem results in a local search heuristic—so-called single qubit correction (SQC), presented in algorithm 3.1.1—that has demonstrated to improve the quality of results, attained by the D-Wave quantum annealers [60, 10].

**Algorithm 3.1.1:** Single qubit correction (SQC) heuristic for relaxing an input sample to a sample with lower energy value.

**Input:** $\mathbf{z}, \mathbf{h}$ and $J$
**Output:** $\mathbf{z}^*$

```
1  N ← |z|
2  z* ← z
3  do
4  |  z ← z*
5  |  for i ← 1 to N do
6  |  |  z_i ← −z_i
7  |  |  if E_Ising(z*, h, J) > E_Ising(z, h, J) then
8  |  |  |  z* ← z
9  |  |  end
10 |  end
11 while E_Ising(z*, h, J) > E_Ising(z, h, J)
12 return z*
```

From an optimization viewpoint, SQC has a local search scheme so it is very likely to converge to a local optimum—i.e., SQC is very sensitive to the initial point. To solve a problem on a quantum annealer, we generally draw several samples (up to 10,000 samples/reads per QMI). Therefore, in practice, SQC explores a broader area (i.e., the neighborhood of all samples that have been drawn from a ground state of the given Ising Hamiltonian).

Multi-qubit correction (MQC) is another postprocessing method that tries to address error correction of quantum annealers more from a results improvement perspective than a qubit value correction point of view [60]. Similar to SQC, MQC is a (classical) postprocessing approach that we apply on drawn samples by a quantum annealer. Thus, MQC does not depend on the hardware architecture of the target quantum processors.

In addition, unlike the majority of the gate model error correction techniques, MQC does not utilize ancillary qubits for error correction purposes.

MQC assumes that quantum annealers can recognize the region(s) of the ground state(s) of the given Ising Hamiltonians efficiently; however, they can fail to get to the global minimum, regardless of how close they are to the ground state. While SQC processes qubits of a sample individually to find a new sample with lower energy value, MQC compares two samples at a time to recognize tunnels (groups of qubits) where we can flip their states simultaneously to find a new sample whose energy value is less than or (in the worst case) equal to the energy value of the input samples. For a given Ising Hamiltonian—denoted by its linear and quadratic coefficients, $\mathbf{h}$ and $J$, respectively—algorithm 3.1.2 shows how MQC reduces two input samples, denoted by $\mathbf{z}^1$ and $\mathbf{z}^2$, to a new sample (denoted by $\mathbf{z}^*$) whose energy value is guaranteed to be less than or equal to energy values of $\mathbf{z}^1$ and $\mathbf{z}^2$ [60].

In fact, the Reduce function, shown in algorithm 3.1.2, is the extended version of SQC method, shown in 3.1.1, that acts on a group of qubits simultaneously rather than flipping the values of qubits individually. Algorithm 3.1.3 illustrates the MQC method that receives a sample set as input, and tries to iteratively reduce it to a (new) sample with lower energy value. MQC can find notably better solutions (i.e., samples with lower energy value) [60]. For a sample set with $n$ samples/reads, MQC takes $\log_2(n)$ steps and in each iteration, the size of the sample set is divided by two. It is worth noting that MQC performs better functionality when we use raw samples of a D-Wave quantum annealer. Recent studies have demonstrated that MQC can significantly improve the quality of results, specifically when the problem has a Chimera-like structure [60, 61, 81]. However,

**Algorithm 3.1.2:** Reducing two input samples to a new sample with lower energy value based on virtual tunnels.

**Function** `Reduce`($\mathbf{z}^1$, $\mathbf{z}^2$, $\mathbf{h}$, $J$)**:**
  $\mathbf{z}^* \leftarrow \mathbf{z}^1$
  $N \leftarrow |\mathbf{z}^*|$
  $S \leftarrow \{\}$
  $D \leftarrow \{\}$
  **for** $i \leftarrow 1$ **to** $N$ **do**
    **if** $\mathbf{z}_i^1 = \mathbf{z}_i^2$ **then**
      $S \leftarrow S \cup \{i\}$
    **else**
      $D \leftarrow D \cup \{i\}$
    **end**
  **end**
  $Adj \leftarrow [\,]$
  **for** $i \in D$ **do**
    $Adj_i = \{\}$
  **end**
  **for** $i, j \in J$ **do**
    **if** $J_{ij} \neq 0$ & $i \in D$ & $j \in D$ **then**
      $Adj_i \leftarrow Adj_i \cup \{j\}$
      $Adj_j \leftarrow Adj_j \cup \{i\}$
    **end**
  **end**
  $T \leftarrow$ `ConnectedComponents`($Adj$)
  **for** $k \leftarrow 1$ **to** $|T|$ **do**
    $I^k_{\mathbf{z}^1} = \sum_{i \in T^k} \mathbf{h}_i \mathbf{z}_i^1 + \sum_{i \in T^k} \sum_{j \in S} J_{ij} \mathbf{z}_i^1 \mathbf{z}_j^1$
    **if** $I^k_{\mathbf{z}^1} > 0$ **then**
      **for** $l \in T^k$ **do**
        $\mathbf{z}_l^* \leftarrow -\mathbf{z}_l^*$
      **end**
    **end**
  **end**
**return** $\mathbf{z}^*$

the performance of MQC approaches to the pure quantum annealing when the structure of the problem approaches to a clique (i.e., we need longer chains for embedding) [10, 16].

**Algorithm 3.1.3:** Multi-qubit correction (MQC) method for reducing a set of samples to a new sample whose energy value is less than all input samples or (in worst case) is equal to the lowest energy value of input samples.

**Input:** $Z = \left[\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^n\right], \mathbf{h}$ and $J$
**Output:** $\mathbf{z}^*$

```
while |Z| > 1 do
    Ẑ ← Z
    Z ← {}
    while |Ẑ| > 1 do
        z^A ← Ẑ.Pop()
        z^B ← Ẑ.Pop()
        z^AB ← Reduce(z^A, z^B, h, J)
        Z.Append(z^AB)
    end
    if |Ẑ| > 0 then
        Z.Append(Ẑ.Pop())
    end
end
z* ← Z.Pop()
```

## 3.2 Benchmarking the D-Wave Quantum Annealers

Although quantum annealers can draw very high-quality samples in near-constant time—i.e., the energy value of drawn samples approaches to the energy value of the ground state of the given Ising Hamiltonian—they generally fail to attain the global optimum due to various technological barriers. In this section, we evaluate recent software and hardware advances that aim to address the drawbacks of quantum annealers in finding the ground state of Ising Hamiltonians. As a benchmark, we generated three different types of Ising Hamiltonians as follows:

- random Ising Hamiltonians whose linear and quadratic coefficients were randomly drawn from $\{-1, +1\}$, based on a Bernoulli distribution with equal probabilities for -1 and +1, denoted by "binary";

- random Ising Hamiltonians in which all coefficients are uniform random numbers in $[-1, +1]$, denoted by "uniform";

- random Ising Hamiltonians whose coefficients follow the standard Gaussian distribution—average and standard deviation were 0 and 1, respectively, denoted by "normal".

All randomly generated benchmark problems were compatible with the current working graph of the D-Wave QPU (so-called Chimera topology). Therefore, until the next maintenance that can change the working graph of the quantum annealer, one can directly execute them without embedding problems to a target graph. For each problem type, we generated 50 instances (random problems).

As our primary ground truth method, we use raw samples, denoted by $\mathrm{QA}^1$, drawn by a D-Wave 2000Q quantum processor. Recent studies have revealed that using spin-reversal transforms (a.k.a. gauge transforms)—i.e., flipping the qubits randomly without altering the ground state of the original Ising Hamiltonian—can reduce analog errors of the quantum annealers [145]. Thus, the second method in this benchmarking (denoted by $\mathrm{QA}^2$), enabled 5 (five) spin-reversal-transforms before submitting QMIs to be executed on a D-Wave QPU. Since quantum annealers samples from the ground state(s) of the given Ising Hamiltonians, when we submit a problem to a D-Wave quantum annealer, it is a common practice to request for several samples/reads (up to 10,000 per QMI). For every read, the D-Wave QPU initializes all qubits and repeats the annealing process. The

third method in this benchmarking study, denoted by $QA^3$, puts a longer delay between successive reads/samplings to reduce the sample-to-sample correlation, albeit longer runtime.

Since executing a QMI on a quantum annealer is not guaranteed to achieve the ground state of the corresponding Ising Hamiltonian, even if we request for many samples/reads, applying classical postprocessing heuristics on raw samples (attained by the quantum annealers) can result in samples with lower energy values. Therefore, our fourth method in this benchmarking, denoted by $QA^4$, performs optimization postprocessing on all samples, which is available in D-Wave's Ocean SDK. In the same manner, in $QA^6$ we performed sampling postprocessing (which is also available in D-Wave's Ocean SDK) on raw samples. In $QA^6$ we: (1) enabled 5 (five) spin-reversal-transforms; (2) put a longer delay between successive reads; and (3) performed optimization postprocessing on all samples.

### 3.2.1 Experiment A: Applying SQC

In this experiment, we aim to study the characteristics of the drawn samples by the D-Wave 2000Q quantum annealers. To this end, in $QA^1 + SQC$, we applied the SQC local search heuristic, shown in algorithm 3.1.1, on all raw samples, attained by $QA^1$. Figure 3.1 illustrates the performance comparison between $QA^1$ and $QA^1 + SQC$ in minimizing the benchmark problems (i.e., random binary, normal and uniform Ising Hamiltonians).

### 3.2.2 Experiment B: Applying MQC

Recent studies have demonstrated that applying MQC on drawn samples by a D-Wave quantum annealer can notably improve the quality of results [60, 81, 61, 10, 16, 14].

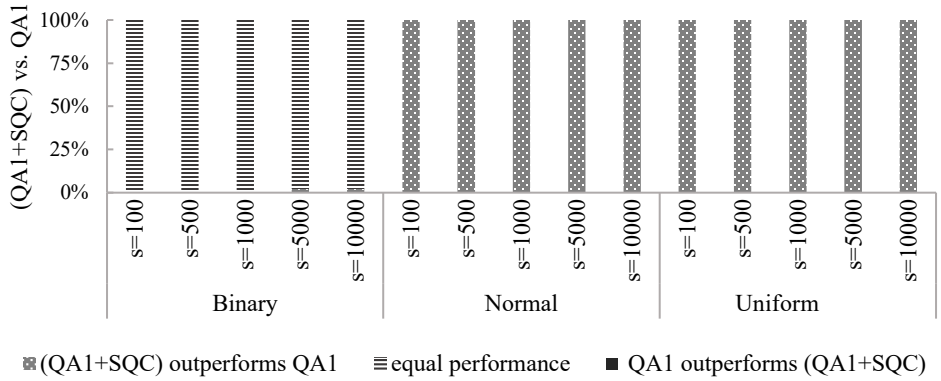


Figure 3.1: Performance comparison between $\mathrm{QA}^1$ and $\mathrm{QA}^1 + SQC$ on minimizing 50 random (binary, normal and uniform) Ising Hamiltonians with different sample sizes.

In this experiment, we aim to evaluate the performance of MQC method, shown in algorithm 3.1.3, and compare it with recent software and hardware advances in quantum annealing. To this end, in $\mathrm{QA}^1 + \mathrm{MQC}$, we applied the MQC method, shown in algorithm 3.1.3, on all raw samples, attained by $\mathrm{QA}^1$. Figure 3.2 compares the performance of $\mathrm{QA}^1 + MQC$ with the abovementioned arrangements in our benchmarking—namely $\mathrm{QA}^1, \ldots, \mathrm{QA}^5$ and $\mathrm{QA}^6$—in minimizing the benchmark problems (i.e., random binary, normal and uniform Ising Hamiltonians).

Figure 3.2 demonstrates that MQC outperforms all aforementioned software and hardware advancements. In other words, in majority of cases, MQC was able to find a sample with lower energy value. In the same way, Fig. 3.3 displays the performance comparison between applying SQC and MQC on raw samples of a D-Wave quantum annealer, denoted by $\mathrm{QA}^1 + \mathrm{SQC}$ and $\mathrm{QA}^1 + \mathrm{MQC}$, respectively.

### 3.2.3 Experiment C: Reproducibility of Results

From an application perspective, in addition to relaxing to the excited states of a given Hamiltonian (instead of its ground state), results/samples attained by the physical quantum annealers are not reproducible. According to Anderson localization [3], as an

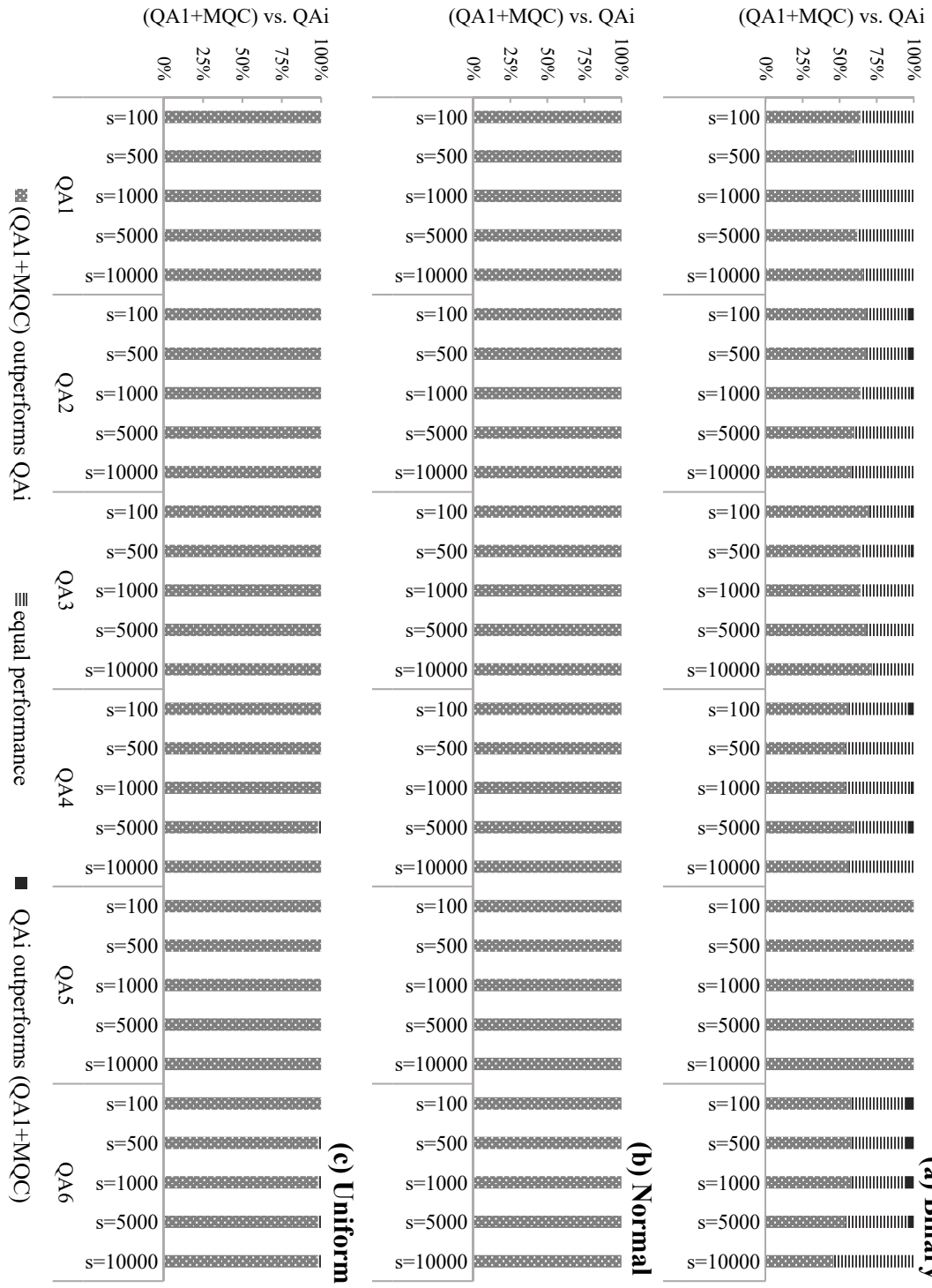


Figure 3.2: Comparing the performance of $\mathrm{QA}^1+MQC$ with recent software and hardware advances in the realm of quantum annealing.

example, the energy gap between the ground and first excited states is shrunk close to the end of the annealing. The landscape of glassy Hamiltonians generally includes many

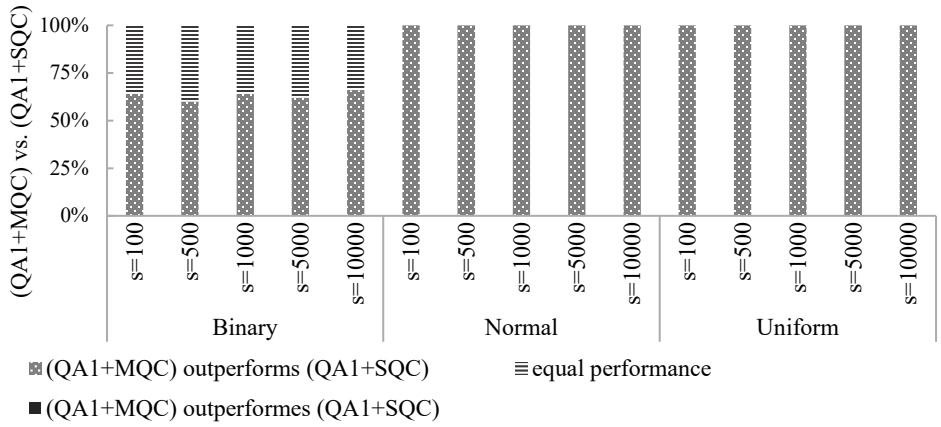


Figure 3.3: Comparing the performance of $\mathrm{QA}^1 + \mathrm{SQC}$ with $\mathrm{QA}^1 + \mathrm{MQC}$.

excited states and a physical quantum annealer is likely to relax to one of these excited states. As a conclusion, problem-solving with quantum annealers results in a distribution of (potential) ground states, and variance of the corresponding energy values are large enough to lessen the reproducibility of results.

In this experiment, we aim to demonstrate that applying post-quantum correction techniques can notably enhance the reproducibility of results, attained by the quantum annealers. To this end, for a random binary, normal and uniform problem, we repeated the problem-solving 50 times. For each method, table 3.1 shows the variance of the energy values of the resulting 50 samples. Table 3.1 demonstrates that applying MQC makes results of quantum annealers significantly reproducible, compared to recent software and hardware advances in quantum annealing.

## 3.3 Randomized Multi-Qubit Correction

In section 3.2.2 we showed that applying MQC on raw samples (drawn by a D-Wave quantum processor) can outperform recent software and hardware advancements in quantum annealing. Similar to $\mathrm{QA}^1 + \mathrm{MQC}$ that applies MQC on (raw) samples from $\mathrm{QA}^1$, $\mathrm{QA}^6 + \mathrm{MQC}$ performs MQC on resulting samples of $\mathrm{QA}^6$. Figure 3.4 compares the

Table 3.1: Variance of the energy values for 50 repeats.

| Coefficients | Samples | $QA^1$ | $QA^2$ | $QA^3$ | $QA^4$ | $QA^6$ | $QA^1 + MQC$ |
|---|---|---|---|---|---|---|---|
| Binary | 100 | 1.3696 | 2.8304 | 1.4544 | 2.7264 | 5.2416 | 0.9984 |
| | 200 | 0.9936 | 3.4576 | 0.9664 | 1.4656 | 8.1936 | 0.8704 |
| | 500 | 0.7696 | 2.3936 | 0.9104 | 1.3584 | 3.9184 | 0.6400 |
| | 1000 | 0.4816 | 1.9584 | 0.9984 | 0.8464 | 3.3936 | 0.2944 |
| Uniform | 100 | 1.6558 | 1.1323 | 0.6401 | 0.6606 | 1.1116 | 0.0704 |
| | 200 | 0.8674 | 0.9988 | 0.7189 | 0.4792 | 1.2079 | 0.0171 |
| | 500 | 0.5471 | 0.9898 | 0.5596 | 0.4471 | 0.7513 | 0.0092 |
| | 1000 | 0.6006 | 0.5860 | 0.5792 | 0.2757 | 0.7606 | 0.0049 |
| Normal | 100 | 5.9365 | 4.0989 | 2.5668 | 0.4009 | 1.0033 | 0.0563 |
| | 200 | 3.8940 | 3.5666 | 2.4667 | 0.3343 | 0.5284 | 0.0071 |
| | 500 | 2.8559 | 2.0883 | 2.0424 | 0.4017 | 0.4056 | 0.0000 |
| | 1000 | 2.7226 | 2.3799 | 1.5881 | 0.2467 | 0.2899 | 0.0000 |

performance of $QA^1 + MQC$ and $QA^6 + MQC$ in minimizing benchmark (binary, normal and uniform) Ising Hamiltonians.

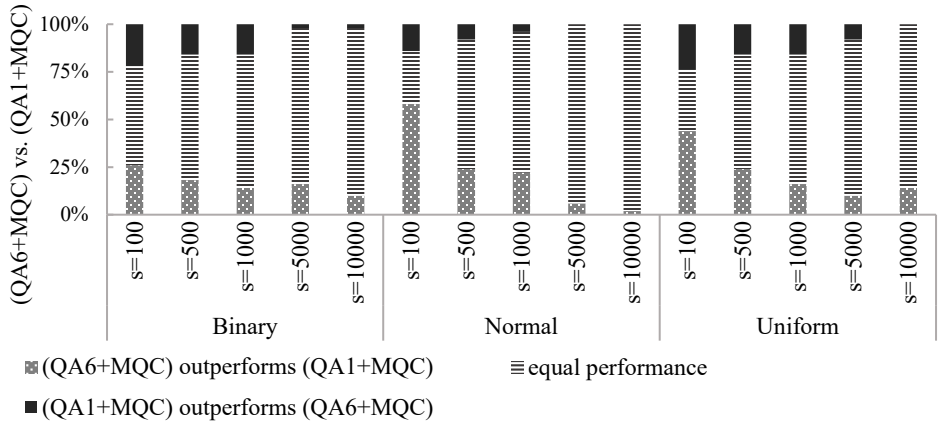


Figure 3.4: Comparing the performance of applying MQC on samples from $QA^1$ and $QA^6$ in solving the benchmark (binary, normal and uniform) random problems.

Figure 3.4 reveals that the performance of MQC depends on the quality of the input sample set. Hence, we hypothesize that we may extend MQC to achieve a better post-quantum error correction approach for quantum annealers. To this end, we extend MQC and introduce a novel method, called randomized multi-qubit correction (RMQC). Algorithm 3.3.1 presents the RMQC algorithm. In RMQC, we repeat the MQC method

$r$ times. We start with the raw sample set (similar to MQC) and then shuffle the sample set in each iteration. Finally, we apply MQC on $r$ samples (results from applying MQC on shuffled sample set) to obtain the final solution.

**Algorithm 3.3.1:** Randomized multi qubit correction (RMQC).

**Input:** $Z = \left[\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^n\right], r > 0, \mathbf{h}$ and $J$
**Output:** $\mathbf{z}^*$

```
S ← {}
do
    r ← r − 1
    S.Append(MQC(Z))
    Z ← Shuffle(Z)
while r > 0
z* ← MQC(S)
return z*
```

Figure 3.5 compares the performance of applying RMQC for $r = 1$ on raw samples (attained by $\mathrm{QA}^1$, denoted by $\mathrm{QA}^1 + \mathrm{RMQC}(1, 1)$, and compare it with results from applying MQC on same sample set, denoted by $\mathrm{QA}^1 + \mathrm{MQC}$. When $r = 1$, RMQC can have a stochastic behavior—i.e., it is possible that MQC can find a better sample. However, for $r > 1$, RMQC is guaranteed to outperform MQC—albeit $r$ times more (classical) computation time.

Figure 3.6 illustrates the performance of RMQC for $r = 5$ and 10—denoted by $\mathrm{QA}^1 + \mathrm{RMQC}(5, 1)$ and $\mathrm{QA}^1 + \mathrm{RMQC}(10, 1)$, respectively—and compare it with the performance of $\mathrm{QA}^1 + \mathrm{MQC}$.

## 3.4 Discussion

Quantum annealers are a type of adiabatic quantum computer that provides a hardware implementation for minimizing Ising Hamiltonians for addressing computationally intensive problems that are intractable in the realm of classical computing. From an

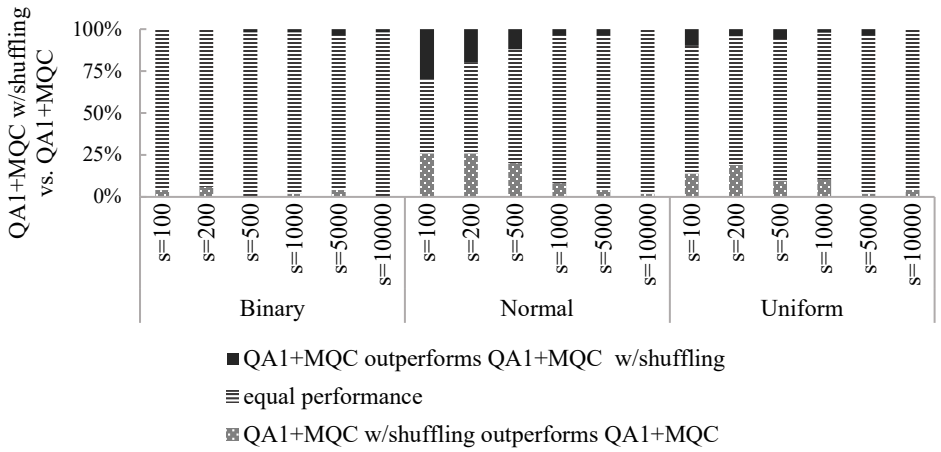


Figure 3.5: Performance comparison between $\mathrm{QA}^1 + \mathrm{MQC}$ and $\mathrm{QA}^1 + \mathrm{RMQC}(1, 1)$.

optimization point of view, quantum annealers can draw many high-quality samples in near-constant time (i.e., drawn samples are very likely to be an excited state). However, they generally fail to find the global minimum, specifically when the energy gap(s) between the ground and excited states are small.

In this chapter, we first studied the impact of applying a local search heuristic, called SQC, on raw samples, drawn by a D-Wave 2000Q quantum annealer. Figure 3.1 reveals that applying SQC on raw samples for a given Ising Hamiltonian with normal and uniform coefficients always results in a sample with lower energy value. In other words, none of the drawn samples for normal and uniform problems were a local optimum, because SQC was able to perform a local search and find another sample with lower energy value. On the other side, roughly all drawn samples for binary problems were a local optimum and applying SQC was not able to improve the quality of the attained samples. There are two possibilities for this observation:

- Ising Hamiltonians with binary (or discrete) coefficients are easier problems and sampling with a D-Wave quantum annealer is very likely to result in the ground or (at least an excited) state;

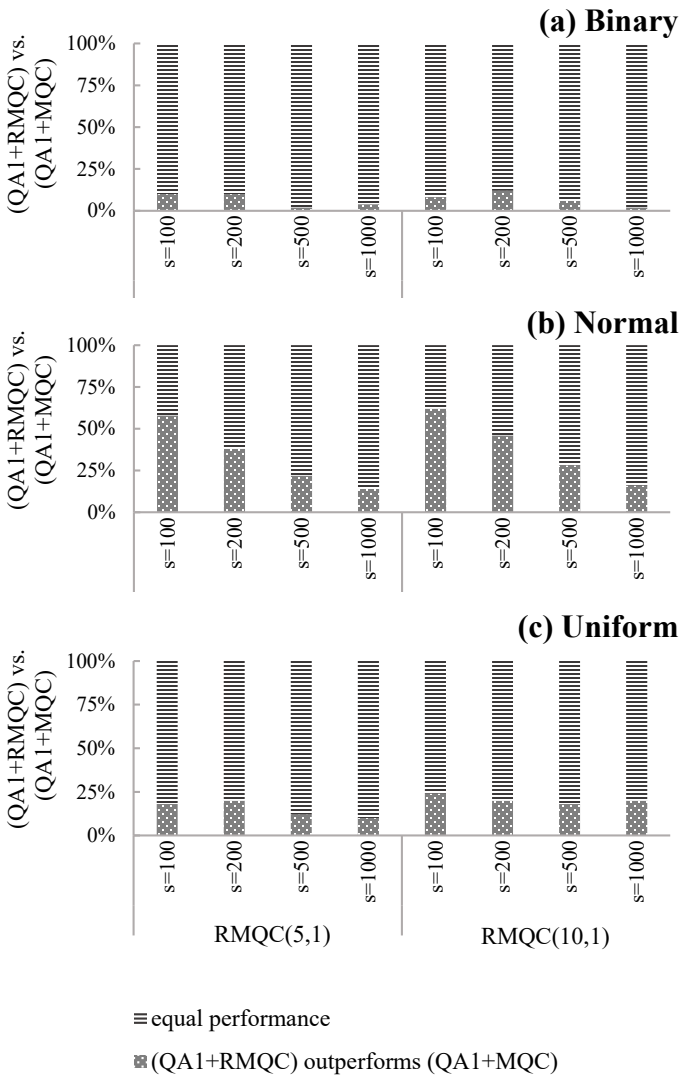


Figure 3.6: Performance comparison between $\mathrm{QA}^1 + \mathrm{MQC}$ and $\mathrm{QA}^1 + \mathrm{RMQC}$ for $r = 5$ and $r = 10$, denoted by $\mathrm{QA}^1 + \mathrm{RMQC}(5, 1)$ and $\mathrm{QA}^1 + \mathrm{RMQC}(10, 1)$, respectively.

- owing to the precision limitations of the D-Wave QPUs, the Ising Hamiltonian that is being minimized by a physical quantum annealer is different from the given Ising Hamiltonian (for example with double precision).

Indeed, not only SQC exploits the neighborhood of an input sample for finding a sample with lower energy, but it can also remediate the precision limitations

Afterward, we studied the impact of a recent postprocessing method, called MQC, and showed that it outperforms software and hardware advances in quantum annealing.

More specifically, we experimentally demonstrated that applying MQC on raw samples, attained by the quantum annealers, can not only find samples with lower energy but also make results of the quantum annealers more reproducible. We extended MQC and introduced a novel method, called randomized MQC (RMQC) that is guaranteed to outperform MQC (state-of-the-art in the realm of quantum annealing), albeit more (classical) computations.

# CHAPTER IV

# SAT++: A Quantum Programming Paradigm

## 4.1 Introduction

Unlike conventional computing machines (i.e., classical and gate model quantum computers) that have a rich set of machine instructions, quantum annealers can execute only one quantum machine instruction (QMI)—minimizing a quadratic objective function. Programming classical and gate model quantum computers define a sequence of operations that emulates the control flow of the intended algorithm. To solve a problem on a quantum annealer, on the other side, one needs to define coefficients of an Ising Hamiltonian, shown in Eq. (2.4), which represents a solution to the original problem of interest in its ground state.

## 4.2 Compiling Classical Algorithms for Quantum Annealers

In this section, we introduce SAT++ as a novel model that bridges the gap between implementing classical algorithms and programming adiabatic quantum computers. From a problem-solving viewpoint, our model receives a classical algorithm (implemented in

classical programming languages) and transforms it into an SAT instance and subsequently into a system of inequalities whose variables are coefficients of Ising Hamiltonians that represent the solution for the original problem of interest in their ground states. We then map these Ising Hamiltonians to adiabatic quantum machine instructions (QMI) that are executable by quantum annealers. Figure 4.1 illustrates the architecture of of SAT++.

## 4.3 Proof-of-Concept

In this section, we aim to demonstrate the validity of the proposed SAT++ model in executing classical algorithms (implemented in classical programming languages) on quantum annealers.

### 4.3.1 Compiling Classical Algorithms to SAT

According to Cook—Levin theorem, we can reduce any problem of class NP to SAT (as an NP-complete problem) in polynomial-time [79]. In SAT++, we use Boolean satisfiability as an intermediate representation of problems. Thus, we need a systematic approach for reducing any classical algorithm to SAT instances.

In this study, we applied Z3 (a high-performance theorem prover by Microsoft Research) for reducing classical algorithms to SAT instances [51]. In this study, we used Python for implementing classical algorithms. It is worth highlighting that Z3 also supports C/C++, Java, OCaml and Microsoft .Net programming languages (like C# and F#). Therefore, SAT++ is not limited to Python and we can extend it to other (Z3 supported) languages. Since SAT is well-nourished in both academia and R&D, there are other (high-performance) tools that one can utilize for SAT-based computations. Ow-

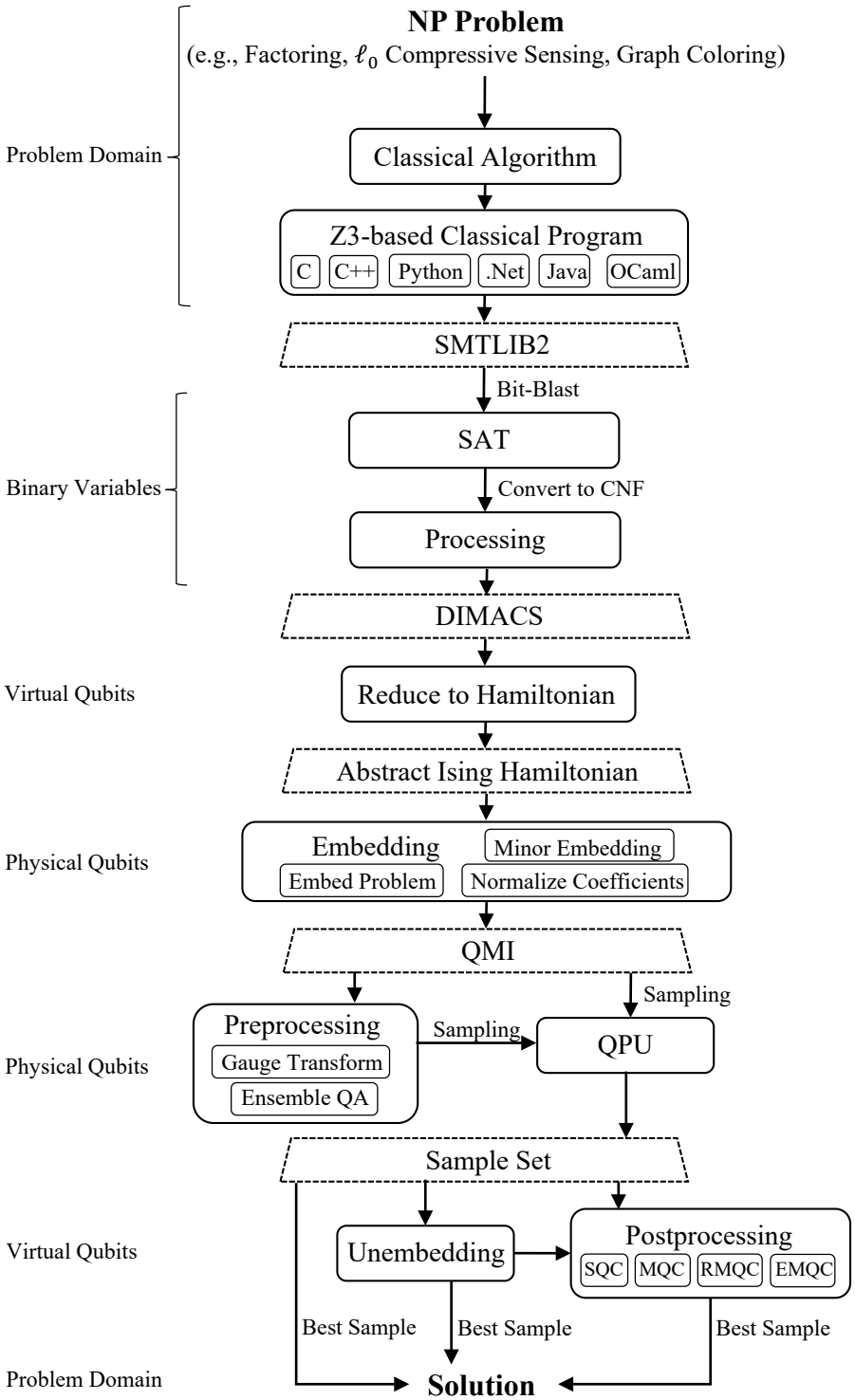


Figure 4.1: Architecture of SAT++ for compiling classical algorithms (implemented in classical programming languages) to quantum machine instructions, executable on quantum annealers.

ing to modular nature of SAT++, it is also possible to replace Z3 with other symbolic computing frameworks.

### 4.3.2 Representing SAT as minimizing Ising Hamiltonians

In this section, we adopt the idea of penalty methods for representing SAT as minimizing Ising Hamiltonians. In this mapping, we aim to find coefficients of Eq. (2.4) such that the ground state of the resulting Ising Hamiltonian represents the satisfying solution of the original SAT instance. In this formulation, $\mathbf{z}_i$ represents the Boolean variable $\mathbf{x}_i$, and we interpret -1 and +1 as "False" and "True", respectively. For a clause with $k$ literals, there are $2^k$ different possibilities among which, we can distinguish the only state that makes the clause to be false—called infeasible state. Hence, we represent each clause of the given SAT instance with two inequalities as:

$$D_i \leq E_{\text{infeasible}} \tag{4.1}$$

and

$$D_i \geq \sum E_{\text{feasible}}, \tag{4.2}$$

where $D_i \in \mathbb{R}$ is the boundary variable corresponding to the clause $C_i$, AND $E_{\text{feasible}}$ and $E_{\text{infeasible}}$ represent the contribution of the feasible and infeasible states in the ultimate energy function, respectively. For $C_i = \mathbf{x}_1 \vee \neg\mathbf{x}_4 \vee \mathbf{x}_9$, as an example, Eq. (4.1) reduces to

$$D_i \leq -\mathbf{h}_1 + \mathbf{h}_4 - \mathbf{h}_9 - J_{1,4} + J_{1,9} - J_{4,9}$$

and we can represent Eq. (4.2) as follows:

$$\begin{aligned} D_i \geq & -\mathbf{h}_1 - \mathbf{h}_4 - \mathbf{h}_9 + J_{1,4} + J_{1,9} + J_{4,9} \\ & -\mathbf{h}_1 - \mathbf{h}_4 + \mathbf{h}_9 + J_{1,4} - J_{1,9} - J_{4,9} \\ & -\mathbf{h}_1 + \mathbf{h}_4 + \mathbf{h}_9 - J_{1,4} - J_{1,9} + J_{4,9} \\ & +\mathbf{h}_1 - \mathbf{h}_4 - \mathbf{h}_9 - J_{1,4} - J_{1,9} + J_{4,9} \\ & +\mathbf{h}_1 - \mathbf{h}_4 + \mathbf{h}_9 - J_{1,4} + J_{1,9} - J_{4,9} \\ & +\mathbf{h}_1 + \mathbf{h}_4 - \mathbf{h}_9 + J_{1,4} - J_{1,9} - J_{4,9} \\ & +\mathbf{h}_1 + \mathbf{h}_4 + \mathbf{h}_9 + J_{1,4} + J_{1,9} + J_{4,9}. \end{aligned}$$

A clause with $k$ literals includes $2^k - 1$ different feasible states so the size of Eq. (4.2) grows exponentially with $k$.

**Theorem 4.3.1.** *Sum of the energy values for all possible states in every Ising model is zero.*

*Proof.* Let $Z$ denotes the set of all possible states in Eq. (2.4). Because spins in the Ising model (here $\mathbf{z}_I$) take their values from $\{-1, +1\}$, $Z$ is a closed set under the complement operation (i.e., $\mathbf{z}, -\mathbf{z} \in Z$), and $|Z| = 2^N$. Accordingly, sum of the energy values for all possible states in the Ising model is:

$$\begin{aligned} \sum_{\mathbf{z}\in Z} E_{\text{Ising}}(\mathbf{z}) &= \sum_{\mathbf{z}\in Z} \left( \sum_{i=1}^{N} \mathbf{h}_i \mathbf{z}_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} J_{ij} \mathbf{z}_i \mathbf{z}_j \right) \\ &= \sum_{i=1}^{N} \left( \frac{2^N}{2} \mathbf{h}_i + \frac{2^N}{2} (-\mathbf{h}_i) \right) + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left( \frac{2^N}{2} J_{ij} + \frac{2^N}{2} (-J_{ij}) \right) \\ &= 0. \end{aligned}$$

□

According to Theorem 4.3.1, we can rewrite Eq. (4.2) as follows:

$$D_i \geq -E_{\text{infeasible}}$$

that obeys

$$0 \leq D_i. \tag{4.3}$$

Note that clauses in CNF representation are connected with "AND" operator. Hence, after representing each clause of the SAT with two inequalities, Eq. (4.1) and (4.3), we aggregate the resulting sub-systems of inequalities to form a larger system of inequalities. After representing the given SAT instance with $M$ clauses with a system of $2M$ inequalities, we augment the resulting system of inequalities via adding inequalities (2.6) and (2.7) to address coefficient range limitations of the D-Wave quantum annealers—i.e., $\mathbf{h}_i \in [-2, +2]$ and $J_{i,j} \in [-1, +1]$, respectively.

To obtain coefficients of the Ising Hamiltonian, shown in Eq. (2.4), whose ground state can represent a satisfying solution of the given SAT instance, we solve the following objective function:

$$\max_{\mathbf{h},J,D} \sum_{i=1}^{M} D_i. \tag{4.4}$$

Note that inequalities(4.1), (4.3), (2.6) and (2.7) are linear. Thus, the objective function in Eq. (4.4) is tractable by linear programming and convex optimization techniques. Considering that biases and couplers on a D-Wave QPU are bounded, Eq. (2.6) and (2.7), convergence of problem (4.4) is guaranteed.

### 4.3.3 Compiling Ising Hamiltonians to Executable QMI

Although we showed how to address the coefficient range limitations of the D-Wave quantum annealers, solving problem (4.4) results in an Ising Hamiltonian which is not necessarily compatible with the D-Wave hardware graph (Chimera topology for the current generation). Therefore, we need to embed the resulting (abstract) Ising Hamiltonian to the working graph of the target quantum annealer. In this study, we apply the minor-embedding heuristic [35] for embedding the problem to the physical lattice of qubits on a D-Wave QPU.

## 4.4 Prime Factoring with SAT++: A Study Case

In number theory, the problem of prime factoring aims to find the canonical representation of a given integer number $n \in \mathbb{N}, n > 1$, as follows:

$$n = \prod_{i}^{k} n_i^{r_i} = n_1^1 \times n_2^2 \times \cdots \times n_k^k,$$

where $p_1 < p_2 < \cdots < p_k$ are prime numbers and $r_i$ are positive (nonzero) integer numbers. According to the *fundamental theorem of arithmetic*, canonical representation of any positive integer number is unique. Although there are debates on the class (or complexity) of this problem, there is no known efficient (nonquantum) algorithm for factoring numbers in polynomial-time [109, 188, 18].

Besides the length of the given composite numbers, on a binary basis, the difficulty of integer factoring also depends on the properties of integer numbers. The hardest instances of this problem are factoring pseudo-prime numbers (i.e., the product of two same size prime numbers). Security of the modern public-key cryptography systems (namely RSA)

mainly relies on the difficulty of factoring very large pseudo-prime numbers [62, 146]. Moreover, the best-known algorithm for Fast Fourier transform (FFT) depends on the integer factoring problem [163].

Despite the problem of prime factoring has not been proven to be in class NP, attempts to address it efficiently in the realm of classical computing (even with employing large-scale supercomputers) have failed. Heuristic search approaches (namely the simulated annealing and Tabu search) and meta-heuristics (e.g., evolutionary algorithms, swarm intelligence techniques and hybrid approaches like memetic algorithms) have demonstrated outstanding functionalities in addressing complex real-world applications [68]. Studies on adopting intelligent optimization techniques for prime factoring, nevertheless, have revealed that such algorithms generally converge to local optimums when we want to factor pseudo-prime numbers [186, 50, 44]. In other words, intelligent optimization techniques approximate the global optimum solutions quickly; however, prime factorization requires exact global optimum [121].

Since modern SAT solvers can handle instances with thousands of clauses and millions of variables (in CNF representation), several studies attempted to reduce the problem of prime factoring to SAT and employ HPC based SAT solvers to factor pseudo-prime numbers. Experimental results, however, showed that SAT-based prime factoring cannot address the problem of prime factoring efficiently [163, 110]. In 1994, Peter Shor introduced a polynomial-time quantum algorithm for integer factoring [169, 170]. Shor's algorithm, nevertheless, requires a quantum computer with a reasonably large number of qubits that is beyond the capacity of current (gate model) quantum processors.

### 4.4.1 Results

In this experiment, we used Z3 [51] for implementing the proposed model in Python and we applied the Tseitin transformation [104] to represent the resulting problems in CNF. Note that solving problem (4.4) will result in an Ising Hamiltonian which is not necessarily compatible with the D-Wave hardware graph (Chimera topology for the current generation). Therefore, we applied the minor-embedding heuristic [35] for embedding the problem onto the physical lattice of qubits on a D-Wave QPU. As the case of study, we employed SAT++ model to solve the problem of prime factorization [146] on a D-Wave 2000Q quantum processor. To factor a pseudo-prime number in the realm of classical computing we can:

- look for two integer numbers $p$ and $q$ whose multiplication is equal to the input number;
- find an integer $p$ where the remainder of the input number to $p$ is zero.

Figure 4.2 displays the code sketch of multiplication based factoring of pseudo-prime numbers with SAT++ in Python.

While the majority of quantum annealing based integer factoring studies have adopted the multiplication based factoring approach, Fig. 4.3 illustrates the code sketch of the remainder based factoring of pseudo-prime numbers with SAT++ in Python. Our experimental results on a D-Wave 2000Q quantum processor revealed that, although remainder -based factoring forms a smaller classical program, multiplication-based factoring results in a smaller SAT instances. Multiplication-based factoring, therefore, can be a better choice, specifically due to the capacity of the current QPUs.

```
1  import numpy as np
2  from z3 import *
3  from SATpp import *
4
5  a=11*13 #number to factor
6  n=np.int(np.ceil(np.log2(a)))
7  n1=np.int(np.floor(k/2))
8  n2=n-n1
9
10 y=BitVecVal(a, n)
11 p=BitVec('p', n1)
12 q=BitVec('q', n2)
13
14 g=Goal()
15 g=bitmapping(g, p)
16 g=bitmapping(g, q)
17 r=bv_multiply(p,q)
18 g.add(ZeroExt(r.size(), y)==ZeroExt(y.size(), r))
19 g.add(UGT(p,1))
20 g.add(UGT(q,1))
21
22 cnf=ToCNF(g, ToPropositional=True)
23 result=Ising_SAT_Solver(cnf)
```

Figure 4.2: Code sketch for multiplication based factoring with SAT++.

Table 4.1: Results for multiplication-based factoring with SAT++on a D-Wave 2000Q.

| Method | Result |
|---|---|
| SAT++ with QA | $35 = 5 \times 7$ |
| SAT++ with SQC | $143 = 11 \times 13$ |
| SAT++ with MQC (on virtual qubits) | $143 = 11 \times 13$ |
| SAT++ with MQC (on Boolean variables) | $12,317 = 109 \times 113$ |

We first employed the standard quantum annealing (i.e., with out using any post-quantum error correction technique) to satisfy the resulting SAT instances by SAT++. Using a D-Wave 2000Q, we were able to only factor 35 to its prime factors, 5 and 7, which is trivial. Applying SQC, shown in algorithm 3.1.1, enabled SAT++ to factor 143 to its prime factors, 11 and 13. In the same manner, we applied MQC, shown in algorithm 3.1.3 to extend the results from employing SQC; however, SAT++ with MQC could not demonstrate a better performance, compared to SAT++ with SQC. Applying MQC on

```
1  import numpy as np
2  from z3 import *
3  from SATpp import *
4
5  a=11*13 #number to factor
6  n=np.int(np.floor(np.log2(a)))+1
7  n1=np.int(np.ceil(n/2))
8
9  y=BitVecVal(a, n)
10 p=BitVec('p', n)
11
12 g=Goal()
13 g=bitmapping(g, p)
14 g.add(URem(y,p)==0)
15 g.add(UGT(p,1))
16 g.add(ULT(p,2**n1))
17
18 cnf=ToCNF(g, ToPropositional=True)
19 result=Ising_SAT_Solver(cnf)
```

Figure 4.3: Code sketch for remainder based factoring with SAT++.

Boolean variables (i.e., problem variable domain) enabled SAT++ to factor 12,317 to its prime factors, 109 and 113. Table 4.1 summarizes the experiment results.

## 4.5 Discussion

Classical computers perform a sequence of operations on memory registers for solving encoded problems; however, quantum annealers are single-instruction computing machines that can only sample from the ground state(s) of the given Ising Hamiltonians. In this chapter, we introduced SAT++ as a novel quantum programming paradigm that enables us to execute classical algorithms (implemented in classical programming languages) on quantum annealers. Our model first reduces the given classical algorithm to SAT—intermediate representation of problems in SAT++—and then maps the CNF representation of the SAT to an Ising Hamiltonian whose ground state can represent a solution of the original problem of interest.

As a proof-of-concept, we showed how one can reduce decision problems, implemented in Python, to SAT. We also introduced a novel heuristic for mapping SAT instances to Ising Hamiltonians. It is important to highlight that our objective in this study was not to employ quantum annealers for addressing the NP-complete problem of SAT. Moreover, the proposed heuristic is not guaranteed to solve all SAT instances, even if we have access to an ideal quantum annealer.

As a case of study, we employed our model to address the problem of prime factorization on a D-Wave 2000Q quantum processor and we successfully factored 12,317 to its prime factors (109 and 113). Representing SAT instances in CNF, in the worst-case scenario, can be exponentially time-consuming. Thus, we used the Tseitin transformation [104] to represent the resulting problems in CNF which runs in polynomial-time—albeit defining ancillary variables. Although applying the Tseitin transformation increases the size of the SAT linearly, resulting in problems that can quickly exceed the capacity of current quantum annealers. In the same manner to compilers that apply various code optimization techniques for enhancing the performance of the machine code on a specific CPU, employing preprocessing and postprocessing techniques can reduce the size of SAT instances (intermediate representation of problems in SAT++) which will increase the size of problems that SAT++ can address. It is worth noting that our research objective in this study was not to set a new record for quantum factorized integers, which for the current generation of the D-Wave quantum annealers is 1,005,973 [146].

From a classical programming viewpoint, remainder based factoring looks simpler than a multiplication based approach. However, our experimental results demonstrated that remainder based SAT instances require more ancillary variables in CNF representation, compared to the multiplication based factoring approach. Hence, a multiplication

based factoring approach in SAT++ was able to factor larger pseudo-prime numbers. From a complexity perspective, instead of analyzing the difficulty of problems, we may define a metric for analyzing the difficulty of instances of a problem.

In the first generation of SAT++, we only considered SAT-based representation of classical problems. Therefore, our study cases were limited to decision problems. However, SAT++ is not limited to decision problems and one can use extensions of SAT to address further problem types. For example, we can reduce optimization problems to MAX-SAT and Weighted-MAX-SAT instances. Similarly, we can use #SAT as an intermediate representation of model counting problems.

# CHAPTER V

# AI Hybrid Quantum Annealing

## 5.1 Introduction

Quantum artificial intelligence (QAI) and quantum machine learning (QML) are emerging fields that leverage quantum computing to address certain types of (hard AI) problems that are intractable in the realm of classical computing [102, 20, 64]. Unlike most current research studies in QAI and QML that apply quantum computing models to hard AI problems, in this section, we explore how we might apply AI techniques to improve quantum information processing.

## 5.2 Reinforcement Quantum Annealing

Assume that the given problem $\Pi$ that we aim to solve on a quantum annealer contains a finite set of constraints (components), denoted by $\pi_i$ for $i \in \{1, 2, \ldots, M\}$, over the same variables as follows:

$$\Pi := \{\pi_i, \pi_2, \ldots, \pi_M\}, \tag{5.1}$$

where $M$ indicates the number of constraints and our ultimate objective is to find a solution that addresses (satisfies) all constraints. Let $H_i$ be an Ising Hamiltonian whose ground state represents a solution for $\pi_i$ and $H_\Pi$ be the corresponding Ising Hamiltonian of $\Pi$ that all are acting on the same spins (variables). In addition, let $E_0^{H_\Pi}$ be the ground state energy of $H_\Pi$ and $E_0^{H_i}$ be ground state energy of corresponding $H_i$. If there exists $\mathbf{z}$ that puts $H_i$ ($\forall i, i \in \{1, 2, \ldots, M\}$) in their ground states (i.e., satisfies all constraints in $\Pi$), then $\mathbf{z}$ also puts $H_\Pi$ in its ground state [125]. In other words,

$$E_0^{H_\Pi} = E_0^{H_1} + E_0^{H_2} + \cdots + E_0^{H_M}. \tag{5.2}$$

Hence, we can represent $H_\Pi$ as follows:

$$H_\Pi = H_1 + H_2 + \cdots + H_M. \tag{5.3}$$

This setting appears in a vast range of problem formulations—including, but not limited to SAT [23], constraint satisfaction problems [182, 183], planning and scheduling [155, 179, 177], fault detection and diagnosis [148, 22], and compressive sensing [17, 127]—specifically when we adopt the idea of penalty methods to reduce problems of interest to minimizing Ising Hamiltonians.

**Theorem 5.2.1.** *For any problem in class NP, there are infinite different Ising Hamiltonians whose ground states are all identical to the solution of the original problem.*

*Proof.* According to Cook—Levin theorem, we can reduce any NP problem to minimizing an Ising Hamiltonian (which is also in the class NP) in polynomial-time [79]. Multiplying all coefficients of the Ising model by a positive non-zero real number results in a new Ising

model whose ground state will be identical to the original Ising model. Since the number of positive real numbers are infinite, we can generate infinite different Ising models whose ground states represent the solution for the original problem of interest. □

According to Theorem 5.2.1, there are infinite different Ising Hamiltonians whose ground states all represent the solution of the original problem of interest—nevertheless owing to the range and precision limitations on the D-Wave QPUs, we have finite different Ising models for a problem. In theory, these different Ising models are equivalent to each other—i.e., an adiabatic annealing process always attains the ground state which is identical for all corresponding Ising models of a problem. In practice, however, each of these (theoretically) equivalent Ising models are analogous to a pseudo-Boltzman distribution whose parameters are different. Consequently, when we minimize the corresponding QMIs with a physical quantum annealer (like the D-Wave QPU), the probability of finding the global minimum for different Ising Hamiltonians of a given problem varies from zero to one. As an example, an annealing process on a D-Wave QPU may become diabatic because the required anneal time exceeds the maximum possible anneal time (2,000 micro-seconds), which can substantially reduce the probability of finding the global minimum. Note that for a given Ising Hamiltonian, we cannot estimate the probability of finding the ground state prior to executing the corresponding QMI. We present the Reinforcement Quantum Annealing (RQA) scheme that leverages the idea of learning automata to iteratively find better Ising Hamiltonians—i.e., probability of finding the global minimum through executing the corresponding QMIs on a quantum annealer is increased over the time.

### 5.2.1 RQA: A Learning Automata Approach

Learning automata (LA) [130] are adaptive decision-making models (i.e., type of reinforcement learning [96]) that try to maximize the accumulative reward when they are interacting with stochastic environments. In a similar manner to reinforcement learning, LA use Markov-decision-processes for representing the automaton-environment structure [96, 130, 175]. In a learning automaton, the agent has a set of $r$ actions, denoted by

$$\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_r\},$$

and each action has a corresponding probability to be taken by the agent, denoted by $\mathbf{p}_i$ such that

$$\sum_{i}^{r} \mathbf{p}_i = 1.$$

In each episode, the agent takes (applies) the action $\alpha_i$ (according to $\mathbf{p}$), and (correspondingly) the stochastic environment returns its feedback $\beta$ that specifies the performance evaluation of the action $\alpha_i$. The agent uses this feedback to learn from the environment and aims to take optimal actions over time. For $\beta \in [0, 1]$ —so-called S-Type learning automata—in episode $t$, if the agent takes the action $\alpha_i$ and receives the feedback $\beta^t$, we can update $\mathbf{p}$ as follows:

$$\mathbf{p}_j^{t+1} = \begin{cases} \mathbf{p}_j^t - \theta_2 \left(1 - \beta^t\right) \mathbf{p}_j^t + \theta_1 \beta^t \left(1 - \mathbf{p}_j\right), & i = j; \\ \mathbf{p}_j^t + \theta_2 \left(1 - \beta^t\right) \left(\frac{1}{r-1} - \mathbf{p}_i^t\right) - \theta_1 \beta^t \mathbf{p}_j, & i \neq j, \end{cases} \tag{5.4}$$

where $\beta = 0$ represents the lowest action performance and $\beta = 1$ represents the highest action performance, respectively, $\theta_1, \theta_2 \in [0, 1]$ are learning factors, and $i, j \in \{1, 2, \ldots, r\}$ [130].

We introduce a novel scheme, called reinforcement quantum annealing (RQA), in which an intelligent agent interacts with a quantum annealer as a stochastic environment and tries to iteratively find better Ising Hamiltonians that sampling from their ground state(s), by quantum annealers, results in a better distribution—i.e., the probability of finding the global optimum for the original problem of interest is increased over time. To this end, we extend Eq. (5.3) as follows:

$$\widetilde{H}_\Pi = \widetilde{H}_1 + \widetilde{H}_2 + \cdots + \widetilde{H}_M, \tag{5.5}$$

such that:

$$\widetilde{H}_i = \chi\left(H_i, \boldsymbol{\rho}_i\right), \tag{5.6}$$

where $\boldsymbol{\rho}_i \in \mathbb{R}$ denotes the impact (or influence) factor of $\pi_i$ (or $H_i$) and $\chi$ is a function that maps the input Hamiltonian to a different Hamiltonian which satisfies:

- any $\mathbf{z}$ that puts $H_i$ in its ground state also puts $\widetilde{H}_i$ in its ground state, and vice versa;
- if $\boldsymbol{\rho}_i^1 < \boldsymbol{\rho}_i^2$ then $\chi\left(H_i, \boldsymbol{\rho}_i^1\right) \geq \chi\left(H_i, \boldsymbol{\rho}_i^2\right)$.

Now we extend learning automata to allow the agent to take multiple actions in each episode. Let $\hat{\alpha}^t \subset \alpha$ be the set (list) of actions that the agents takes in episode $t$.

We can extend the Eq. (5.4) as follows:

$$\mathbf{p}_j^{t+1} = \begin{cases} \mathbf{p}_j^t - \theta_2 \left(1-\beta^t\right) \mathbf{p}_j^t + \theta_1 \beta^t \left(1-\mathbf{p}_j\right), & \alpha_j \in \hat{\alpha}^t; \\ \mathbf{p}_j^t - \theta_2 \left(1-\beta^t\right) \left(\frac{1}{r-\hat{r}} - \hat{p}_{\hat{\alpha}}^t\right) - \frac{\theta_1 \beta^t}{r-\hat{r}} \left(\hat{r} - \hat{p}_{\hat{\alpha}}^t\right), & \alpha_j \notin \hat{\alpha}^t, \end{cases} \tag{5.7}$$

where $\hat{r} = \left|\hat{\alpha}^t\right|$ and,

$$\hat{p}_{\hat{\alpha}}^t = \sum \mathbf{p}_i; \qquad \text{for } \alpha_i \in \hat{\alpha}^t.$$

Finally, we leverage multi-task learning automata—let $\boldsymbol{\rho}_i = \mathbf{p}_i$ and $M = r$—to propose the RQA scheme. RQA is an iterative process that we can start it with a uniform distribution of influence factors as follows:

$$\boldsymbol{\rho} = \{\frac{1}{M}\}^M.$$

In each iteration, the agent applies Eq. (5.5) and submits the corresponding QMI to a quantum annealer. After performing the necessary post-processing methods (like remediating broken-chains and applying post-quantum error correction heuristics), the agent estimates $\beta$ according to the number of satisfied constraints ($\pi_i$) and employs Eq. (5.7) to update the influence factor $\boldsymbol{\rho}$.

### 5.2.2 Proof of Concept: RQA for Solving SAT Instances

In this section, as a proof-of-concept, we show that adopting the proposed RQA scheme can notably improve the performance of quantum annealers when we apply them for solving SAT and MAX-SAT instances. To adopt the proposed RQA scheme, we rewrite the inequality (4.3) as follows:

$$\boldsymbol{\rho}_i \leq D_i, \tag{5.8}$$

where $\boldsymbol{\rho}_i$ denotes the influence factor of the clause $C_i$. Here, we define $\boldsymbol{\rho}$ as:

$$\boldsymbol{\rho}_i = \frac{1}{M} - \mathbf{p}_i, \tag{5.9}$$

where $\mathbf{p}_i$ is the corresponding probability of constraint $\pi_i$ (here the clause $C_i$) in Eq. (5.7). Note that when $\mathbf{p}_i = \frac{1}{M}$, the inequalities (4.3) and (5.8) are identical. The architecture of the proposed agent contains the following components:

- $\Phi^t$—set of unsatisfied clauses in episode $t$;
- $\boldsymbol{\rho}^t$—tuple of $M$ influence factors in episode $t$;
- $\text{QMI}^t$—action of the agent in episode $t$, the Ising Hamiltonian for solving the given SAT instance (according to $\Phi^t$ and $\boldsymbol{\rho}^t$);
- $\mathbf{z}$—perception of the agent from the stochastic environment, resulting sample(s) from executing the $\text{QMI}^t$ on a quantum annealer.

For a given Boolean formula in CNF, the agent initializes its internal state as:

$$\Phi^0 = \emptyset$$

and

$$\mathbf{p}^0 = \{\frac{1}{M}\}^M.$$

In each episode, the agent forms a system of inequalities with Eq. (4.1) and (5.8), and embeds Eq. (2.6) and (2.7). Afterward, the agent solves problem (4.4), and submits the resulting Ising Hamiltonian ($\text{QMI}^t$) to a D-Wave QPU. The environment (here the D-Wave QPU) draws sample(s) from the corresponding pseudo-Boltzmann distribution,

and returns the resulting sample(s). The episode ends with updating the internal state of the agent as follows:

$$\Phi^t = \text{set of unsatisfied clauses with } \mathbf{z}^t,$$

$$\beta^t = 1 - \frac{\left|\Phi^t\right|}{M},$$

and (finally) updating probabilities with (5.7). In RQA, the action of the agent in episode $t$ depends on $\boldsymbol{\rho}_{t-1}$; therefore, Markov property holds here [47].

### 5.2.3 Results

In this section, we aim to evaluate the performance of RQA scheme on solving benchmark SAT instances, and compare it with recent software and hardware enhancements to the quantum annealers. For every SAT instance, we used the number of unsatisfied clauses as the metric for performance comparisons. In this study, we used Z3 (from Microsoft Research) as a framework for symbolic computing implementations [51] and we executed each QMI on the D-Wave 2000Q quantum annealer, located at Burnaby, British Columbia. We implemented all experiments in Python 3.7.4, and executed them on a 64-bit Windows 10 based system with 32 GB RAM and Intel Xeon processor at 3.00 GHz.

For every SAT instance, we used inequalities (4.1), (5.8), (2.6) and (2.7) to represent the given SAT instance as a system of inequalities. Afterward, we solved problem (4.4) for reducing the SAT to an executable QMI on a D-Wave quantum processor. Note that solving problem (4.4) will result in an Ising Hamiltonian which is not necessarily compatible with the D-Wave hardware graph (Chimera topology for the current generation). Therefore, we applied the minor-embedding heuristic [35] for embedding the problem to the physical lattice of qubits on a D-Wave QPU. To avoid the impact of chaining physical

qubits in our evaluations, we employed fixed embeddings of cliques in all instances—i.e., we used the pre-defined embeddings of cliques for the chimera architecture.

Recent studies have revealed that using spin-reversal transforms (a.k.a. gauge transforms)—i.e., flipping the qubits randomly without altering the ground state of the original Ising Hamiltonian—can reduce analog errors of the quantum annealers [145]. Thus, as a preprocessing technique, we applied spin-reversal transforms prior to submitting the QMIs to the physical QPU. We also put a delay between measurements to reduce the sample-to-sample correlation, albeit longer run-time.

To remediate possible broken chains in the resulting raw samples from the D-Wave QPU, we performed voting among the physical qubits of chains. After unembedding samples (i.e., representing variables in the original problem domain), we applied the multi-qubit correction (MQC) heuristic [60] which has demonstrated a significant ability to improve the probability of finding the global minimum, attained by the D-Wave QPU. Finally, we performed a local search heuristic, so-called single-qubit correction (SQC), to construct the final solution of the given SAT [60].

#### 5.2.3.1 Experiment A: Factoring Pseudo-Prime Numbers

In this study, we use the problem of prime factorization as a benchmark to evaluate the performance of RQA. It is worth noting that our research objective in this study was not to set a new record for quantum factorized integers, which for the current generation of the D-Wave quantum annealers is 1,005,973 [146]. Indeed, since the security of the modern public-key cryptography systems (like RSA) mainly relies on the difficulty of factoring very large pseudo-prime numbers [62, 146], we relied on the difficulty of prime factorization problem for generating benchmark SAT instances. Let $f(\mathbf{x}_1, \mathbf{x}_2)$be a Boolean

function as follows:

$$\mathbf{q} = f(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1 \times \mathbf{x}_2, \tag{5.10}$$

where $\mathbf{x}_1 \in \{0,1\}^{n_1}$ and $\mathbf{x}_2 \in \{0,1\}^{n_2}$ are integer-valued numbers in binary representation (here, $\mathbf{x}_1, \mathbf{x}_2 \geq 2$), and the multiply operator is in binary base—each element of the vector $\mathbf{q}$ is a Boolean function of $\mathbf{x}_1$ and $\mathbf{x}_2$.

Assume that $\hat{\mathbf{q}}$ is a pseudo-prime integer number in binary base (i.e., $\hat{\mathbf{q}}$ has two prime factors. We can map the problem of factoring $\hat{\mathbf{q}}$ to SAT as follows

$$g = f_{\text{SAT}}(\mathbf{q}, \hat{\mathbf{q}}) = \bigwedge_{i=1}^{n} \neg(\mathbf{q}_i \oplus \hat{\mathbf{q}}_i), \tag{5.11}$$

where $n = n_1 + n_2$ denotes the length of $\mathbf{q}$. We can look at the process of generating SAT instances from a reverse-engineering viewpoint. To this end, we generated pseudo-prime numbers via multiplying two prime numbers, and represented them in binary base (denoted by $\hat{\mathbf{q}}$). For each instance, we then used the Eq. (5.11) to map the factorization of $\hat{\mathbf{q}}$ to a satisfiable Boolean formula. Since $g$ is a Boolean expression of $x$, we applied the Tseitin transformation [104] to represent $g$ in conjunctive normal form (CNF) [27]. We also performed pre-processing techniques, namely "ctx-solver-simplify", "recover-01", "propagate-values" and "reduce-args" tactics from Z3 [52]. Note that applying the Tseitin transformation can increase the size of $g$ linearly, due to defining auxiliary variables. Since the capacity of the current D-Wave 2000Q quantum processors is limited to a complete graph of size 63, we eliminated SAT instances (in CNF) with more than 63 Boolean variables which resulted in 136 satisfiable SAT instances.

Figure 5.1 illustrates results—minimum (circles), maximum (triangles), average and variance of the number of unsatisfied clauses—for solving these 136 satisfiable SAT in-

stances, and compares the performance of the proposed RQA scheme with quantum annealing (QA) and quantum annealing with multiple post-quantum processes (SMQC). To enhance the standard quantum annealing technique, we used two spin-reversal-transforms [145], as well as the delay between measurements to reduce the inter-sample correlation. In the second method (SMQC), we first used the multi-qubit correction (MQC) method [60], in problem variable level—which is the state-of-the-art technique in the realm of post-quantum correction for quantum annealers—and then applied a local search to maximize the quality of results, attained by the SMQC arrangement.

To update the influence factors of clauses in RQA, Eq. (5.7), we used $\theta_1 = 0.1$ and $\theta_2 = 0$. Learning automata generally require a notable number of episodes to converge to an optimal (or sub-optimal) policies. In this experiment, nevertheless, the agent terminates the process after at most $T = 10$ episodes (due to QPU time limitations) or finding a solution that satisfies all clauses. Hence, we formed a hall-of-fame—a set of final solutions from all episodes—and applied MQC (followed by SQC) on them to obtain the ultimate solution of RQA. Our empirical observations showed that this technique can implicitly address the limited number of allowed episodes in RQA. Note that RQA utilizes at most the same number of samples as QA and SMQC.

#### 5.2.3.2 Experiment B: Uniform Random 3-SAT with Phase Transitions

Sampling from the phase transition region of uniform Random 3-SAT is a common practice for generating benchmark SAT (and MAX-SAT) problems [42, 167, 1, 139]. In this experiment, as our second study case, we used the satisfiable benchmark test-set of uniform random 3-SAT with phase transitions [89]. Considering the capacity of the

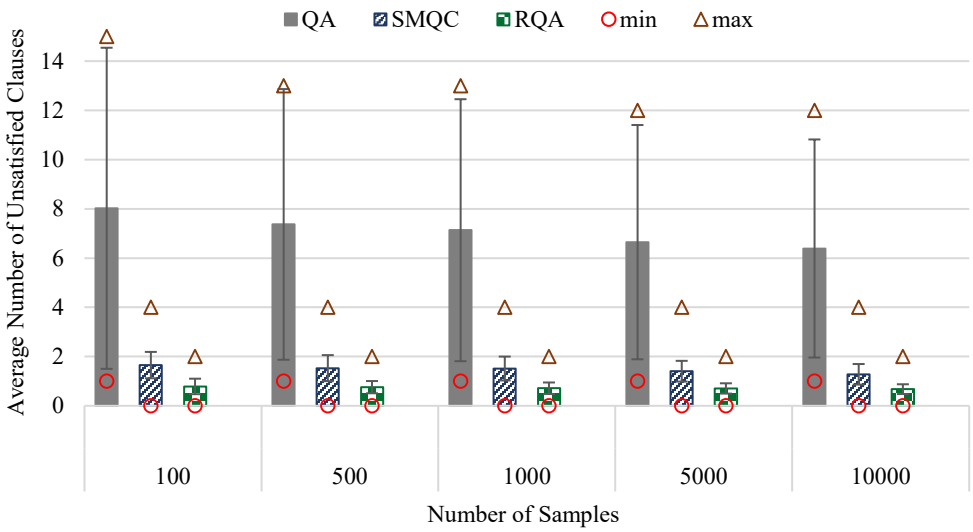


Figure 5.1: Experiment results for solving 136 satisfiable SAT instances (with at most 63 Boolean variables) for factoring pseudo-prime numbers with quantum annealing (QA), quantum annealing with classical post-processing (SMQC) and reinforcement quantum annealing (RQA).

current generation of the D-Wave quantum annealers—we can embed a clique of size at most 63 on chimera architecture—we employed the test-set with 50 Boolean variables.

Figure 5.2 demonstrates results—minimum (circles), maximum (triangles), average and variance of the number of unsatisfied clauses—for solving the first 100 instances from the benchmark test-set, and (similar to the previous experiment) compares the performance of the proposed RQA scheme with quantum annealing (QA) and quantum annealing with multiple post-quantum processes (SMQC). The setting for this experiment was identical to the previous experiment, except the number of instances (136 vs. 100) and the number of variables (variant vs. 50).

## 5.3 Greedy Quantum Annealing

Greedy algorithms are a problem-solving paradigm where we make locally optimal choices at each stage and expect that they yield a globally optimum solution. Although most greedy algorithms fail to achieve the global optimum, in many applications they are

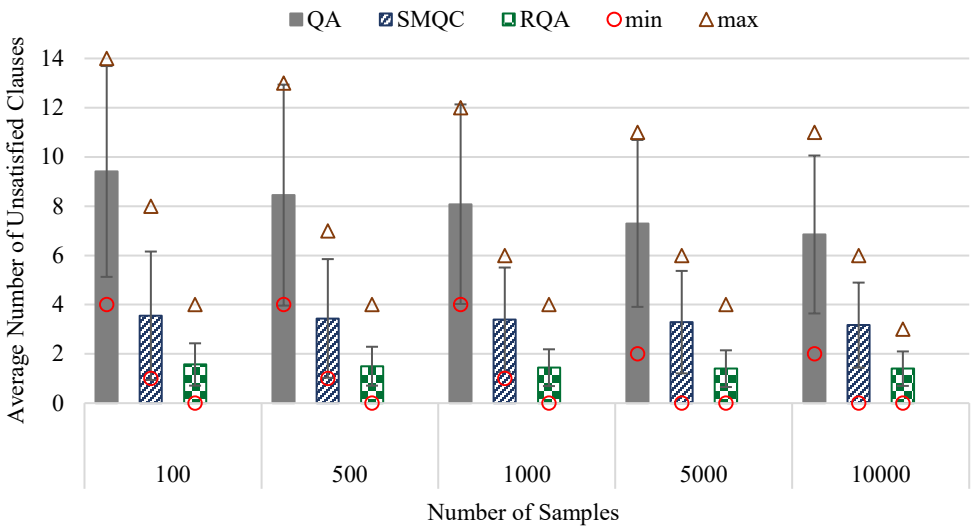


Figure 5.2: Experiment results for solving 100 satisfiable uniform random 3-SAT instances with phase transitions—each SAT instance contains 50 Boolean variable—using quantum annealing (QA), quantum annealing with classical post-processing (SMQC) and reinforcement quantum annealing (RQA).

the best choice due to their efficiency [55]. As an example, greedy algorithms are widely used in sparse recovery applications, at the cost of lower recovery accuracy (compared to convex optimization methods in compressive sensing) [127]. Note that greedy algorithms can find globally optimum solutions if the problem exhibits optimal substructure. In this section, we propose a novel hybrid approach, called greedy quantum annealing(GQA), that leverages quantum annealers to better select candidates in each stage of the greedy algorithms.

### 5.3.1 Method

Let $H$ be an Ising Hamiltonian, representing a solution to the problem of interest in its ground state, as follows:

$$H := \sum_{i=1}^{N} \mathbf{h}_i \mathbf{z}_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} J_{ij} \mathbf{z}_i \mathbf{z}_j, \tag{5.12}$$

, where $\mathbf{h}$ and $J$ represent linear and quadratic coefficients, respectively. Also, let $\mathbf{z}^*$ denotes the ground state of $H$ as

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} H. \tag{5.13}$$

In this study, we aim to find $\tilde{\mathbf{z}}$ such that:

$$|H(\mathbf{z}^*) – H(\tilde{\mathbf{z}})| \to 0. \tag{5.14}$$

In other words, our objective is to find a sample whose corresponding Ising energy value, shown in Eq. (2.4), approaches the energy value of the ground state of the given Ising Hamiltonian. The proposed greedy quantum annealing (GQA) starts with

$$\tilde{\mathbf{z}} = \{\}$$

and

$$H^{t=0} = H,$$

where $t$ denotes the episode (or iteration) of the GQA algorithm. At each iteration, GQA uses a quantum annealer to draw $n$ samples from the ground state of $H^t$. To solve problem (5.13) on a D-Wave quantum processor, as an example, one needs to embed $H^t$ to the working graph of the D-Wave quantum processors (i.e., the Chimera topology for the current generation). We also need to normalize $\mathbf{h}$ and $J$ to be in $[-2, +2]$ and $[-1, +1]$, respectively [16].

Let $Z$ denotes the set of all samples, drawn by a quantum annealer from the ground state of $H^t$, as follows:

$$Z = \{\mathbf{z}^1, \mathbf{z}^2, \cdots, \mathbf{z}^n\},$$

where $\mathbf{z}^j \in \{-1, +1\}^N$. All samples (i.e., $\mathbf{z}^j$; for $j = 1, 2, \ldots, n$) contain a measurement for every qubit. Hence, we can look at each problem variable ($\mathbf{z}_i$, for $i = 1, 2, \cdots, N$) as a random variable with Bernoulli distribution that takes its value from $\{-1, +1\}$. Note that in this representation, the value -1 in (2.4) is analogous to 0 in (2.5); therefore, we can extend all analysis to QUBO form and employ GQA in binary settings.

After retrieving the sample set $Z$, we estimate the uncertainty of every problem variable ($\mathbf{z}_i$) as follows:

$$u(\mathbf{z}_i) = 1 - \frac{|\sum_{j=1}^{n} \mathbf{z}_i^j|}{n}. \tag{5.15}$$

For any variable $\mathbf{z}_i$ of $H^t$ that

$$u(\mathbf{z}_i) \leq \theta, \tag{5.16}$$

where $\theta \in [0, 0.5)$ specifies the threshold parameter, we fix the value of optimum solution as follows:

$$\tilde{\mathbf{z}}_i = \text{sgn}\left(\sum_{j=1}^{n} \mathbf{z}_i^j\right). \tag{5.17}$$

Since $\theta < 0.5$, $\tilde{\mathbf{z}}_i$ is guaranteed to take its value from $\{-1, +1\}$. We initialize the Ising Hamiltonian of next episode as $H^{t+1} = H^t$. After fixing the value of $\mathbf{z}_i$:

- we remove $\mathbf{h}_i$ from $H^{t+1}$;
- we add the value of $\tilde{\mathbf{z}}_i J_{ij}$ (or $\tilde{\mathbf{z}}_i J_{ji}$) to $\mathbf{h}_j$, for $j = 1, 2, \ldots, N$ and $i \neq j$;
- we remove the coupler $J_{ij}$ (or $J_{ji}$) from $H^{t+1}$.

GQA terminates when $H^t \equiv H^{t+1}$. If GQA ended without fixing all problem variables, we apply the MQC method [60] on the samples from the last stage of the GQA and assign values for the remaining problem variables.

Contracting variables with negligible uncertainties results in a new Ising Hamiltonian ($H^{t+1}$) which is smaller and sparser, compared to $H^t$. Hence, at each iteration of GQA, the remaining Ising Hamiltonian becomes easier to be solved with physical quantum annealers. Note that contracting variables reduces the number of qubits ($N$) correspondingly. Algorithm 5.3.1 illustrates the GQA process. Finally, we can apply a classical local search on $\tilde{\mathbf{z}}$ to increase the probability of finding the ground state.

### 5.3.2 Results

In this section, we proposed a hybrid approach, called greedy quantum annealing (GQA), for improving the performance of physical quantum annealers in finding the global minimum (i.e., the ground state) of Ising Hamiltonians, shown in (2.4). It is important to emphasize that, therefore, our objective was not to introduce a new greedy algorithm that can address combinatorial optimization problems nor to guarantee that the quantum annealers will find the global minimum of given Ising Hamiltonians.

Benchmarking optimization techniques with computationally hard problems that we know the optimum solution is a common practice. However, recent studies have demonstrated that applying MQC [60] on the drawn samples by the D-Wave quantum processors always finds the optimum solution of such problems [61]. Note that the current generation of the D-Wave quantum annealers is limited to represent a clique of size at most 63 and MQC may fail to find the global optimum of benchmark problems when we notably increase the number of variables, which is beyond the capacity of the current quantum an-

**Algorithm 5.3.1:** Greedy quantum annealing (GQA) algorithm for finding the ground state of Hamiltonians.

**Input:** $H, \theta$
**Output:** $\tilde{\mathbf{z}}$

$\tilde{\mathbf{z}} \leftarrow \{\}$
$H^t \leftarrow H$
**do**
  $H^{t+1} \leftarrow H^t$
  $Z \leftarrow \{\mathbf{z}^1, \mathbf{z}^2, \cdots, \mathbf{z}^n\} = \arg\min_{\mathbf{z}} H^t$
  **for** $i \leftarrow 1$ **to** $N$ **do**
    **if** $u(\mathbf{z}_i) \leq \theta$ **then**
      $\tilde{\mathbf{z}}_i \leftarrow \mathrm{sgn}\left(\sum_{j=1}^{n} \mathbf{z}_i^j\right)$
      $H^{t+1}_{\mathbf{h}_i} \leftarrow 0$
      **for** $j \leftarrow 1$ **to** $N$ **do**
        **if** $J_{ij} \in H^{t+1}$ **then**
          $H^{t+1}_{\mathbf{h}_j} \leftarrow H^{t+1}_{\mathbf{h}_j} + J_{ij}\tilde{\mathbf{z}}_i$
          $H^{t+1}_{J_{ij}} \leftarrow 0$
        **end**
        **if** $J_{ji} \in H^{t+1}$ **then**
          $H^{t+1}_{\mathbf{h}_j} \leftarrow H^{t+1}_{\mathbf{h}_j} + J_{ji}\tilde{\mathbf{z}}_i$
          $H^{t+1}_{J_{ji}} \leftarrow 0$
        **end**
      **end**
    **end**
  **end**
**while** $H^{t+1} \neq H^t$
**if** $|\tilde{\mathbf{z}}| < N$ **then**
  $\tilde{\mathbf{z}} \leftarrow \tilde{\mathbf{z}} \cup \mathrm{MQC}(Z)$
**end**

nealers. It is worth highlighting that MQC performs well only on sparse problems—when we reduce the sparsity of the problem, i.e., increasing the number of non-zero quadratic coefficients in Eq. (2.4)—performance of the MQC approaches to a local search heuristic like single-qubit correction (SQC).

Generating random problems (i.e., drawing linear and quadratic coefficients of an Ising Hamiltonian randomly, according to a specific distribution) results in hard problems

for benchmarking quantum annealers [99, 59, 60]. Hence, as a proof-of-concept, we use randomly generated Ising Hamiltonians for evaluating the performance of GQA.

#### 5.3.2.1 Experiment A

For every problem in this experiment, we generated a random graph of size 50 with a specified sparsity rate ($s \in \{0.05, 0.25, 0.5, 0.75, 1.0\}$). More specifically, we randomly selected edges from a complete graph with $N = 50$ nodes where the sparsity rate $s$ denoted the probability of selecting edges. Afterward, we set values of the corresponding biases and couplers randomly. We generated three different types of benchmark problems: (a) we picked values of biases and couplers randomly from $\{-1, +1\}$ (binary coefficients); (b) we used random numbers in $[-1, +1]$ with uniform distribution to assign values of $\mathbf{h}$ and $J$ (uniform coefficients); and (c) coefficients' values were drawn randomly from a standard normal distribution (normal coefficients).

Our objective in this study was to improve the quality of results (i.e., finding samples with lower energy value), attained by the D-Wave quantum annealers. Hence, as our ground-truth, we compared the performance of GQA with two different settings. Since applying spin-reversal-transforms (a.k.a. gauge transforms) has demonstrated to address the impact of analog errors of the physical quantum annealers [145], as our first ground-truth method, we compared GQA with quantum annealing with 10 spin-reversal transforms (here, denoted by QA). We also enabled the inter-sample delay to reduce the sample-to-sample correlations in successive reads/measurements, albeit longer execution time. As our second ground-truth method, we applied MQC on samples from QA—quantum annealing with 10 spin-reversal-transforms and enabled inter-sample delay—denoted by MQC.

For each QMI, we requested 1,000 samples for all methods. The uncertainty threshold in GQA was $\theta = 0.0$—i.e., all spins must have the same value so GQA can fix them for the next stage. We obtained this threshold empirically via evaluating the performance of GQA on a small problem set. In the next experiment, we have studied the impact of $\theta$ on the performance of GQA.

Since randomly generated problems are not compatible with the working graph of the current D-Wave quantum processors, we used the minor-embedding heuristic [35] for embedding the arbitrary random graphs to the Chimera topology of the D-Wave 2000Q quantum processors. In addition, GQA iteratively contracts variables whose uncertainties are negligible; thus, the Ising Hamiltonian in GQA has a dynamic structure, in terms of the number of qubits and their connectivity. In this sense, at each stage, GQA needs to embed the remaining Ising Hamiltonian to an executable QMI on the target quantum annealer.

Figure 5.3 illustrates the performance comparisons between GQA and QA in minimizing Ising Hamiltonians with binary, uniform and normal coefficients. For each case, we generated 100 random problems with the corresponding (Bernoulli, uniform or normal) distributions. In this experiment, all randomly generated Ising Hamiltonians had 50 spin variables ($N = 50$). For each sparsity rate, the corresponding column illustrates the number of times that QA has found a better sample (compared to GQA), number of times that QA and GQA demonstrated same performance (i.e., best samples from both methods had identical Ising energy values), and number of times that GQA has outperformed QA.

In the same manner, Fig. 5.4 illustrates the performance comparisons between GQA and MQC in minimizing the same 100 randomly generated Ising Hamiltonians with binary, uniform and normal coefficients. Note that all arrangements in this experiment (e.g.,

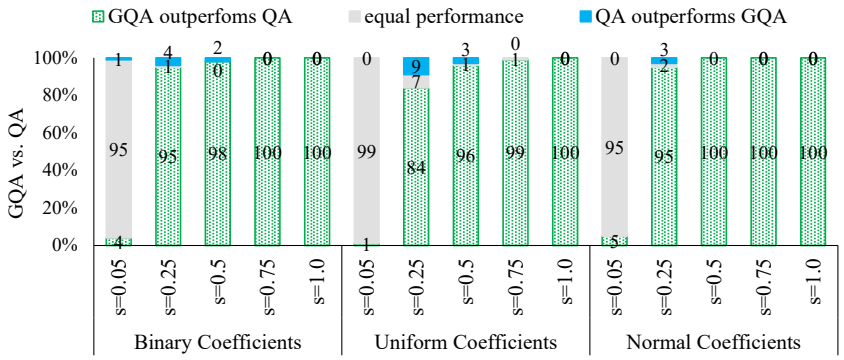


Figure 5.3: Performance comparison between GQA and QA (with spin-reversal-transforms and inter-sample delays) in solving 100 random benchmark problems with $N = 50$ spin variables and different sparsity rates $s$.

number of variables, sparsity rate, number of spin-reversal-transforms, etc.) were identical to the previous experiment. Similar to Fig. 5.3, each column in Fig. 5.4 represents the number of times that MQC has found a sample with lower energy, MQC and GQA had similar performance, and GQA resulted in a sample with lower Ising energy value.

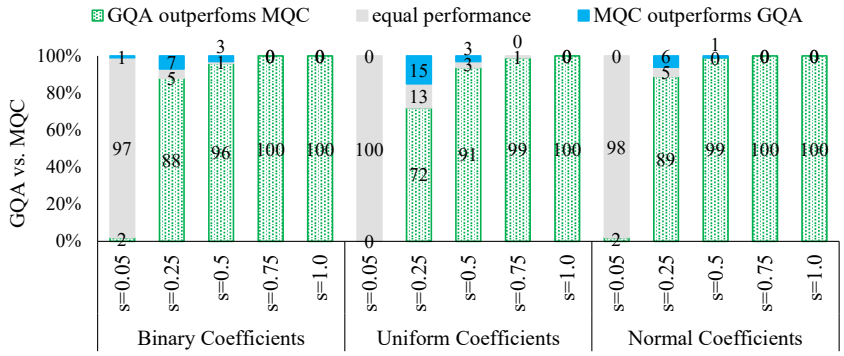


Figure 5.4: Performance comparison between GQA and MQC in solving 100 random benchmark problems with $N = 50$ spin variables and different sparsity rates $s$.

#### 5.3.2.2 Experiment B

In this experiment, we aim to study the impact of the threshold parameter on the performance of GQAA. To this end, we measured the average number of stages (iterations) that GQA takes to be converged. Table 5.1 illustrates the average number of iterations that GQA takes to solve 100 random benchmark problems with $N = 50$ variables and

normal coefficients. Since the randomly generated Ising Hamiltonians had arbitrary graph structures, we used the minor-embedding heuristic [35] for mapping them to executable QMIs on the D-Wave quantum annealers.

Table 5.1: Average number of iterations for GQA in solving 100 random benchmark problems with different thresholds ($\theta$) and sparsity rate ($s$).

| $\theta$ \ s | 0.05 | 0.25 | 0.50 | 0.75 | 1.00 |
|---|---|---|---|---|---|
| 0.25 | 3.25 | 2.80 | 3.15 | 3.40 | 3.30 |
| 0.15 | 3.60 | 3.40 | 3.35 | 3.45 | 3.65 |
| 0.05 | 4.10 | 4.45 | 4.30 | 3.20 | 4.05 |
| 0.00 | 5.50 | 6.20 | 2.10 | 2.05 | 2.30 |

Table 5.1 reveals that for Chimera like problems (i.e., sparse problems where $s = 0,05$ or $0.25$), when $\theta \to 0$, GQA takes more iterations (i.e., GQA has slower convergence). On the other side, for dense Ising Hamiltonians, where $s \to 1$ (i.e., clique like problems), when $\theta \to 0$, GQA converges quickly—the maximum number of iterations appears on $\theta \sim 0.1$.

## 5.4 Discussion

While most studies in quantum artificial intelligence aim to apply quantum computers for addressing hard AI problems, in this chapter we showed that AI can advance quantum information processing. In this chapter, we introduced a novel scheme—called reinforcement quantum annealing (RQA)—that leverages reinforcement learning (more specifically learning automata) to enhance the quality of results, attained by the quantum annealers. Ramezanpour (2018) has proposed to improve the simulated quantum annealing algorithm by adding reinforcement to the standard quantum annealing algorithm [151]. Simulated quantum annealing is an iterative algorithm (similar to simulated annealing)

that is implemented and run on classical computers. On the other hand, quantum annealers are physical, single-instruction, quantum processing units where the entire annealing process is atomic. Thus, we cannot modify or adjust the annealing process after starting the annealing (i.e., putting quantum bits on their superposition). in RQA, similar to the standard reinforcement learning scheme, iterations emulate the interactions between an agent and its environment. Ramezanpour's method, however, has an adaptive optimization scheme in which iterations simulate one quantum annealing process on a classical computer and is not applicable to physical quantum annealers.

As a proof-of-concept, we demonstrated that applying the proposed RQA scheme (on a D-Wave quantum annealer) results in notably better solutions. It is worth highlighting that, however, the proposed approach (i.e., hybridization of reinforcement learning and quantum annealing) is applicable to a vast range of classic AI problems like constraint satisfaction, planning, and scheduling.

We applied the proposed RQA scheme on two different SAT problem sets and compared its performance with quantum annealing (QA) and quantum annealing with post-quantum error corrections (so-called SMQC) which is the state-of-the-art in the realm of quantum annealing [81, 10, 60]. The first problem set includes 136 satisfiable SAT instances which represent factoring pseudo-prime numbers that have at most 63 Boolean variables, in CNF representation. Besides the length of the given composite numbers, the difficulty of integer factoring also depends on the properties of integer numbers. The hardest instances of this problem are factoring pseudo-prime numbers (product of two prime numbers) whose factors have the same size (in binary base). SAT instances in experiment A are not the hardest cases of prime-factoring— restricting the SAT instances in experiment A to a composite of the same size prime factors resulted in only 8 problems.

It is worth noting that our main objective in this experiment was not to address the prime factorization nor to use quantum annealers for solving SAT or MAX-SAT problems.

Figure 5.1 illustrates results—minimum, maximum, average and variance of a number of unsatisfied clauses—of solving these 136 satisfiable SAT instances for 100, 500, 1,000, 5,000 and 10,000 samples. In RQA, increasing the number of samples from 100 to 10,000 reduces the average number of unsatisfied clauses from 1.57 to 1.40. Similarly, in QA and SMQC, the average number of unsatisfied clauses is reduced from 9.41 and 3.55 to 6.85 and 3.17, respectively. Although increasing the number of samples in all three methods reduces the average number of unsatisfied clauses, RQA with 100 samples outperforms both QA and SMQC approaches in all arrangements (even with 10,000 samples). It is worth highlighting that QA was not able to satisfy all clauses of any of these 136 SAT instances, even when we requested for 10,000 samples, while both SMQC and RQA methods were able to find a satisfying solution for at least one of the instances in all cases. From a robustness viewpoint, increasing the number of samples from 100 to 10,000 lowers the variance and range (the difference between maximum and minimum) of QA, SMQC and RQA from 4.28, 2.60 and 0.86 to 3.21, 1.73 and 0.70, respectively. Therefore, RQA demonstrated better robustness (i.e., higher reproducibility rate), compared to both QA and SMQC approaches.

In the second study case, shown in Fig. 5.2, we used the first 100 SAT instances of the satisfiable benchmark test-set of uniform random 3-SAT with phase transitions [89]. Similar to Fig. 5.1, Fig. 5.2 demonstrates that RQA with 100 samples outperforms both QA and SMQC approaches in all settings. More specifically, increasing the number of samples from 100 to 10,000 in QA, SMQC and RQA decreases the average number of unsatisfied clauses from 8.02, 1.65 and 0.79 to 6.39, 1.28 and 0.68, respectively. Also, the

variances are reduced from 6.52, 0.53 and 0.31 to 4.43, 0.42 and 0.20, respectively. Note that the minimum number of unsatisfied clauses in RQA for all settings is zero while SMQC needed at least 5,000 samples to satisfy all clauses of at least one SAT instance. Note that RQA has an iterative scheme—RQA executes multiple QMIs for a given problem; hence, in all experiments, we restricted the total number of samples in RQA not to exceed the sample size of QA and SMQC. As an example, when QA and SMQC requested for 1,000 samples, RQA (with $T = 5$ iterations) asked for 200 samples in each iteration.

Our second proposal was leveraging quantum annealers to better select candidates in greedy algorithms. Quantum annealing is a meta-heuristic for addressing discrete optimization problems in near-constant time; however, owing to some technological barriers, physical quantum annealers (like the D-Wave quantum processors) generally result in excited states (i.e., close to global minimum), rather than converging to the ground state (i.e., global minimum). In other words, quantum annealers can find very high-quality solutions for combinatorial optimization problems in a fraction of a second; nevertheless, they generally fail to get to the ground state of the given Ising Hamiltonians.

We introduced a novel hybrid approach, called greedy quantum annealing (GQA), that employs the quantum annealers for making globally optimum choices at each stage of a greedy algorithm. To this end, we looked at the quantum annealers as a physical process that naturally draws samples from the ground state of Ising Hamiltonians (i.e., a problem-dependent Boltzmann distribution) at cryogenic temperatures. From a problem-solving point-of-view, conjugating quantum annealers and greedy algorithms address drawbacks of both methods and results in remarkably better solutions (i.e., samples with lower energy value), at the cost of executing multiple QMIs for one problem.

Our empirical results, using a D-Wave 2000Q quantum processor, on several randomly generated benchmark problems demonstrated that GQA finds samples with notably lower energy values, compared to the best-known techniques in the realm of quantum annealing—namely spin-reversal-transforms (or gauge transform) and multi-qubit correction (MQC). For sparse problems (i.e., the structure of the problem is close to the Chimera architecture), the performance of the GQA approaches to the performance of MQC. When the sparsity decreases, however, GQA shows supremacy in terms of finding samples with lower energy. In other words, GQA implicitly addresses the embedding-related issues, specifically the broken-chains.

# CHAPTER VI

# Compressive Sensing: A Study Case

## 6.1 Introduction

Digital systems employ analog-to-digital converters (ADC) and digital-to-analog converters (DAC) for acquiring data from the environment and representing the computational results, respectively. From a signal processing point of view, the Nyquist—Shannon theorem (also known as sampling theorem) plays a cornerstone role for bridging the gap between the continues-time (analog) and discrete-time (digital) signals. The sampling theorem introduces the rate which is sufficient for perfect reconstruction of the desired signal [105, 178, 149].

In real-world applications, remarkable portions of original signal sources are either sparse or compressible (possibly after a change of basis); therefore, traditional signal acquisition methods result in too many samples compared to the actual information contained in the original signal [74, 153, 127]. Also, when we perform necessary transforms on digital signals, the majority of transformed coefficients are discarded, and we only retain larger ones for storage and transmission purposes [153]. Moreover, in several applications like

multi-band signals having a wide spectral range, the suggested sampling rate by sampling theorem can exceed the specifications of best available analog-to-digital converters [153]. A novel tool to resolve these drawbacks is compressive sensing where we utilize linear measurements instead of point measurements [74, 153, 127].

Compressive sensing (a.k.a. compressed sensing, compressive sampling or sparse sampling) is a randomized data acquisition method that linearly samples sparse or compressible signals at a rate much below of the Nyquist-Shannon sampling theorem [36, 58, 37]. From a data acquisition point of view, instead of sensing $N$ samples uniformly and then compressing them into a vector of size $s$, compressive sensing performs $m$ linear measurements such that $m$ is reasonably close to $s$, more precisely

$$m = O\left(s \log \frac{N}{s}\right). \tag{6.1}$$

From a signal processing viewpoint, compressive sensing exploits the sparsity of signals through optimization methods to reconstruct the original signal from far fewer samples than the imposed rate by the sampling theorem. In other words—since the majority of signals are either self-sparse in their original domain or have a sparse representation in some transform domain—compressive sensing performs both sensing and size reduction tasks simultaneously and samples the signal at a rate much below the Nyquist sampling rate [74, 127].

### 6.1.1 Problem Definition and Related Works

In traditional signal processing approaches, sensing is much more complicated than the recovery of the original signals. In compressive sensing, however, the encoding process is simple and the signal reconstruction is NP-hard [128, 127]. For a given measurement

vector $\mathbf{y} \in \mathbb{R}^m$ and a design matrix $A \in \mathbb{R}^{m \times N}$ with $m \ll N$, the original problem of compressive sensing aims to recover a sparse signal $\mathbf{x} \in \mathbb{R}^N$ such that

$$\mathbf{y} = A\mathbf{x}. \tag{6.2}$$

Equation (6.2) represents an underdetermined system which can have infinite solutions and compressive sensing guarantees the uniqueness of sparse solutions under different conditions [74].

Let $\|\mathbf{x}\|_0$ stands for the sparsity level of $\mathbf{x}$, i.e., the number of nonzero entries of $\mathbf{x}$, the ultimate goal of compressive sensing is to solve the following problem:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{y} = A\mathbf{x}, \tag{6.3}$$

where we exploit the sparsity of $\mathbf{x}$ through $\ell_0$-minimization to reconstruct the original signal from far fewer samples than required by the sampling theorem [74, 127]. Problem (6.3)—i.e., the original problem of compressive sensing—is NP-hard [128, 15] due to its combinatorial nature in selecting the best support set and consequently, it is not tractable and must be handled indirectly in the realm of classical computing. Greedy algorithms like orthogonal matching pursuit (OMP) can address the $\ell_0$-minimization under some restrictive constraints [74, 127]. In many real-world applications, however, greedy algorithms suffer in recovering signals with high-enough accuracy [67]. From an application viewpoint, compressive sensing started to revolutionize real-world applications through convexifying the problem (6.3). Because $\ell_p$-norm approximates $\ell_0$-norm as nonnegative $p$ goes down to zero, the standard approach to tackle this (intractable) problem is exploiting

the following $\ell_p$-norm problem:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_p \quad \text{subject to} \quad \mathbf{y} = A\mathbf{x}. \tag{6.4}$$

and then investigating that under what conditions and for which $p$'s problems (6.3) and (6.4) appoint an identical solution. We generally apply additional stringent constraints on the design matrix to certify this possibility using tools such as restricted isometry constants or null and range space properties [19].

For $p > 1$, the unique solution of this strictly convex problem is generically full-support, i.e., each of its components is non-zero [168]. When $p = 1$, we can reduce problem (6.4) to a linear program, and sufficient conditions are available to guarantee the uniqueness of the optimal solution [126]. But this convex program and the original nonconvex $\ell_0$-norm problem appoint an identical and unique sparse solution under appropriate circumstances [74]. For instance, the mutual coherence of the measurement matrix $A$ is defined below:

$$\mu(A) = \max_{i \neq j} \frac{|\langle A_i, A_j \rangle|}{\|A_i\|_2 \|A_j\|_2}.$$

This simply seeks the largest correlation between two different columns and provides a sufficient condition for the equivalence of the $\ell_0$ and $\ell_1$ problems. More specifically, when

$$\|\mathbf{x}\|_0 < 0.5 \left(1 + \frac{1}{\mu(A)}\right),$$

$\mathbf{x} \in \mathbb{R}^N$ is an identical and unique solution for both problems (6.3) and (6.4) [66]. This property provides an improper upper bound because it is only practical when the original signal is highly sparse.

The measurement matrix $A$ has the restricted isometry property (RIP) of order $s$ if there exists $\delta_s \in (0,1)$ that satisfies the following:

$$(1-\delta_s)\|\mathbf{x}\|_2^2 \leq \|A\mathbf{x}\|_2^2 \leq (1+\delta_s)\|\mathbf{x}\|_2^2 \qquad \forall \mathbf{x};\ \|\mathbf{x}\|_0 \leq s.$$

The $\ell_0$-norm and $\ell_1$-norm problems appoint the same $s$-sparse (i.e., $\mathbf{x}$ includes at most $s$ nonzero entries) solution if [122]

$$\delta_{2s} < 0.4931.$$

The RIP demands each column submatrix $A_S$ with $card(S) \leq s$ to behave like an identity matrix, precisely, to have singular values in $[1-\delta_s, 1+\delta_s]$. Since this property involves all the $s$-tuples of columns of $A$, it is more rigorous than the mutual coherence (in fact, $\delta_2 = \mu$) so that it leads to better upper bounds on the sparsity level of a vector to be recovered. From the sparsity point of view, the RIP enables the compressive sensing to recover remarkably less sparse signals, compared to the mutual coherence property [122]. Convex optimization-based methods like basis pursuit (BP), Dantzig selector, and sub-gradient-based algorithms generally require significantly more computational resources but they can outperform greedy algorithms and hybrid approaches (e.g., compressive sampling matching pursuit and stage-wise OMP) in terms of recovery accuracy [74, 127].

Although the choice of $p = 1$ seems the most interesting as $\ell_1$ is the closest convex norm to $\ell_0$ [152] and convex optimization is extremely well-nourished, the shape of a unit ball associated with $\ell_p$-norm for $0 < p < 1$ motivates to explore this case as well. For certain values of such $p$, we can obtain not only more robust and stable theoretical guarantees but also much less restrictive conditions than $\ell_1$-norm recovery [159, 41]. For instance, a sufficient condition for recovering an $s$-sparse vector in the noiseless case via

$\ell_{0.5}$ minimization is

$$\delta_{3s} + 27\delta_{4s} < 26,$$

which is notably less restrictive than the analogous result for $\ell_1$-norm recovery [159] that requires

$$\delta_{2s} + 2\delta_{3s} < 1.$$

Compressive sensing relies on the sparsity of signals and linearly samples them at a rate much below the Nyquist sampling rate; however, the majority of signals in real-world applications are not precisely sparse in their original domain. Since the majority of signals have an approximately sparse representation in some transformed domain (e.g., wavelet domain for natural images, Fourier domain for speech signals, Radon domain for medical images, etc.), we extend $\ell_p$-norm recovery analysis to the case of compressible vectors that are practically more valuable [153]. Roughly speaking, a vector $\mathbf{x}$ is compressible or nearly $s$-sparse in $\ell_p$-norm if $\|\mathbf{x} - \mathbf{x}_S\|_p$ is small enough, where $S$ is an index set containing its $s$ largest absolute components. Besides, for handling noisy measurements, it is advantageous to apply the idea of penalty methods and address this LASSO-type unconstrained problem:

$$\min_{\mathbf{x}} \|\mathbf{y} - A\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_0. \tag{6.5}$$

There is a trade-off situation between feasibility and sparsity in the nature of this problem that we control it by the penalty parameter $\lambda$ [71]. Majority of results for recovering $s$-sparse signals in compressive sensing literature are based on the RIP constants; specifically in the form of $\delta_s$ or $\delta_{ks} \leq \delta$ for natural number $k > 0$ and $\delta \in (0, 1)$.

The probabilistic nature in such results for the standard problem of compressive sensing, as an important special case of compressive sensing with matrix uncertainty problem, is not favorable because such conditions are not indeed verifiable, due to its computational complexity [176]. Hence, there is an interest in constructing deterministic matrices with small RIP constants, but the number of measurements $m$ scales quadratically in sparsity level $s$ in the ongoing results. A breakthrough in this field took place when random matrices demonstrated their capability to satisfy such appealing inequalities with a high probability. As an illustration, for an $m \times N$ random matrix, where each entry is independently drawn according to Gaussian or Bernoulli distributions, we have

$$\delta_s \leq \delta^*$$

for

$$m \geq C\delta^{-2} s \ln\left(\frac{eN}{s}\right)$$

in which $C > 0$ does not depend on $s, m$ and $N$ [19]. It is worth noting that various applications, however, impose a measurement matrix that is not necessarily random. As a result, the recovery process is not guaranteed even with a high probability.

In standard compressive sensing, a major assumption is to have a perfect measurement matrix, although this is not appropriate for many vital practical circumstances such as the erroneous measurement matrices that emerge in parameter discretization, telecommunication, source separation, and model mismatch [86, 69]. In other words, a realistic model not only accounts for the measurement error $\mathbf{y}$ but also handles numerical errors in the measurement matrix $A$. A well-documented approach to model this issue is considering an additive perturbation for the measurement matrix and minimizing the

magnitude of this unknown uncertainty matrix along with the measurement error. There is extensive literature to investigate the impact of uncertainty assumption on the existing efficient algorithms in compressive sensing including Basis Pursuit and Compressive Sampling Matching Pursuit [190, 87, 43].

Compressive sensing with matrix uncertainty, associated to the Bayesian setting with prior informative, is a more recent approach that aims to reconstruct a sparse or compressible signal $\mathbf{x} \in \mathbb{R}^N$ and finding an uncertainty parameter vector $\mathbf{d} \in \mathbb{R}^r$ from a noisy measurement vector $\mathbf{y} \in \mathbb{R}^m$ such that

$$\mathbf{y} = A_0\mathbf{x} + \sum_{i=1}^{r} \mathbf{d}_i A_i \mathbf{x} + \mathbf{e}, \tag{6.6}$$

where $A_i \in \mathbb{R}^{m\times N}; \forall i$, and the unknown noise vector $\mathbf{e} \sim \mathcal{N}(0, I/\gamma)$ [142, 119].

### 6.1.2 Applications

Compressive sensing outperforms traditional data acquisition approaches where: (a) traditional sensing techniques are very time-consuming like magnetic resonance imaging (MRI) and functional MRI (FMRI); (b) energy efficiency is vital, for example, wireless sensor networks (WSN) and wireless body sensor nodes (WBSN); (c) sensing is too expensive, namely high-speed analog-to-digital converters (ADCs); and (d) we have to utilize few sensors such as non-visible wavelengths [153].

#### 6.1.2.1 Compressive Imaging

Traditional digital imaging (or digital image acquisition) techniques try to increase the quality of their visual representation of the objects through the increase in the number of measurements. For example, digital cameras utilize millions of pixels to provide a

high-quality representation of natural images. Compressive imaging is a novel trend in technology that employs compressive sensing techniques to reconstruct high-quality images from significantly fewer samples or measurements [153].

Among all proposed architectures, a single-pixel camera (a.k.a. one-pixel camera) has gained remarkable attention regarding the use of compressive sensing for image acquisition applications [153, 74]. In contrast to digital cameras that record millions of pixels simultaneously, the single-pixel camera produces digital images through recording only a few thousands of (single) pixels sequentially [63]. Current single-pixel camera prototypes take notably more time for acquiring an image (e.g., approximately 15 minutes for a one-shot ) and the resulting images have a lower resolution compared to available mega-pixel digital cameras. However: (a) single-pixel cameras require much less information to reconstruct an image; (b) the technology can make simpler, smaller, and cheaper digital cameras; and (c) single-pixel cameras can operate across a broader spectral range, compared to silicon-based digital cameras [63, 153].

Compressive sensing has also demonstrated outstanding performance in various radar imaging applications—including, but not limited to, synthetic aperture radar (SAR), inverse synthetic aperture radar (ISAR), through the wall imaging radar (TWR) and ground-penetrating radar imaging (GPR). Compressive sensing results in significantly higher-resolution radar images through much fewer measurements. It also offers advantages like robustness in ISAR imaging, and resistance to countermeasures and interception in SAR imaging [153]. Similarly, compressive sensing has advanced other imaging domains ranging from parallel and underwater imaging to microwave and sub-wavelength Imaging [153].

#### 6.1.2.2 Compressive Machine Learning

In many real-world machine learning applications, we generally perform size reduction techniques (e.g., feature selection and feature extraction methods) for not only subsiding the complexity of the learning models but also ebbing the impacts of irrelevant parameters. From a problem representation perspective, for a given machine learning task (like classification or regression), the original feature space generally consists of limited relevant factors and many irrelevant (or redundant) parameters [28, 132].

In compressive machine learning, we try to neglect the unnecessary parameters (without performing additional preprocessing techniques) to take advantage of their sparsity structure. In other words, the synergy of simultaneous sampling and size reduction paradigms in compressive sensing can result in better machine learning models [74]. As an example, compressive sensing based face recognition techniques have shown to be invariant to rotations, re-scaling, and translations of the data. In the same manner, compressive sensing has demonstrated more robust results in speaker recognition applications [153]. In addition, hybridization of compressive sensing and support vector machines (SVM) has shown to be as effective as (or even better than) the state of the art techniques in deep learning (namely AlexNet and RESNET) in recognizing handwritten digit images [135].

#### 6.1.2.3 NLP and Speech Processing

Compressive sensing has demonstrated remarkable advancements in speech processing applications—including, but not limited to, speech coding (e.g., exploiting the sparsity in phonological features), speech enhancement, pattern retrieval in the residual domain, and capturing sparse feature vectors from voiced and nonvoiced speeches [153].

In the same way, compressive sensing has started to benefit natural language processing (NLP) applications. As an illustration, compressive sensing has performed remarkable functionalities in information-preserving and low-dimensional embedding of n-grams for supervised and unsupervised text mining applications [6, 127]. Similarly, compressive NLP has shown outstanding functionalities in text classification applications like topic modeling and document categorization [160].

#### 6.1.2.4 Biomedical Applications

Compressive sensing has demonstrated significant enhancements in biomedical applications. As an example, traditional MRI techniques perform time-consuming processes to construct high-enough-resolution images for diagnosis purposes (i.e., several minutes for one patient). Compressive sensing has revealed that high-quality MRI is doable in much less time, compared to traditional MRI techniques [111, 112]. Compressive sensing has also demonstrated outstanding improvements in other biomedical applications—ranging from processing biological signals like an electrocardiogram (ECG), electroencephalographic (EEG) and neural signals to genomic sensing, DNA microarrays, the study of proteins and bacterial composition reconstruction [153].

#### 6.1.2.5 Remote Sensing

In remote sensing, unlike on-site sensing, we acquire the data from remote objects without making any physical contact. Recently, satellite-based and aircraft-based sensor technology has revolutionized the field of remote sensing which benefits a vast variety of applications—including, but not limited to, earth science, geography, ecology, agriculture, forestry, weather and hydrology. In many real-world remote sensing applications,

we generally have to face the VVVV concerns (volume, variety, velocity, and veracity of observables). Remote sensing applications are also limited to the bottlenecks in the communication systems [48, 12].

Recent studies try to employ compressive sensing techniques for subsiding the challenges in remote sensing via a remarkable reduction of the sampling rate thresholds. As an instance, compressive sensing based data fusion method has shown to enhance the resolution of the Advanced Microwave Sounding Unit (AMSU) images using the data from a collocated infrared/visible sensor [187]. A recent spatial-temporal data fusion approach based on Fixed Rank Kalman Filtering and Fixed Ranked Smoothing [48] has been used for a very large global remote sensing data set, namely, GOSAT and AIRS tropospheric CO2 [134].

#### 6.1.2.6 Other Applications

Recent studies have revealed that compressive sensing can benefit a vast variety of applications ranging from communication systems, video processing, manifold processing to micro/nano-electronics and VLSI [74, 153].

## 6.2 Why Quantum Compressive Sensing?

All $\ell_p-$norm sparse recovery methods, for $p \in (0, 1]$, assume that both $\ell_0$ and $\ell_p$ problems—shown in Eq. (6.3) and (6.4), respectively—appoint an identical solution, and RIP is the most commonly used tool to verify it. This is a crucial requirement; nevertheless, we generally neglect it in many applications because randomly GENERATED DESIGN matrices satisfy the RIP with a very high probability. There are several problems (like cluster-expansion) that find real-world applications in a vast range of disciplines

(e.g., material science, quantum chemistry, statistical and quantum mechanics, biophysics, nuclear physics, etc.), that impose non-random problem-specific coding matrices which do not necessarily satisfy the RIP. In addition, many relevant physical properties(e.g., thermodynamic phase-diagrams) depend on highly accurate low-energy states; thus, we cannot use greedy methods due to the recovery accuracy concerns [92]. In this section, we offer to directly tackle the $ell_0$-norm problem of sparse recovery in the realm of quantum computing.

## 6.3 SAT-based Compressive Sensing

Solving satisfiability instances can require exponentially large computational resources. Worst cases, nevertheless, are less likely to happen in practice and modern SAT solvers can handle real-world applications with thousands of variables and millions of clauses. We introduce a novel approach to reduce the original problem of compressive sensing (i.e., $\ell_0$-norm recovery), shown in (6.3), to Weighted-MAX-SAT instances. In this reduction, we assume that sparse signals come from noiseless sources and one needs to extend our approach for noisy measurements.

### 6.3.1 Reducing Compressive Sensing to Weighted-MAX-SAT

From a problem-solving perspective, the recovery module in compressive sensing receives the measurement vector $\mathbf{y} \in \mathbb{R}^m$ and the design matrix $A \in \mathbb{R}^{m \times N}$ as input, and recovers a sparse vector $\mathbf{x} \in \mathbb{R}^N$ that: (a) $\mathbf{x}$ satisfies $\mathbf{y} = A\mathbf{x}$; and (b) $\mathbf{x}$ has maximum number of zeros (i.e., the sparsest solution of $\mathbf{y} = A\mathbf{x}$).

In SAT-based compressive sensing, we first relax the key aspects of the original problem of compressive sensing—shown in Eq. (6.3)—and define two new problems as

follows:

$$f^1 := \mathbf{y} - A\mathbf{x} = 0, \tag{6.7}$$

and,

$$f^2 := \min_{\mathbf{x}} \|\mathbf{x}\|_0. \tag{6.8}$$

Afterward, we represent $f^1$ and $f^2$ as two separate SAT instances in CNF as follows:

$$f^1_{\text{SAT}} := \bigwedge_{i=1}^{M_1} C^1_i, \tag{6.9}$$

and,

$$f^2_{\text{SAT}} := \bigwedge_{i=1}^{M_2} C^2_i, \tag{6.10}$$

where $M_1$ and $M_2$ are total number of clauses in $f^1_{\text{SAT}}$ and $f^2_{\text{SAT}}$, respectively. Note that $f^2_{\text{SAT}}$ are two different Boolean expressions (in CNF), but over the same variables $\mathbf{x}$.

We then concatenate problems $f^1_{\text{SAT}}$ and $f^2_{\text{SAT}}$ via the "AND" operator as follows:

$$f_{\text{SAT}} := \bigwedge_{i=1}^{M_1} C^1_i \wedge \bigwedge_{i=1}^{M_2} C^2_i. \tag{6.11}$$

Since both problems $f^1_{\text{SAT}}$ and $f^2_{\text{SAT}}$ are in CNF, problem $f_{\text{SAT}}$ is also a Boolean formula in CNF. Also,

$$\forall \mathbf{x}, \mathbf{x} \models f_{\text{SAT}} \implies \mathbf{x} \models f^1_{\text{SAT}} \text{ and } \mathbf{x} \models f^2_{\text{SAT}}.$$

In other words, any $\mathbf{x}$ that satisfies problem $f_{\text{SAT}}$ will also satisfy problems $f^1_{\text{SAT}}$ and $f^2_{\text{SAT}}$.

Finally, we assign weights to all clauses in problem $f_{\text{SAT}}$ and form a Weighted-MAX-SAT problem as follows:

$$f_{\text{CS}} := \bigwedge_{i=1}^{M_1} w^1 C^1_i \wedge \bigwedge_{i=1}^{M_2} w^2_i C^2_i, \tag{6.12}$$

where $w^1$ and $w^2$ denote the corresponding weights for clauses in problems $f^1_{\text{SAT}}$ and $f^2_{\text{SAT}}$, respectively.

**Theorem 6.3.1.** *Assuming that, under appropriate conditions, the sparsest solution for* $\mathbf{y} = A\mathbf{x}$ *is unique,*

$$w^1 > \sum_{i=1}^{M_2} w_i^2, \tag{6.13}$$

*is sufficient to guarantee that the optimum solution of the Weighted-MAX-SAT instance* ($f_{\text{CS}}$)*—shown in problem* (6.12)*—is identical to the sparsest solution of the given original problem of compressive sensing—shown in problem* (6.3)*.*

*Proof.* Let $X = \{\mathbf{x}^1, \mathbf{x}^2, \dots\}$ be the set of all solutions for $\mathbf{y} = A\mathbf{x}$—regardless of the sparsity rate of solutions—and $\mathbf{x}^* \in X$ denotes the sparsest solution of the given problem of compressive sensing. The case $X = \emptyset$ is not a feasible scenario because compressive sensing (under some conditions) guarantees the existence and uniqueness of a sparse solution.

Let $\kappa(f_{\text{CS}}; \mathbf{x})$ specifies the total penalty of unsatisfied clauses when we use $\mathbf{x}$ as a model for the Weighted-MAX-SAT instance $f_{\text{CS}}$. We can represent the global optimum of $f_{\text{CS}}$ as:

$$\tilde{\mathbf{x}}^* = \arg\min_{\mathbf{x}} \kappa(f_{\text{CS}}; \mathbf{x}),$$

which implies that $\tilde{\mathbf{x}}^* \models f^1_{\text{SAT}}$–assuming that the given problem of compressive sensing has a (sparse) solution—and $\tilde{\mathbf{x}}^*$ can satisfy maximum possible clauses of $f^2_{\text{SAT}}$. We aim to show that $\tilde{\mathbf{x}}^* = \mathbf{x}^*$.

The case $\tilde{\mathbf{x}}^* \notin X$ implies that $\tilde{\mathbf{x}}^*$ is not a solution for $\mathbf{y} = A\mathbf{x}$; therefore,

$$\tilde{\mathbf{x}}^* \not\models f^1_{\text{SAT}}.$$

In other words, $\tilde{\mathbf{x}}^*$ cannot satisfy at least one of the clauses of $f^1_{\text{SAT}}$. According to Eq. (6.13), the penalty of not satisfying any of the clauses in problem $f^1_{\text{SAT}}$ is greater than the summation of penalties in problem $f^2_{\text{SAT}}$. Since $X \neq \{\}$:

$$\exists \hat{\mathbf{x}} \in X, \quad \hat{\mathbf{x}} \models f^1_{\text{SAT}},$$

which is a contradiction because it falsifies that $\tilde{\mathbf{x}}^*$ is the global optimum of the Weighted-MAX-SAT instance $f_{\text{CS}}$ because:

$$\kappa(f_{\text{CS}}; \hat{\mathbf{x}}) < \kappa(f_{\text{CS}}; \tilde{\mathbf{x}}^*).$$

The case $\tilde{\mathbf{x}}^* \in X$ and $\tilde{\mathbf{x}}^* \neq \mathbf{x}^*$ implies that:

$$\exists \hat{\mathbf{x}}, \quad \hat{\mathbf{x}} \models f^1_{\text{SAT}},$$

and,

$$\kappa(f_{\text{CS}}; \hat{\mathbf{x}}) < \kappa(f_{\text{CS}}; \tilde{\mathbf{x}}^*).$$

Considering the fact that the sparsest solution of the given problem of compressive sensing is unique, we will have:

$$\hat{\mathbf{x}} \neq \tilde{\mathbf{x}}^*,$$

which is again a contradiction to the initial assumption that $\tilde{\mathbf{x}}^*$ is the global optimum of the Weighted-MAX-SAT instance $f_{\text{CS}}$. □

It is crucial to highlight that we do not solve $f^1$ and $f^2$ independently. Indeed, we solve (6.12) which is the SAT-based representation of the original problem of compressive

sensing. In this formulation, $f^1$ defines the feasible domain for $f^2$, and $f^2$ exploits the feasible space to minimize the $\|\mathbf{x}\|_0$.

### 6.3.2 Proof-of-Concept

Restricting the vector $\mathbf{x}$ in the problem (6.3) to take its values from $\{0, 1\}$ leads to an NP-hard discrete optimization problem –called binary compressive sensing (BCS) [129, 106, 103]. From an application point of view, not only binary signal sources have real-world applications (e.g., event detection in wireless sensor networks, group testing, spectrum hole detection for cognitive radios, etc.), but we also can leverage it to other types of signals [129]. In this section, as a proof of concept, we show: (a) how one can use the proposed reduction method and map the original problem of binary compressive sensing to Weighted-MAX-SAT; (b) how to address the resulting Weighted-MAX-SAT instances—which are NP-hard, and (c) how to extend the demonstrated mapping to non-binary signals.

While most of the studies in compressive sensing focus on random design matrices with Gaussian distribution, solid theoretical and experimental results are available to guarantee that we can also exactly recover sparse (or compressible) signals from random binary matrices (i.e., with Bernoulli distribution) with a very high probability [189]. In the realm of binary compressive sensing, nevertheless, binary coding matrices have demonstrated remarkably lower performance compared to non-binary design matrices [129, 106]. As an example, sparse random design matrices with Bernoulli distribution can only recover highly sparse binary signals—i.e., $s/N < 0.1$ [129]. Several studies, therefore, have focused on generating non-binary design matrices to remediate the performance and quality of binary compressive sensing techniques [129]. Hence, without losing the generality

of the proposed SAT-based compressive sensing (i.e., one can use it for non-binary coding matrices), we focus on the most challenging arrangement where both $\mathbf{x}$ and $A$ are binary.

For a given measurement vector $\mathbf{y} \in \mathbb{N}^m$ and a coding matrix $A \in \{0,1\}^{m \times N}$ (where $m \ll N$), the objective in SAT-based binary compressive sensing is to construct a Weighted-MAX-SAT instance, shown in Eq. (6.12), over the binary vector $\mathbf{x} \in \{0,1\}^N$ and guarantee that both problems (6.3) and (6.12) appoint an identical solution.

We start with defining $f^1$, shown in Eq. (6.7) that tries to only solve $\mathbf{y} = A\mathbf{x}$. Because both $A$ and $\mathbf{x}$ take their values from $\{0,1\}$, calculating $\langle A_{i.}, \mathbf{x} \rangle$—i.e., the inner product of $\mathbf{x}$ and the $i^{\text{th}}$ row of $A$—is equivalent to finding the Hamming weight (or population count) of $\{A_{i1}\mathbf{x}_1, A_{i2}\mathbf{x}_2, \ldots, A_{iN}\mathbf{x}_N\}$, as follows:

$$\mathbf{u}_i = \langle A_{i.}, \mathbf{x} \rangle = \sum_{j=1}^{N} A_{ij}\mathbf{x}_j.$$

Hence, we can represent problem $f^1$ as:

$$\mathbf{y}_i = \mathbf{u}_i, \quad \forall i \in [1, m].$$

Since both $\mathbf{y}$ and $\mathbf{u}$ take their values from $\{0, 1, \ldots, N\}$, we can represent them in binary basis as $\hat{\mathbf{y}} \in \{0,1\}^{m(\lfloor \log_2 N \rfloor + 1)}$, and $\hat{\mathbf{u}} \in \{0,1\}^{m(\lfloor \log_2 N \rfloor + 1)}$, respectively. Afterward, we can use the "XNOR" Boolean logic operator for representing the $f^1$, in binary basis, as follows:

$$f^1 := \bigwedge_{i=1}^{m(\lfloor \log_2 N \rfloor + 1)} \neg \left( \hat{\mathbf{y}}_i \oplus \hat{\mathbf{u}}_i \right)$$

Finally, we represent $f^1$ in CNF and form $f^1_{\text{SAT}}$, shown in (6.9). For the given measurement vector $\mathbf{y}$, we can convert the basis and form the binary vector $\hat{\mathbf{y}}$. To construct the

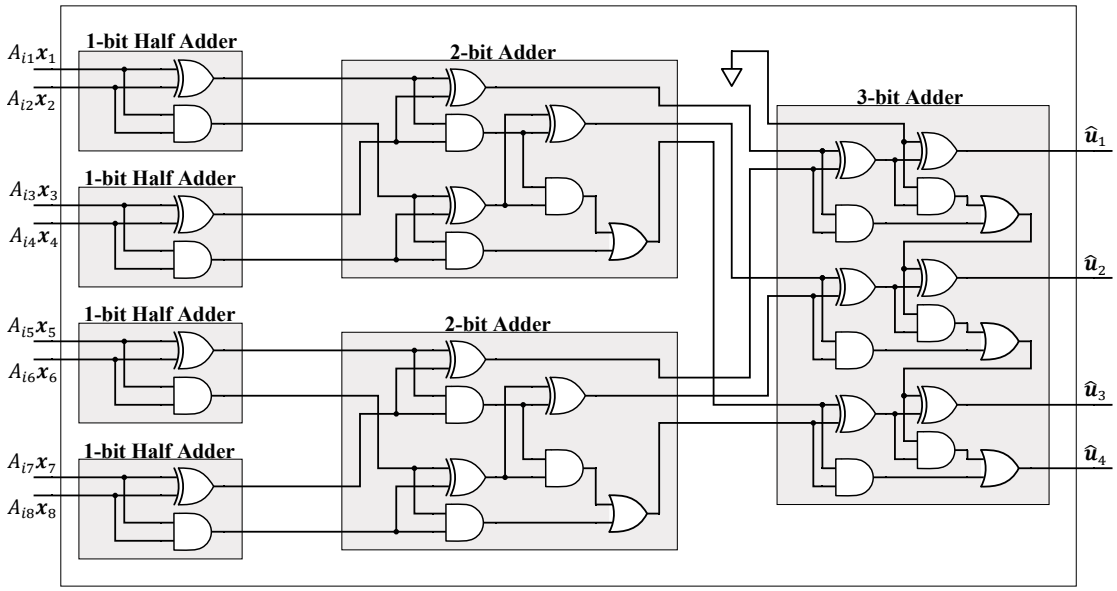


Figure 6.1: Logic circuit of a Hamming weight module for an 8-bit binary input.

binary vector $\hat{\mathbf{u}}$, we can employ the gate model of "half-adder" and "full-adder" modules in a tree-based structure and build a digital circuit for the problem of Hamming weight. Figure 6.1 illustrates an eight-bit Hamming weight module that generates the result of $\langle A_{i.}, \mathbf{x} \rangle$. In this tree-based structure, the Hamming weight module requires $O\left(\log_2 p_B N\right)$ full-adders where $p_B$ denotes the Bernoulli distribution parameter—i.e., $p_B$ and $1 - p_B$ denote the probability of generating "1" and "0", respectively. Hence, the encoding process can also take advantage of sparse coding matrices which generally results in smaller Boolean expressions. Figure 6.2 illustrates a prototype circuit for constructing the Boolean expression for representing $f^1$ when $N = 8$ and $m = 4$.

We know that the global minimum of $\|\mathbf{x}\|_0$ appears at the zero vector $\mathbf{x} = 0$. Therefore, we can represent $f^2$, in CNF, as follows:

$$f^2 := \bigwedge_{i=1}^{N} \neg \mathbf{x}_i,$$

and, form $f_{\text{CS}}$, shoen in Eq. (6.12).

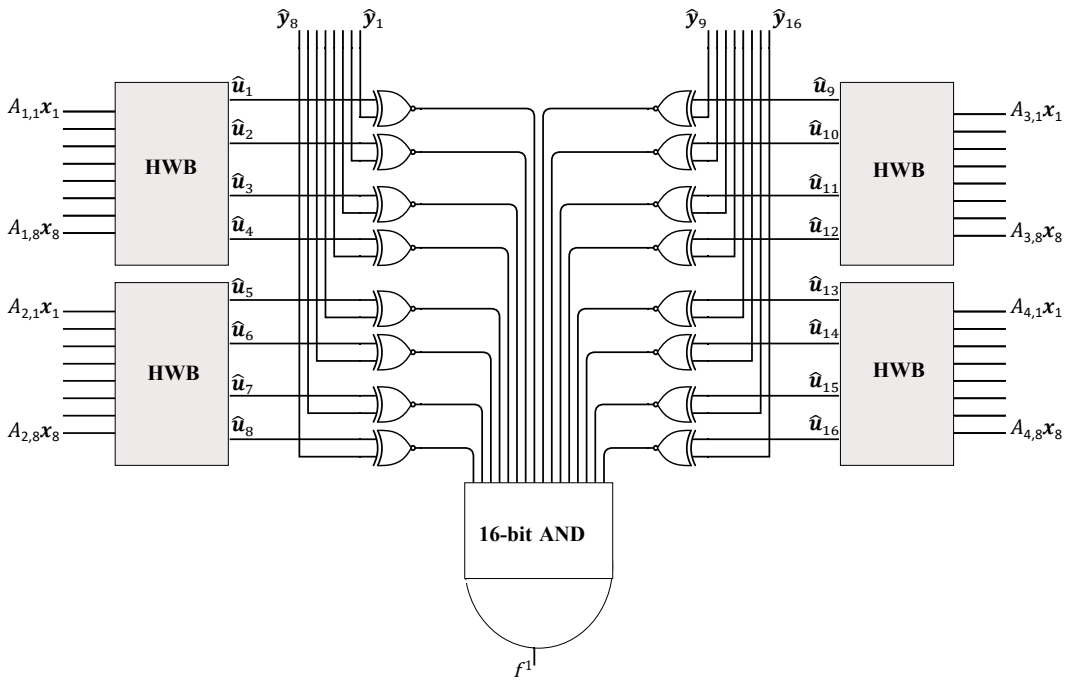


Figure 6.2: Logic circuit of $f^1$, $\mathbf{y} = A\mathbf{x}$, for binary signal of size 8, $\mathbf{x} \in \{0,1\}^8$, and $4 \times 8$ binary design matrix, $A \in \{0,1\}^{4\times 8}$, where HWB represents the module of Hamming weight in binary basis.

Satisfying (6.13) guarantees that the original problem of compressive sensing and the resulting Weighted-MAX-SAT appoint an identical solution. Because $\mathbf{x}$ is a binary vector, Eq. (6.13) reduces to:

$$w_1 > N w_2.$$

Note that the proposed reduction is applicable for non-binary signals. As an example, when $\mathbf{x} \in \{0,1,2,3\}^N$, we need two binary digits for representing each component of the signal $\mathbf{x}$ in binary basis. Correspondingly, Eq. (6.13) reduces to:

$$w_1 > 2N w_2.$$

For real-valued signals (i.e., $\mathbf{x} \in \mathbb{R}^N$), we can adopt binary representation of floating-points, which are generally available in logic synthesis tools like Z3 [51].

### 6.3.3 Results

For implementing the proposed symbolic SAT-based compressive sensing, we used Z3, a theorem prover by Microsoft Research [51]. Since $f^1$ is an arbitrary Boolean formula, we applied the Tseitin transformation [104] to represent $f^1$ in CNF and form $f^1_{\text{SAT}}$. Note that applying the Tseitin transformation can increase the size of $f^1_{\text{SAT}}$ linearly, due to defining auxiliary variables. We also applied pre-processing techniques—namely propagate-values, ctx-solver-simplify, recover-01, reduce-args and pb-preprocess tactics from Z3 [52]—to improve the performance of the symbolic computations.

To satisfy/optimize the ultimate Weighted-MAX-SAT instances, we used the core-guided solver with compressed MAX-SAT resolution [131], implemented in Z3—which is not guaranteed to find the global optimum. It is important to emphasize that our objective in this study is not to demonstrate the supremacy of SAT-based compressive sensing over the convex optimization techniques in sparse recovery. Indeed, we aim to show the possibility and impact of addressing the original problem of compressive sensing (i.e., $\ell_0$-norm sparse recovery)—rather than solving ill-posed cases. In these experiments, we used the bounded $\ell_1$-norm recovery method [103] for our comparisons because having any prior knowledge about the original signal can enhance the recovery.

In our first experiment, we measured the oversampling factor, $\frac{m}{s \log N/s}$, for different sparsity rates, $\frac{s}{N} \in [0.05, 0.5]$. To this end, for a fixed signal size (here, $N = 30$) and a specific sparsity rate, we generated 10 independent random examples. Since the sparsity rate of all instances is fixed, for every instance, we randomly selected the position of non-zero entries. For each test instance, we generated the coding matrix randomly with Bernoulli distribution—$p_B = 0.5$—where $p_B$ denotes the probability of generating "1"

and $1 - p_B$ represents the probability of generating "0", respectively. After generating the signal randomly (with the specified sparsity rate), we constructed the measurement vector through $\mathbf{y} = A\mathbf{x}$. For each method, i.e., SAT-based compressive sensing and bounded $\ell_1$-norm sparse recovery, the objective in this experiment was to find an appropriate $m$ such that we can exactly recover all 10 randomly generated test instances. In other words, instead of measuring the average recovery error, we adjust the number of required measurements to recover all elements of $\mathbf{x}$ correctly.

Figure 6.3 illustrates the results of this experiment, which reveals the supremacy of the proposed SAT-based compressive sensing over the bounded-$\ell_1$-norm recovery. For highly sparse signals ($s/N \approx 0.1$), the bounded-$\ell_1$-norm recovery needs on average 8.3% more measurements than the SAT-based compressive sensing. Figure 6.3 also shows that the SAT-based compressive sensing is less sensitive to the sparsity rate. More precisely, when the sparsity rate increases, the distance between the oversampling factors of bounded-$\ell_1$-norm recovery and SAT-based compressive sensing increases. As an example, for $s/N = 0.5$, the SAT-based compressive sensing can exactly recover the signals with 22.2% fewer measurements, compared to the bounded $\ell_0$-norm method.

In the second experiment, we keep the sparsity rate fixed and measure the average recovery error for different compression rates, $\frac{m}{N} \in [0.1, 1]$. For each specific compression rate, we generated 10 random test instances, for different signal sizes $N \in [20, 30]$, and measured the average recovery error over all test instances. Since $\mathbf{x} \in \{0, 1\}$, we divided the Hamming distance between the recovered and original signals by $N$ for representing the error for each recovery. Figure 6.4 illustrates the experiment results for $p_B = 0.5$, where $p_B$ denotes the probability of generating "1" in Bernoulli distribution, and $\frac{s}{N} = 0.5$. Similarly, Figure 6.5 shows the experiment results for $p_B = 0.3$ and $\frac{s}{N} = 0.3$. Not only

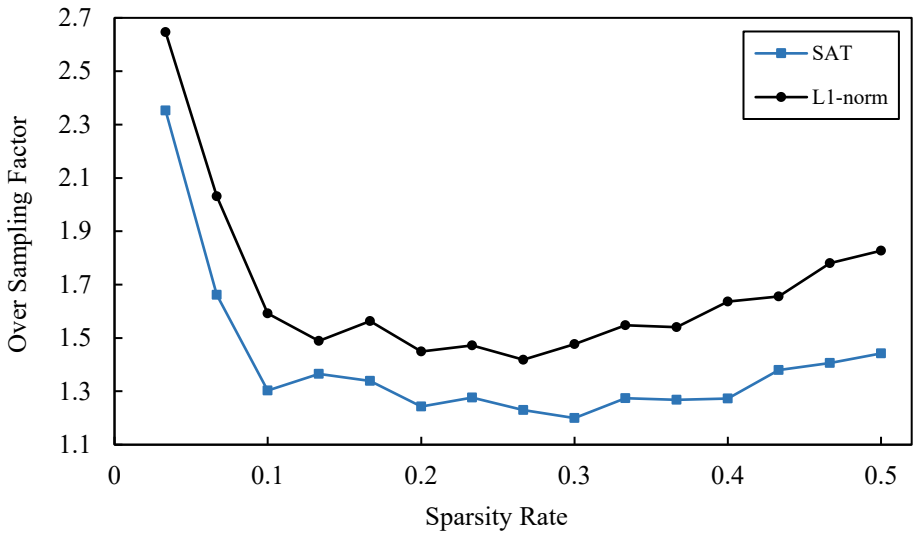


Figure 6.3: Performance comparison between SAT-based and bounded-$\ell_1$-minimization recovery based on optimum oversampling factor ($m/s \log N/s$) for different sparsity rate ($s/N \in (0, 0.5]$).

the SAT-based compressive sensing provides higher performance, but it also demonstrates more robust functionality.

## 6.4 Quantum Annealing based Binary Compressive Sensing

We introduce quantum annealing based binary compressive sensing (QABCS) that directly addresses the $\ell_0$-norm problem of binary compressive sensing to reconstruct a binary sparse or compressible signal ($\mathbf{x} \in \{0,1\}^N$) from a small number of linear measurements. To this end, we reformulate Eq. (6.5) for sparse binary signals as follows:

$$\min_{\mathbf{x}\in\{\mathbf{0},\mathbf{1}\}^{\mathbf{N}}} \|\mathbf{y} - A\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_0, \tag{6.14}$$

From an adiabatic quantum computing viewpoint, for a given measurement vector $\mathbf{y} \in \mathbb{R}^m$ and a coding matrix $A \in \mathbb{R}^{m\times N}$, we aim to define a QUBO form of an executable QMI, shown in (2.5), whoes ground state represents the sparsest solution of (6.14).

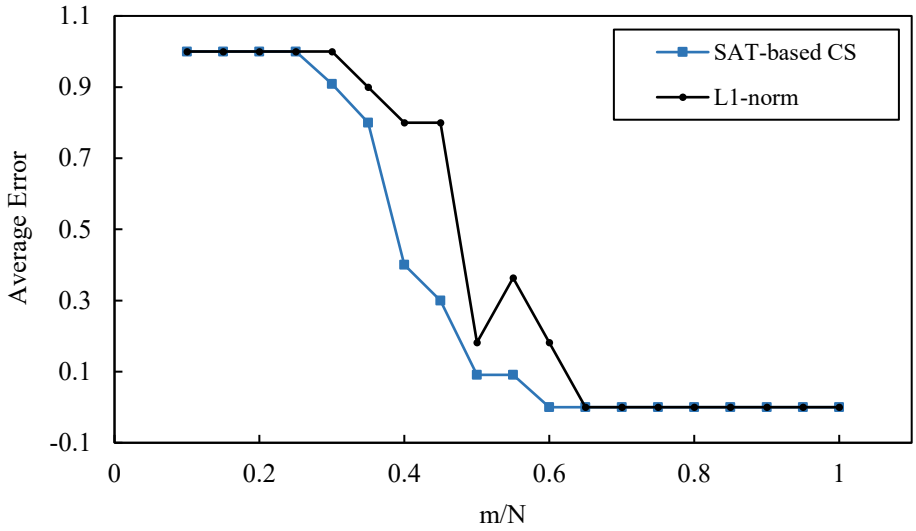


Figure 6.4: Recovery performance comparison for $p_B = 0.5$ and $s/N = 0.5$.

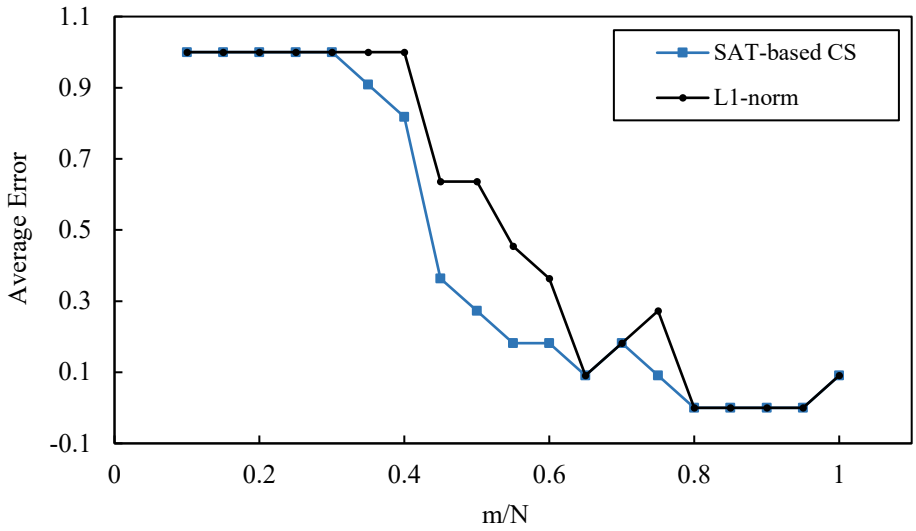


Figure 6.5: Recovery performance comparison for $p_B = 0.3$ and $s/N = 0.3$.

Hence, we need to rewrite

$$f(\mathbf{x}) := -2\mathbf{y}^T A\mathbf{x} + \mathbf{x}^T A^T A\mathbf{x} + \lambda\|\mathbf{x}\|_0$$

in the QUBO form, that is,

$$f(\mathbf{x}) = \sum_i Q_{ii}\mathbf{x}_i + \sum_{i<j} Q_{ij}\mathbf{x}_i\mathbf{x}_j.$$

We start with the simple quadratic function of $\mathbf{x}^T B\mathbf{x}$ for a square symmetric matrix $B$. First, note that

$$\begin{aligned}
\mathbf{x}^T B\mathbf{x} &= \sum_i \mathbf{x}_i (B\mathbf{x})_i = \sum_i \mathbf{x}_i \left( \sum_j B_{ij}\mathbf{x}_j \right) \\
&= \sum_{i,j} B_{ij}\mathbf{x}_i\mathbf{x}_j \\
&= \sum_i B_{ii}\mathbf{x}_i^2 + \sum_{i<j} B_{ij}\mathbf{x}_i\mathbf{x}_j + \sum_{i>j} B_{ji}\mathbf{x}_i\mathbf{x}_j \\
&= \sum_i B_{ii}\mathbf{x}_i^2 + \sum_{i<j} 2B_{ij}\mathbf{x}_i\mathbf{x}_j.
\end{aligned}$$

When $\mathbf{x}_i \in \{0, 1\}$, we have

$$\mathbf{x}^T B\mathbf{x} = \sum_i B_{ii}\mathbf{x}_i + \sum_{i<j} 2B_{ij}\mathbf{x}_i\mathbf{x}_j.$$

Thus, for binary $\mathbf{x}$, we see that:

$$\begin{aligned}
f(\mathbf{x}) &= -2\mathbf{y}^T A\mathbf{x} + \mathbf{x}^T A^T A\mathbf{x} + \lambda \sum_i \mathbf{x}_i \\
&= \sum_i \left( \sum_l -2\mathbf{y}_l A_{li} + \lambda \right) \mathbf{x}_i + \mathbf{x}^T A^T A\mathbf{x}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_i \left( \sum_l -2\mathbf{y}_l A_{li} + \lambda \right) \mathbf{x}_i + \sum_i B_{ii}\mathbf{x}_i + \sum_{i<j} 2B_{ij}\mathbf{x}_i\mathbf{x}_j \quad (\text{where} B = A^T A) \\
&= \sum_i \left( \sum_l -2\mathbf{y}_l A_{li} + \lambda \right) \mathbf{x}_i + \sum_i \left( \sum_l A_{li}A_{li} \right) \mathbf{x}_i + \sum_{i<j} \left( 2\sum_t A_{ti}A_{tj} \right) \mathbf{x}_i\mathbf{x}_j \\
&= \sum_i \left( \sum_l -2\mathbf{y}_l A_{li} + \lambda + \sum_l A_{li}A_{li} \right) \mathbf{x}_i + \sum_{i<j} \left( 2\sum_t A_{ti}A_{tj} \right) \mathbf{x}_i\mathbf{x}_j \\
&= \sum_i \left( \lambda + \sum_l A_{li}\left(-2\mathbf{y}_l + A_{li}\right) \right) \mathbf{x}_i + \sum_{i<j} \left( 2\sum_t A_{ti}A_{tj} \right) \mathbf{x}_i\mathbf{x}_j.
\end{aligned}
$$

The above implies that

$$
\|\mathbf{y} - A\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_0 - \|\mathbf{y}\|_2^2 = \sum_i Q_{ii}\mathbf{x}_i + \sum_{i<j} Q_{ij}\mathbf{x}_i\mathbf{x}_j;
$$

for the following coefficients:

$$
Q_{ii} = \lambda + \sum_l A_{li}\left(-2\mathbf{y}_l + A_{li}\right); \tag{6.15}
$$

$$
Q_{ij} = 2\sum_l A_{li}A_{lj}. \tag{6.16}
$$

### 6.4.1 Results

In this section, we aim to evaluate the performance and applicability of the proposed quantum annealing based binary compressive sensing (QABCS) method in the recovery of sparse binary signals on a D-Wave quantum processor.

#### 6.4.1.1 Generating Benchmark Problems

The current generation of the D-Wave quantum annealers (namely the D-Wave 2000Q QPUs) includes more than 2,000 qubits. Due to the sparse connectivity of qubits, however, they are limited to cliques of size at most 63. Hence, in this study, for all

randomly generated sparse binary signals (a.k.a. benchmark problems) $N = 60$. It is worth noting that for all benchmark problems in this experiment, we generated random coding matrices whose entries follow the standard normal distribution—i.e., average and standard deviation are 0 and 1, respectively. To avoid the impact of embedding (i.e., chaining multiple physical qubits for representing virtual qubits with higher connectivity) in our evaluations, for all test instances, we used a fixed embedding of a clique of size 60 on the current working graph of the D-Wave QPU. In the same manner, we set the chaining-strength of all problem embeddings to 1.5.

To guarantee the uniqueness of the sparse solution (with a high probability)—considering that $N = 60$—we also fixed the sparsity of randomly generated problems to $s = 5$. Accordingly, we generated three sets of benchmark problems with $m = 30, 40$ and 50. Several studies have demonstrated that the performance of sparse recovery is highly sensitive to the penalty parameter $\lambda$. Calibrating this penalty parameter is nontrivial and can become even intractable [192, 38, 25]. In addition, the optimum value of $\lambda$ can be different for every randomly generated coding matrix $A$.

We first generated a few random (normal) coding matrices with $N = 60$ columns and $m = 30, 40$ and 50 rows. For every randomly generated matrix, we generated a 5-sparse random binary signal, denoted by $\mathbf{x}$, and calculated the measurement vector as

$$\mathbf{y} = A\mathbf{x}.$$

Afterward, we applied Eq. (6.15) and (6.16) to represent the resulting problem of binary compressive sensing in a QUBO form. We used a D-Wave 2000Q quantum annealer to recover $\mathbf{x}$ from $\mathbf{y}$ and empirically found $\lambda = 16.5$ to be optimum.

After fixing the penalty parameter, $\lambda = 16.5$, for $m = 30, 40$ and 50 we repeatedly generated random problems (i.e., random coding matrix with normal distribution and a corresponding 5-sparse binary signal) until we find 10 problems whose recovery error is zero. Finally, for every coding matrix, we generated 10 random 5-sparse binary signals and calculated their corresponding measurement vectors. Hence, our benchmark problems include 100 random problems for $m = 30, 40$ and 50, 300 benchmark problems in total.

#### 6.4.1.2 Performance Analysis of QABCS

In this section, we apply the proposed quantum annealing based binary compressive sensing method (QABCS) on benchmark problems, described in 6.4.1.1. In this experiment, we requested for 1,000 samples/reads for all QMIs. After retrieving raw samples from a D-Wave QPU, we performed the majority voting scheme for remediating broken chains. We also applied SQC, shown in algorithm 3.1.1, on all samples and used the best sample (sample with lowest energy value) as the recovered binary signal, denoted by $\tilde{\mathbf{x}}$. For every recovery, we used

$$e = \frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_2^2}{N} \tag{6.17}$$

to measure the recovery error. Figure 6.6 illustrates the minimum, maximum, average and variance of recovery errors of applying QABCS on benchmark problems with different penalty parameters. In the same manner, Fig. 6.7 displays the minimum, maximum, average and variance of the sparsity of the recovered signals through employing QABCS on a D-Wave 2000Q quantum annealer. Figure 6.8 shows the number of perfect recoveries (in percent) for $m = 30, 40$ and 50. Similarly, Fig. 6.9 demonstrates the number of times (in percent) than QABCS was able to exactly recover 5-sparse signals, regardless of the recovery error.

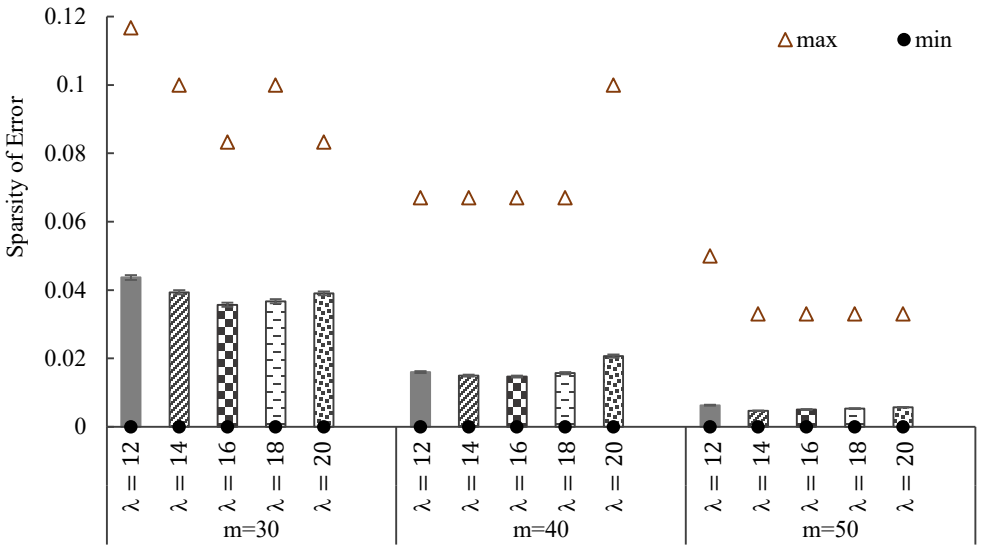


Figure 6.6: Recovery error of QABCS for 5-sparse binary signals of size 60 with a D-Wave 2000Q quantum processor.

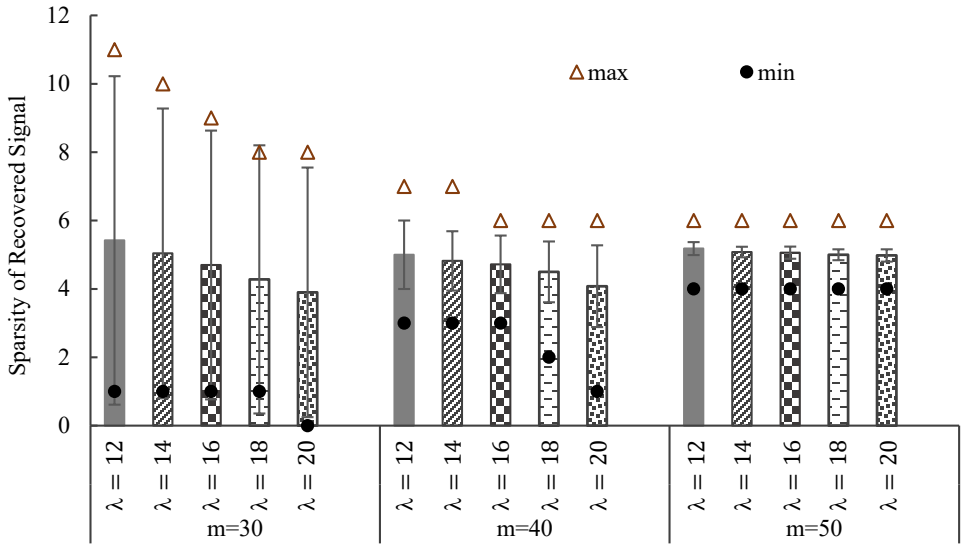


Figure 6.7: Sparsity rate of QABCS in recovery of 5-sparse binary signals of size 60 on a D-Wave 2000Q quantum processor.

## 6.5 Ensemble QA-based Binary Compressive Sensing

Although quantum annealers can draw samples from the ground state(s) of the given Ising Hamiltonians in near-constant time, the current generation of the quantum annealers have limitations that not only restrict the process of mapping problems into an executable quantum machine instruction but also lower the quality of results—including,

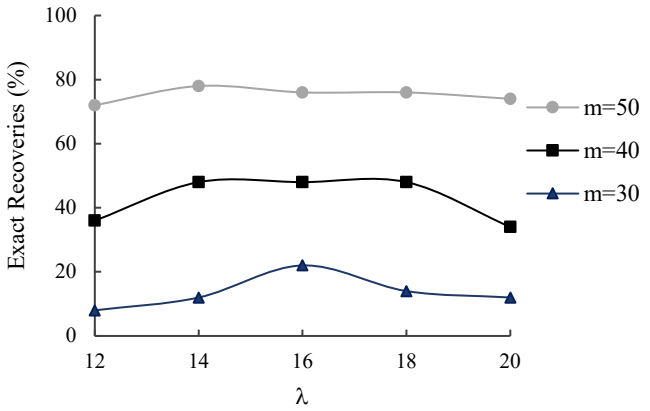


Figure 6.8: Exact recovery of 5-sparse binary signals of size $N = 60$ using QABCS on a D-Wave 2000Q quantum annealer.

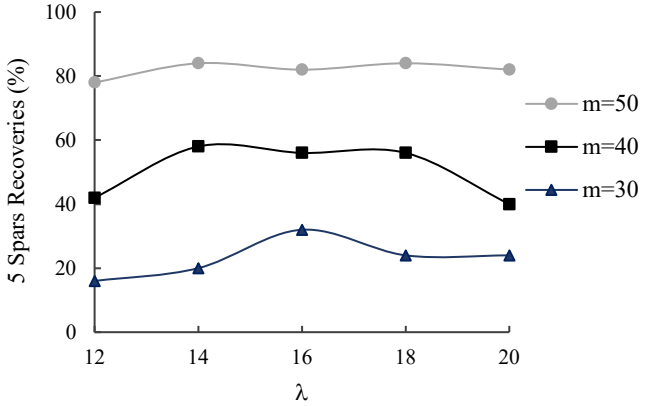


Figure 6.9: Exact 5-sparse recoveries using QABCS with a D-Wave 2000Q quantum annealer for $N = 60$.

but not limited to, sparse connectivity of the qubits, noise, decoherence and coefficients' range/precision limitations [10, 16].

In addition, one needs to find a proper value of the penalty parameter $\lambda$ prior to applying Eq. (6.15) and (6.16) for mapping the given BCS problem to a corresponding quantum machine instruction, executable by the quantum annealers. In practice, calibrating this penalty parameter is challenging, and can become even intractable [192, 38, 25]. In fact, $\lambda$ specifies the amount of shrinkage in Eq. (6.14). When $\lambda \to 0$ the number of eliminated parameters approaches zero (here, $\|\mathbf{x}\|_0 \to N$). On the other side, when $\lambda \to +\infty$, more parameters are eliminated (here, $\|\mathbf{x}\|_0 \to 0$).

In this study, instead of emphasizing on finding the optimum value for $\lambda$, which can be impractical in many real-world applications, we relax the mapping process to take multiple penalty parameters that are not necessarily optimum. Let

$$\Lambda = \{\lambda_1, \lambda_2, \dots, \}$$

be the set of different penalty parameters that we use for a given problem, and let

$$H = \{H_i, H_2, \dots\}$$

denotes the corresponding Ising Hamiltonians that we obtain from applying Eq. (6.15) and (6.16).

Since we assume that the sparsest solution of the given BCS problem is unique, when $|H| \rightarrow +\infty$, we can expect that the majority of the ground states be identical to the sparsest solution of the original problem of interest. Hence, after executing all (different) corresponding quantum machine instructions for a given problem, we aggregate the resulting samples and look at each element of samples as a binary random variable that follows the Bernoulli distribution. In this study, we perform a majority voting scheme for estimating the parameters of the corresponding Bernoulli distributions, while more advanced parameter estimation techniques can lead to better recovery approaches.

### 6.5.1 Results

In this section, we aim to evaluate the performance of the proposed ensemble quantum annealing based binary compressive sensing (EQABCS) in the recovery of benchmark 5-sparse binary signals, described in 6.4.1.1. For $\Lambda = \{12, 16, 20\}$, Fig. 6.10 illustrates

the recovery errors of EQABCS and compares it with the performance of QABCS for different penalty parameters. In the same manner, Fig. 6.11 presents the sparsity of recovered signals via EQABCS and compares them to the sparsity of the recovered signals, attained by QABCS. It is worth noting that all settings in this experiment (e.g., number of samples/reads, embeddings and chaining-strength) were identical to the arrangements in 6.4.1.

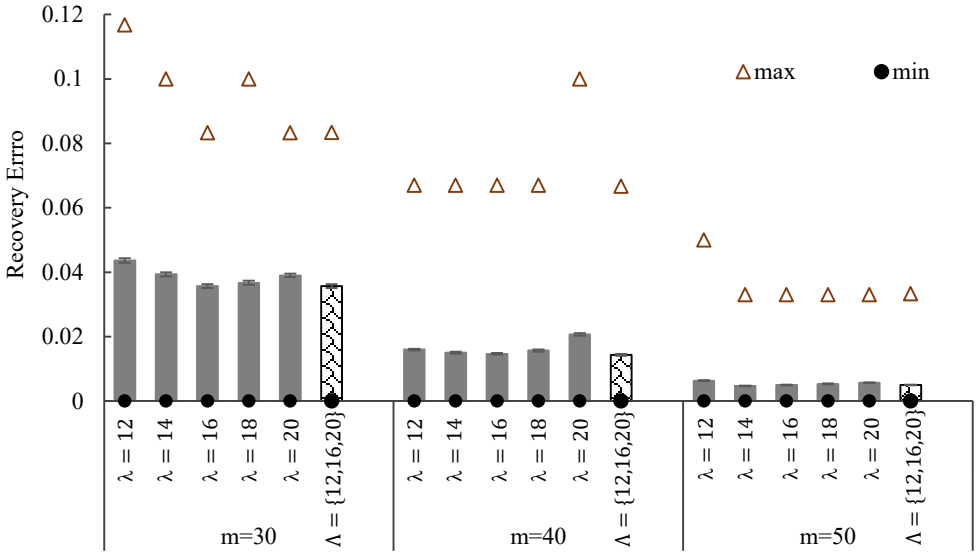


Figure 6.10: Recovery error of ensemble QABCS for 5-sparse binary signals of size 60 with a D-Wave 2000Q quantum processor.

## 6.6 QA-based Binary Compressive Sensing with Matrix Uncertainty

The objective in binary compressive sensing with matrix uncertainty is reconstructing a sparse binary signal $\mathbf{x} \in \{0,1\}^N$ and finding an uncertainty parameter vector $\mathbf{d} \in \mathbb{R}^r$ from a noisy measurement vector $\mathbf{y} \in \mathbb{R}^m$ such that

$$\mathbf{y} = A(\mathbf{d})\mathbf{x} + \mathbf{e}$$

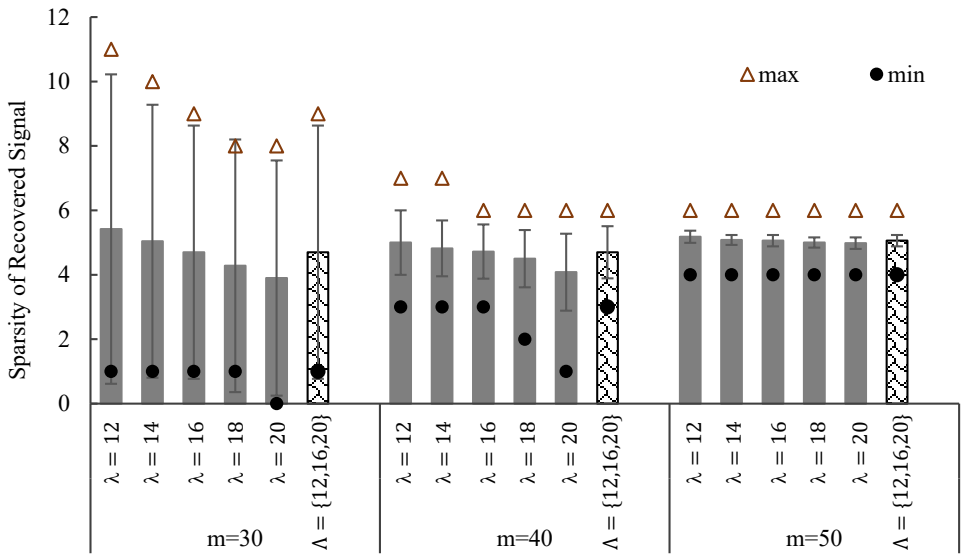


Figure 6.11: Sparsity rate of ensemble QABCS in recovery of 5-sparse binary signals of size 60 on a D-Wave 2000Q quantum processor.

where

$$A(\mathbf{d}) = A_0 + \sum_{i=1}^{r} \mathbf{d}_i A_i$$

and $A_i \in \mathbb{R}^{m \times N}; \forall i$, and the unknown noise vector$\mathbf{e} \sim \mathcal{N}(0, I/\gamma)$. To address the $\ell_0$-norm problem of binary compressive sensing with matrix uncertainty using quantum annealers, we extend Eq. (6.14) to

$$\min_{\mathbf{x}\in\{0,1\}^N, \mathbf{d}} \left\| \left( A_0 + \sum_{i=1}^{r} \mathbf{d}_i A_i \right) \mathbf{x} - \mathbf{y} \right\|_2^2 + \frac{\|\mathbf{d}\|_2^2}{\gamma} + \lambda \|\mathbf{x}\|_0, \tag{6.18}$$

where $\lambda$ and $\gamma$ are positive constants.

This problem entirely inherits favorable properties desired in the compressive sensing with matrix uncertainty by penalizing large sparsity and handling noise in the measurement matrix and vector. In other words, for a given noisy measurement vector $\mathbf{y}$, we can adjust the penalty parameter $\lambda$ to control the sparsity level of the resulting solution. The larger is this parameter; the smaller the sparsity level is desired so that we can adapt it

for specific applications. However, this problem is NP-hard and thus it is not tractable in the realm of classical computing. Here, we leverage the results from the previous section through an alternating minimization scheme to propose a well-posed approach for solving the problem of binary compressive sensing with matrix uncertainty that is tractable by quantum annealers.

Similar to the $\ell_0$-norm problem of compressive sensing, we start with the QUBO representation of the problem and compile it to an executable quantum machine instruction. To see this, let $\mathbf{d}$ be fixed and define

$$A := A_0 + \sum_{i=1}^{r} \mathbf{d}_i A_i,$$

then the first task is finding the QUBO form of:

$$\begin{aligned} f_{\mathbf{d}}(\mathbf{x}) : &= \left\| \left( A_0 + \sum_{i=1}^{r} \mathbf{d}_i A_i \right) \mathbf{x} - \mathbf{y} \right\|_2^2 + \lambda \|x\|_0 \\ &= \|A\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_0. \end{aligned}$$

Since $\mathbf{x}_i \in \{0, 1\}$, for a fixed uncertainty parameter $\mathbf{d}$, we can use (6.15) and (6.16) to rewrite the problem (6.18) as

$$f_{\mathbf{d}}(\mathbf{x}) - \|\mathbf{y}\|_2^2 = \sum_{i=1}^{N} Q_{ii} \mathbf{x}_i + \sum_{i<j} Q_{ij} \mathbf{x}_i \mathbf{x}_j.$$

Next, we assume that $\mathbf{x}$ is fixed so that the objective function which needs to be minimized is the following:

$$g_{\mathbf{x}}(\mathbf{d}) := \|(A_0 + \sum_{i=1}^{r} \mathbf{d}_i A_i)\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\|\mathbf{d}\|_2^2}{\gamma}.$$

By letting

$$G := [A_1\mathbf{x}\, A_2\mathbf{x}\, \ldots\, A_r\mathbf{x}]$$

and

$$\mathbf{c} := \mathbf{y} - A_0\mathbf{x},$$

we have

$$g_{\mathbf{x}}(\mathbf{d}) = \|G\mathbf{d} - \mathbf{c}\|_2^2 + \|\mathbf{d}\|_2^2/\gamma$$

so that the closed form solution of

$$\min_{\mathbf{d}} g_{\mathbf{x}}(\mathbf{d})$$

can be written as

$$\mathbf{d}^* = \left(G^T G + I/\gamma\right)^{-1} G\mathbf{c}. \tag{6.19}$$

Algorithm 6.6.1 incorporates the above information and illustrates our proposed well-posed solution for tackling the problem of binary compressive sensing with matrix uncertainty. First, we start our algorithm with initializing the elements of the uncertainty parameter $\mathbf{d}$ as zero vector and then use an iterative process in which we alternatively update the vectors $\mathbf{x}$ and $\mathbf{d}$. Once the vector $\mathbf{d}$ is fixed, we use the (6.15) and (6.16) to rewrite the resulting problem as a QUBO form and then employ the D-Wave quantum annealer to find the ground state of the resulting QUBO. After updating the vector$\mathbf{x}$, we assume that it is fixed and use the (6.19) to find the new uncertainty parameter $\mathbf{d}$. We repeat the iterations until a termination criterion is met.

**Algorithm 6.6.1:** Alternating minimization scheme for quantum annealing based binary compressive sensing with matrix uncertainty.

**Input:** $A_0, A_1, \ldots, A_r, \mathbf{y}, \lambda > 0, \gamma > 0$ and $\epsilon > 0$
**Output:** sparse vector $\mathbf{x}$ and uncertainty parameter $d$

$d \leftarrow 0$
**while** $\|(A_0 + \sum_{i=1}^{r} \mathbf{d}_i A_i)\mathbf{x} - \mathbf{y}\|_2 > \epsilon$ **do**
  $\mathbf{x} \leftarrow \arg\min_{\mathbf{x}\in\{0,1\}^N} f_{\mathbf{d}}(\mathbf{x})$
  $\mathbf{d} \leftarrow \mathbf{d}^*$ as given in (6.19);
**end**
**return** $\mathbf{x}, \mathbf{d}$

## 6.7 Discussion

Sparse recovery tries to infer high-dimensional sparse or compressible objects including vectors, matrices and high-order tensors from very limited observations. From a problem-solving point of view, sparse recovery has to deal with not only $\ell_0$-norm problem but also large-scale matrices that generally have remarkably more columns. In other words, we must deal with combinatorial problems that have large dimensions. The standard technique to address the $\ell_0$ problem is to replace it by $\ell_1$ and solve the new (possibly) convex problem. In compressive sensing, we are interested in matrices for which these problems attain an identical solution. These conditions are mostly NP-hard to verify, and also may not hold in some applications. Further, the size of the new approximation problem is prohibitive even if efficient interior-point methods are applied.

In this chapter, we showed how to reduce the original problem of compressive sensing (i.e., $ell_0$-norm sparse recovery) to the Weighted-MAX-SAT. The only required assumption here is to have a unique solution for the original problem of compressive sensing, which is much looser than those of relaxations or greedy methods. As a proof of concept, we demonstrated the applicability of our method in tackling the original problem of binary compressive sensing with binary design matrices. Experimental results revealed that the

proposed SAT-based compressive sensing outperforms the bounded-$\ell_1$-norm-based recovery in terms of not only the oversampling factor but also average recovery error. Also, experimental results demonstrated that the proposed SAT-based compressive sensing is less sensitive to the sparsity rate. More precisely, when the sparsity rate increases, the distance between the oversampling factors also increases.

To implement the symbolic computations, we used the Z3 framework, which is a theorem prover by Microsoft Research. Symbolic computing generally requires more computational resources, so we need to utilize high-performance computing for large-scale SAT-based problem-solving. The current release of Z3, however, does not support distributed computing for MAX-SAT. Thus, we performed experiments with remarkably small signals. Although the proposed SAT-based compressive sensing was able to outperform the bounded-$\ell_1$-norm recovery technique, it requires more computational resources. Hence, in practice, SAT-based compressive sensing is a proper choice where the lower compression rate ($m/N.$) is crucial.

It is worth highlighting that the employed heuristic for solving Weighted-MAX-SAT instances in this study is not guaranteed to find the global optimum. Hence, we can expect that SAT-based compressive sensing to demonstrate better performance when we apply more advanced optimization techniques. Furthermore, different quantum computing models—namely the gate (or circuit) model, adiabatic quantum computing and quantum annealers—have demonstrated a notable potential to address SAT and its extensions. Thus, this study can also bridge the gap between compressive sensing and quantum computing. Our method bridges the gap between the trends in advancing the SAT solvers and a broad range of compressive sensing based real-world applications. In standard compressive sensing, we assume that the measurements come from noiseless sources, and we also

have perfect knowledge about the coding matrix. In practice, however, such assumptions are not valid. As future work, we are extending the proposed SAT-based compressive sensing to handle the noisy measurements. Besides, we are leveraging the proposed model for tackling the problem of compressive sensing with matrix uncertainty, which is a more general problem that handles the case where we only can approximate the design matrices.

This chapter also introduces quantum annealing based binary compressive sensing (QABCS) that reduces the LASSO representation of the $\ell_0$-norm problem of binary compressive sensing to a QUBO form. Unlike SAT-based compressive sensing that assumes measurements come from noiseless sources, QABCS can handle the noise in linear measurements. On the other side, QABCS is limited to only binary signals.

In QABCS, the performance of the sparse recovery is highly sensitive to the penalty parameter. Calibrating the penalty parameter is nontrivial—in several cases, it can become intractable. Hence, we introduced the idea of ensemble QABCS (EQABCS) that leverages the idea of the statistical ensemble to improve the quality of quantum annealing based binary compressive sensing. Since executing quantum machine instructions on the quantum annealers can result in an excited state, rather than the ground state of the given Ising Hamiltonian, we use different penalty parameters to generate multiple distinct Ising Hamiltonians whose ground state(s) represent a potential solution of the original problem. We then employ the attained samples from minimizing all corresponding (different) Ising Hamiltonians to estimate the solution of the original problem of binary compressive sensing. Our experiments, on a D-Wave 2000Q quantum processor, demonstrated that the proposed ensemble quantum annealing approach is significantly less sensitive to the calibration of the penalty parameter $\lambda$.

Finally, we proposed to address the $\ell_0$-norm problem of binary compressive sensing with matrix uncertainty where we only have an approximation of the coding matrix. The proposed method extends the idea of alternating minimization to iteratively find the optimum values of the sparse binary signal and the uncertainty parameter of the measurement matrices, albeit executing multiple QMIs for a given problem.

# CHAPTER VII

## Conclusions

The era of Moore's law draws to a close and, therefore, the supercomputing community is exploring non-Von-Neumann computing architectures for emerging the next generation of accelerators in post-Moore era. Among the near-term accelerators—namely neural, quantum and Ising processing units—quantum annealers are potentially capable of providing the next disruption in post-Moore era, based on favorable aspects from both IPUs and QPUs. From a problem-solving perspective, not only is the programming quantum annealers nontrivial but are there several technological barriers that lower the quality of results—i.e., quantum annealing results in excited states rather than the ground state and results of the physical quantum annealers are not reproducible. In this thesis, we showed how to leverage different aspects of AI to compile classical algorithms for Ising processors, and improve the quality and reproducibility of results, attained by the quantum annealers.

## 7.1 Post-quantum Error Corrections

Quantum annealers can find very high-quality solutions in near-constant time. However, owing to some technological barriers of physical quantum annealers (e.g., sparse connectivity of qubits, coefficients' range and precision limitations, confined annealing, noise, decoherence, etc.), they generally fail to find the global optimum. Error correction methods in quantum computing generally require additional (auxiliary) qubits that significantly reduces the capacity of quantum computers in solving real-world problems. Chapter III offers to improve the probability of finding the global optimum without using extra qubits. We experimentally demonstrated that applying classical postprocessing techniques, namely SQC and MQC, can outperform recent software and hardware advances of quantum annealers. We also introduced a post-quantum error correction scheme for quantum annealers that outperforms the state-of-the-art techniques in the realm of quantum annealing. Our experimental results revealed that our post-quantum error correction scheme not only finds samples with lower energy but also requires fewer samples, compared to the best-known postprocessing techniques of quantum annealers. We also demonstrated that our proposed techniques can make results of the quantum annealers reproduceable.

### 7.1.1 Future Work

In section 3.3 we showed that the performance of MQC depends on not only the number of input samples but also their order. On this basis, we introduced randomized MQC (RMQC) that is guaranteed to outperform MQC, under some conditions, albeit more (classical) computation time. Our experimental results offered that we can reduce the number of required samples by RMQC, compared to MQC, without losing the performance

of the post-quantum error correction method. We plan to extend MQC via adopting selection methods in evolutionary computing—namely the roulette wheel and tournament selection—to enhance the performance of the post-quantum error correction in terms of the number of required samples and classical computations.

## 7.2 First Generation of Compilers for Ising Processing Units

Conventional computing machines (namely classical and gate model quantum computers) have a rich set of machine instructions for emulating (classical or quantum) algorithms. On the other side, quantum annealers are single-instruction computing machines that can only sample from the ground state of a given Ising Hamiltonian. To solve a problem on a quantum annealer, therefore, one needs to define coefficients of a quadratic objective function whose ground state represents a solution of the original problem of interest.

In chapter IV we introduced a novel (quantum) programming paradigm, called SAT++, that compiles classical algorithms (implemented in classical programming languages) to executable quantum machine instructions on D-Wave quantum annealers. Our model casts classical programs to SAT instances, as an intermediate representation of problems in SAT++, and represent the resulting SAT instances as minimizing Ising Hamiltonians to be compiled to executable QMIs. As a proof-of-concept, we demonstrated the applicability of SAT++ to address decision problems through quantum annealing.

### 7.2.1 Future Work

In this study, as a proof-of-concept, we APPLIED SAT++ to address decision problems on a D-Wave quantum annealer. It is worth emphasizing that SAT++ is not limited

to decision problems and one can use extensions of SAT to address further problem types. As an example, MAX-SAT and Weighted-MAX-SAT problems can serve the intermediate problem representation functionality of optimization problems in SAT++. In the same manner, #SAT can be used for model counting applications. To this end, we need to extend the proposed quantum annealing based SAT solver, described in section 4.3.2, to address extensions of SAT.

Representing SAT instances in CNF, in the worst-case scenario, can be exponentially time-consuming. Thus, we used the Tseitin transformation [104] to represent the resulting problems in CNF which runs in polynomial-time—albeit defining ancillary variables. Although applying the Tseitin transformation increases the size of the SAT linearly, resulting in problems that can quickly exceed the capacity of current quantum annealers. In the same manner to compilers that apply various code optimization techniques for enhancing the performance of the machine code on a specific CPU, employing preprocessing and postprocessing techniques can reduce the size of SAT instances (intermediate representation of problems in SAT++) which will increase the size of problems that SAT++ can address.

It is worth noting that one can adopt SAT++ for other Ising processing units, like optical parametric oscillators and CMOS annealers. In the realm of gate model quantum computing, several studies have adopted Grover's algorithm to propose quantum solutions for solving SAT (and its extensions). In this sense, we plan to extend SAT++ to gate model quantum computers. Specifically, we aim to apply SAT++ to compile classical algorithms for quantum programs, executable on ColdQuanta quantum processors.

## 7.3 Factoring Pseudo-Prime Numbers

As a proof-of-concept, in this study, we applied the proposed SAT++ framework for factoring pseudo-prime numbers that can jeopardize the security of modern public-key cryptography systems. We were able to successfully factor 12,317 to its prime factors (109 and 113) via applying a multiplication based factoring approach. It is worth noting that our research objective in this study was not to set a new record for quantum factorized integers, which for the current generation of the D-Wave quantum annealers is 1,005,973 [146].

### 7.3.1 Future Work

We aim to extend SAT++ via adding high-performance preprocessing tools that are available in SAT domain. These techniques have demonstrated to significantly reduce the size of CNF problems and we plan to set a new record for quantum factorized integers.

## 7.4 AI Hybrid Architectures for Quantum Annealing

Problem-solving with quantum annealers is a one-way process that starts from defining an Ising Hamiltonian, whose ground state represents a solution of the problem of interest, and ends with executing the corresponding QMI on a quantum processor, and possibly some postprocessing. In chapter V, we introduced two AI hybrid quantum annealing approaches and demonstrated that they can find samples with notably lower energy values. Reinforcement quantum annealing (RQA) has an iterative scheme in which an agent interacts with a quantum annealer that plays the role of stochastic environment in a learning automaton. On each iteration of RQA, we take all steps of a traditional quantum annealing based problem-solving, and we learn from previous iterations to define more efficient

Ising Hamiltonians. Greedy quantum annealing (GQA) also has an iterative scheme in which, we define values for some variables and contract them to form a new (smaller) Ising Hamiltonian. Unlike RQA that the agent solves a fixed problem in all iterations, GQA shrinks the problem via contracting those variables that the agent is confident about their values. Our experiment results demonstrated that both RQA and GQA outperforms best-known techniques in quantum annealing, specifically when problems are clique-like (i.e., we use longer chains in the embeddings).

### 7.4.1 Future Work

In this thesis, we proposed AI hybrid quantum annealing schemes and demonstrated that our models can find samples with notably lower energy values. However, the notion of the intelligent quantum annealing (or AI/ML-assisted quantum annealing) is not limited to RQA and GQA. We plan to explore more advanced models in artificial intelligence and machine learning to not only devise more advanced models but also address larger problems that are currently out of reach for quantum processors. As an example, we plan to adopt the idea of adversarial models in machine learning to devise more robust models. In addition, we aim to involve generative models for extending the idea of conjugating both sampling and optimization aspects of quantum annealers.

## 7.5 $\ell_0$-Norm Sparse recovery

In chapter VI, we offered to directly address the $\ell_0$-norm problem of compressive sensing. For noiseless signals, we introduced a novel method for reducing the original problem of compressive sensing to Weighted-MAX-SAT instances. For noisy measurements, we showed how to define the QUBO form of the problem of binary compressive sensing.

We also leveraged the idea of alternating minimization and introduced a novel method for addressing the problem of binary compressive sensing with matrix uncertainty using quantum annealers. The only requirement in our methods is the uniqueness of the sparse (or sparsest) solution that is much loser than conditions for convexifying the problem.

### 7.5.1 Future Work

For future work, we plan to extend SAT-based compressive sensing method to be solved on gate model quantum computers. Specifically, we will focus on ColdQuanta quantum processors for devising our quantum compressive sensing algorithm. Also, we aim to adopt our RQA model and propose a novel method for recovery of $s$-sparse signals—i.e., we can specify the sparsity level of the original signal.

## 7.6 Ensemble Quantum Annealing

For a given problem, there exist several distinct Ising Hamiltonians that share the same ground state, which is identical to a solution of the original problem of interest. These theoretically equivalent Hamiltonians, nevertheless, have different probabilities of finding the ground state when we execute their corresponding QMIs on a physical quantum annealer. This thesis introduced the idea of ensemble quantum annealing in which, we form and minimize multiple Ising Hamiltonians for a problem and aggregate the resulting samples to estimate the optimal solution. We experimentally demonstrated that ensemble quantum annealing can implicitly address the drawbacks of the quantum annealers and yields better solutions, specifically when the problem formulation is not robust.

### 7.6.1 Future Work

For future work, we aim to explore the applicability of learning models (namely semi-supervised learning techniques) to aggregate the resulting samples from minimizing different Ising Hamiltonians. We hypothesize that we can boost the performance of ensemble quantum annealing while we are reducing the number of Ising Hamiltonians of a problem.

# BIBLIOGRAPHY


[1] Dimitris Achlioptas, Carla Gomes, Henry Kautz, and Bart Selman. Generating satisfiable problem instances. *AAAI/IAAI*, 2000:256–261, 2000.

[2] Steven H Adachi and Maxwell P Henderson. Application of quantum annealing to training of deep neural networks. *arXiv preprint arXiv:1510.06356*, 2015.

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

[4] Patricia Amara, D Hsu, and John E Straub. Global energy minimum searches using an approximate solution of the imaginary time schrödinger equation. *The Journal of Physical Chemistry*, 97(25):6715–6721, 1993.

[5] Andris Ambainis. Quantum search algorithms. *arXiv preprint quant-ph/0504012*, 2005.

[6] Sanjeev Arora, Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. A compressed sensing view of unsupervised text embeddings, bag-of-n-grams, and lstms. 2018.

[7] Gilles Audemard and Laurent Simon. Predicting learnt clauses quality in modern sat solvers. In *Twenty-first International Joint Conference on Artificial Intelligence*, 2009.

[8] Gilles Audemard and Laurent Simon. On the glucose sat solver. *International Journal on Artificial Intelligence Tools*, 27(01):1840001, 2018.

[9] Ramin Ayanzadeh. Quantum artificial intelligence for natural language processing applications. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pages 273–273, 2018.

[10] Ramin Ayanzadeh, Milton Halem, John Dorband, and Tim Finin. Quantum-assisted greedy algorithms. *arXiv preprint arXiv:1912.02362*, 2019.

[11] Ramin Ayanzadeh, Milton Halem, and Tim Finin. Solving hard SAT instances with adiabatic quantum computers. In *AGU Fall Meeting Abstracts*, 2018.

[12] Ramin Ayanzadeh, Milton Halem, and Tim Finin. Compressive geospatial analytics. *AGUFM*, 2019:IN53B–0733, 2019.

[13] Ramin Ayanzadeh, Milton Halem, and Tim Finin. An ensemble approach for compressive sensing with quantum annealers. *University of Maryland, Baltimore County*, 2019.

[14] Ramin Ayanzadeh, Milton Halem, and Tim Finin. SAT++: A quantum programming paradigm. *University of Maryland, Baltimore County*, 2019.

[15] Ramin Ayanzadeh, Milton Halem, and Tim Finin. SAT-based compressive sensing. *arXiv preprint arXiv:1903.03650*, 2019.

[16] Ramin Ayanzadeh, Milton Halem, and Tim Finin. Reinforcement quantum annealing: A quantum-assisted learning automata approach. *arXiv preprint arXiv:2001.00234*, 2020.

[17] Ramin Ayanzadeh, Seyedahmad Mousavi, Milton Halem, and Tim Finin. Quantum annealing based binary compressive sensing with matrix uncertainty. *arXiv preprint arXiv:1901.00088*, 2019.

[18] Kannan Balasubramanian and Ahmed Mahmoud Abbas. Integer factoring algorithms. In *Algorithmic Strategies for Solving Complex Problems in Cryptography*, pages 228–240. IGI Global, 2018.

[19] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[20] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195, 2017.

[21] Zhengbing Bian, Fabian Chudak, Robert Israel, Brad Lackey, William G Macready, and Aidan Roy. Discrete optimization using quantum annealing on sparse ising models. *Frontiers in Physics*, 2:56, 2014.

[22] Zhengbing Bian, Fabian Chudak, Robert Brian Israel, Brad Lackey, William G Macready, and Aidan Roy. Mapping constrained optimization problems to quantum annealing with application to fault diagnosis. *Frontiers in ICT*, 3:14, 2016.

[23] Zhengbing Bian, Fabian Chudak, William Macready, Aidan Roy, Roberto Sebastiani, and Stefano Varotti. Solving sat and maxsat with a quantum annealer: Foundations and a preliminary report. In *International Symposium on Frontiers of Combining Systems*, pages 153–171. Springer, 2017.

[24] Zhengbing Bian, Fabian Chudak, William G Macready, and Geordie Rose. The ising model: teaching an old problem new tricks. *D-wave systems*, 2, 2010.

[25] Peter J Bickel, Ya'acov Ritov, Alexandre B Tsybakov, et al. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

[26] Armin Biere. Cadical, lingeling, plingeling, treengeling, yalsat entering the sat competition 2017. *SAT competition*, 2017:1, 2017.

[27] Armin Biere, Marijn Heule, and Hans van Maaren. *Handbook of satisfiability*, volume 185. IOS press, 2009.

[28] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[29] Rupak Biswas, Zhang Jiang, Kostya Kechezhi, Sergey Knysh, Salvatore Mandra, Bryan O'Gorman, Alejandro Perdomo-Ortiz, Andre Petukhov, John Realpe-Gómez, Eleanor Rieffel, et al. A nasa perspective on quantum computing: Opportunities and challenges. *Parallel Computing*, 64:81–98, 2017.

[30] Kelly Boothby, Paul Bunyk, Jack Raymond, and Aidan Roy. Next-generation topology of d-wave quantum processors. Technical report, Technical report, 2019.

[31] Ajinkya Borle and Samuel J Lomonaco. Analyzing the quantum annealing approach for solving linear least squares problems. In *International Workshop on Algorithms and Computation*, pages 289–301. Springer, 2019.

[32] Noureddine Bouhmala. Combining simulated annealing with local search heuristic for max-sat. *Journal of Heuristics*, 25(1):47–69, 2019.

[33] Hans J Briegel, David E Browne, Wolfgang Dür, Robert Raussendorf, and Maarten Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19, 2009.

[34] Benedikt Bünz and Matthew Lamm. Graph neural networks and boolean satisfiability. *arXiv preprint arXiv:1702.03592*, 2017.

[35] Jun Cai, William G Macready, and Aidan Roy. A practical heuristic for finding graph minors. *arXiv preprint arXiv:1406.2741*, 2014.

[36] Emmanuel Candes, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *arXiv preprint math/0409186*, 2004.

[37] Emmanuel Candes and Terence Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *arXiv preprint math/0410542*, 2004.

[38] Emmanuel Candes, Terence Tao, et al. The dantzig selector: Statistical estimation when p is much larger than n. *The annals of Statistics*, 35(6):2313–2351, 2007.

[39] Nicolas J Cerf, Lov K Grover, and Colin P Williams. Nested quantum search and np-hard problems. *Applicable Algebra in Engineering, Communication and Computing*, 10(4-5):311–338, 2000.

[40] Nicholas Chancellor, Stefan Zohren, Paul A Warburton, Simon C Benjamin, and Stephen Roberts. A direct mapping of max k-sat and high order parity checks to a chimera graph. *Scientific reports*, 6:37107, 2016.

[41] Rick Chartrand. Fast algorithms for nonconvex compressive sensing: Mri reconstruction from very few data. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 262–265. IEEE, 2009.

[42] Peter C Cheeseman, Bob Kanefsky, and William M Taylor. Where the really hard problems are. In *IJCAI*, volume 91, pages 331–337, 1991.

[43] Yuejie Chi, Louis L Scharf, Ali Pezeshki, and A Robert Calderbank. Sensitivity to basis mismatch in compressed sensing. *IEEE Transactions on Signal Processing*, 59(5):2182–2195, 2011.

[44] Bhargab Choudhury and Sangita Neog. Particle swarm optimization algorithm for integer factorization problem (ifp). *International Journal of Computer Applications*, 117(13), 2015.

[45] José Coelho and Mario Vanhoucke. Multi-mode resource-constrained project scheduling using rcpsp and sat solvers. *European Journal of Operational Research*, 213(1):73–82, 2011.

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

[47] David Roxbee Cox. *The theory of stochastic processes*. Routledge, 2017.

[48] Noel Cressie, Tao Shi, and Emily L Kang. Fixed rank filtering for spatio-temporal data. *Journal of Computational and Graphical Statistics*, 19(3):724–745, 2010.

[49] Arnab Das and Bikas K Chakrabarti. Colloquium: Quantum annealing and analog quantum computation. *Reviews of Modern Physics*, 80(3):1061, 2008.

[50] Pranav Dass, Harish Sharma, Jagdish Chand Bansal, and Kendall E Nygard. Meta heuristics for prime factorization problem. In *2013 World Congress on Nature and Biologically Inspired Computing*, pages 126–131. IEEE, 2013.

[51] Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer, 2008.

[52] Leonardo De Moura and Grant Olney Passmore. The strategy challenge in smt solving. In *Automated Reasoning and Mathematics*, pages 15–44. Springer, 2013.

[53] Qiang Deng, Dmitri V Averin, Mohammad H Amin, and Peter Smith. Decoherence induced deformation of the ground state in adiabatic quantum computation. *Scientific reports*, 3(1):1–6, 2013.

[54] David Devlin and Barry O'Sullivan. Satisfiability as a classification problem. In *Proc. of the 19th Irish Conf. on Artificial Intelligence and Cognitive Science*, 2008.

[55] Ronald A DeVore and Vladimir N Temlyakov. Some remarks on greedy algorithms. *Advances in computational Mathematics*, 5(1):173–187, 1996.

[56] Youcef Djenouri, Zineb Habbas, Djamel Djenouri, and Philippe Fournier-Viger. Bee swarm optimization for solving the maxsat problem using prior knowledge. *Soft Computing*, 23(9):3095–3112, 2019.

[57] Benjamin Doerr, Frank Neumann, and Andrew M Sutton. Time complexity analysis of evolutionary algorithms on random satisfiable k-cnf formulas. *Algorithmica*, 78(2):561–586, 2017.

[58] David L Donoho et al. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[59] John E Dorband. Extending the d-wave with support for higher precision coefficients. *arXiv preprint arXiv:1807.05244*, 2018.

[60] John E Dorband. A method of finding a lower energy solution to a qubo/ising objective function. *arXiv preprint arXiv:1801.04849*, 2018.

[61] John E Dorband. Applying multi-qubit correction to frustrated cluster loops on an adiabatic quantum computer. *arXiv preprint arXiv:1902.05827*, 2019.

[62] Raouf Dridi and Hedayat Alghassi. Prime factorization using quantum annealing and computational algebraic geometry. *Scientific reports*, 7:43048, 2017.

[63] Marco F Duarte, Mark A Davenport, Dharmpal Takhar, Jason N Laska, Ting Sun, Kevin F Kelly, and Richard G Baraniuk. Single-pixel imaging via compressive sampling. *IEEE signal processing magazine*, 25(2):83–91, 2008.

[64] Vedran Dunjko and Hans J Briegel. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, 81(7):074001, 2018.

[65] Niklas Eén and Niklas Sörensson. An extensible sat-solver. In *International conference on theory and applications of satisfiability testing*, pages 502–518. Springer, 2003.

[66] Michael Elad and Alfred M Bruckstein. A generalized uncertainty principle and sparse representation in pairs of bases. *IEEE Transactions on Information Theory*, 48(9):2558–2567, 2002.

[67] Yonina C Eldar and Gitta Kutyniok. *Compressed sensing: theory and applications*. Cambridge University Press, 2012.

[68] Andries P Engelbrecht. *Computational intelligence: an introduction*. John Wiley & Sons, 2007.

[69] Albert C Fannjiang, Thomas Strohmer, and Pengchong Yan. Compressed remote sensing of sparse objects. *SIAM Journal on Imaging Sciences*, 3(3):595–618, 2010.

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

[71] Mário AT Figueiredo, Robert D Nowak, and Stephen J Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of selected topics in signal processing*, 1(4):586–597, 2007.

[72] AB Finnila, MA Gomez, C Sebenik, C Stenson, and JD Doll. Quantum annealing: a new method for minimizing multidimensional functions. *Chemical physics letters*, 219(5-6):343–348, 1994.

[73] Gianluigi Folino, Clara Pizzuti, and Giandomenico Spezzano. Parallel hybrid method for sat that couples genetic algorithms and local search. *IEEE Transactions on Evolutionary Computation*, 5(4):323–334, 2001.

[74] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Springer Science & Business Media, 2013.

[75] Markus Frick and Martin Grohe. The complexity of first-order and monadic second-order logic revisited. *Annals of pure and applied logic*, 130(1-3):3–31, 2004.

[76] Samuel H Fuller and Lynette I Millett. *The Future of Computing Performance: Game Over or Next Level?* National Academy Press, 2011.

[77] Bartłomiej Gardas and Sebastian Deffner. Quantum fluctuation theorem for error diagnostics in quantum annealers. *Scientific reports*, 8(1):1–8, 2018.

[78] Bartłomiej Gardas, Jacek Dziarmaga, Wojciech H Zurek, and Michael Zwolak. Defects in quantum computers. *Scientific reports*, 8(1):4539, 2018.

[79] Michael R Garey and David S Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002.

[80] Luis Gil, Paulo Flores, and Luis Miguel Silveira. Pmsat: a parallel version of minisat. *Journal on Satisfiability, Boolean Modeling and Computation*, 6:71–98, 2008.

[81] John K Golden and Daniel O'Malley. Pre-and post-processing in quantum-computational hydrologic inverse analysis. *arXiv preprint arXiv:1910.00626*, 2019.

[82] Jens Gottlieb, Elena Marchiori, and Claudio Rossi. Evolutionary algorithms for the satisfiability problem. *Evolutionary computation*, 10(1):35–50, 2002.

[83] Jun Gu, Paul W Purdom, John Franco, and Benjamin W Wah. Algorithms for the satisfiability (sat) problem. In *Handbook of Combinatorial Optimization*, pages 379–572. Springer, 1999.

[84] Milton Halem, Asen Radov, and Devisha Singh. Comparisons of a quantum annealing and classical computer neural net approach for inferring global annual co2 fluxes over land. In *AGU Fall Meeting Abstracts*, 2017.

[85] Kathleen E Hamilton, Catherine D Schuman, Steven R Young, Ryan S Bennink, Neena Imam, and Travis S Humble. Accelerating scientific computing in the post-moore's era. *ACM Transactions on Parallel Computing (TOPC)*, 7(1):1–31, 2020.

[86] Matthew A Herman and Thomas Strohmer. High-resolution radar via compressed sensing. *IEEE transactions on signal processing*, 57(6):2275–2284, 2009.

[87] Matthew A Herman and Thomas Strohmer. General deviants: An analysis of perturbations in compressed sensing. *IEEE Journal of Selected topics in signal processing*, 4(2):342–349, 2010.

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

[89] Holger H Hoos and Thomas Stützle. Satlib: An online resource for research on sat. *Sat*, 2000:283–292, 2000.

[90] Andrei Horbach. A boolean satisfiability approach to the resource-constrained project scheduling problem. *Annals of Operations Research*, 181(1):89–107, 2010.

[91] Feng Hu, Lucas Lamata, Mikel Sanz, Xi Chen, Xingyuan Chen, Chao Wang, and Enrique Solano. Quantum computing cryptography: Finding cryptographic boolean functions with quantum annealing by a 2000 qubit d-wave quantum computer. *Physics Letters A*, page 126214, 2020.

[92] Wenxuan Huang, Alexander Urban, Penghao Xiao, Ziqin Rong, Hena Das, Tina Chen, Nongnuch Artrith, Alexandra Toumar, and Gerbrand Ceder. An l _0 l _1-norm compressive sensing paradigm for the construction of sparse predictive lattice models using mixed integer quadratic programming. *arXiv preprint arXiv:1807.10753*, 2018.

[93] Said Jabbour, Jerry Lonlac, Lakhdar Sais, and Yakoub Salhi. Extending modern sat solvers for models enumeration. In *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*, pages 803–810. IEEE, 2014.

[94] Mark W Johnson, Mohammad HS Amin, Suzanne Gildert, Trevor Lanting, Firas Hamze, Neil Dickson, R Harris, Andrew J Berkley, Jan Johansson, Paul Bunyk, et al. Quantum annealing with manufactured spins. *Nature*, 473(7346):194, 2011.

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

[96] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[97] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[98] Henry Kautz and Bart Selman. Satplan04: Planning as satisfiability. *Working Notes on the Fifth International Planning Competition (IPC-2006)*, pages 45–46, 2006.

[99] James King, Masoud Mohseni, William Bernoudy, Alexandre Fréchette, Hossein Sadeghi, Sergei V Isakov, Hartmut Neven, and Mohammad H Amin. Quantum-assisted genetic algorithm. *arXiv preprint arXiv:1907.00707*, 2019.

[100] Koki Kitai, Jiang Guo, Shenghong Ju, Shu Tanaka, Koji Tsuda, Junichiro Shiomi, and Ryo Tamura. Designing metamaterials with quantum annealing and factorization machines. *Physical Review Research*, 2(1):013319, 2020.

[101] Thaddeus D Ladd, Fedor Jelezko, Raymond Laflamme, Yasunobu Nakamura, Christopher Monroe, and Jeremy Lloyd O'Brien. Quantum computers. *nature*, 464(7285):45, 2010.

[102] Lucas Lamata. Basic protocols in quantum reinforcement learning with superconducting circuits. *Scientific reports*, 7(1):1609, 2017.

[103] Jan-Hendrik Lange, Marc E Pfetsch, Bianca M Seib, and Andreas M Tillmann. Sparse recovery with integrality constraints. *arXiv preprint arXiv:1608.08678*, 2016.

[104] Chu Min Li, Felip Manyà, and Joan Ramon Soler. Clausal form transformation in maxsat. In *2019 IEEE 49th International Symposium on Multiple-Valued Logic (ISMVL)*, pages 132–137. IEEE, 2019.

[105] Jae S Lim, Matthew M Bace, Alan V Oppenheim, Joseph E Bondaryk, Michael S Brandstein, Daniel T Cobra, Joseph E Covell, Meir Feder, Bruce A Musicus, Ramesh Patil, et al. Digital signal processing. Technical report, Research Laboratory of Electronics (RLE) at the Massachusetts Institute of Technology, 1988.

[106] Xiao Lin Liu, Chong Luo, and Feng Wu. Formulating binary compressive sensing decoding with asymmetrical property. In *2011 Data Compression Conference*, pages 213–222. IEEE, 2011.

[107] Samuel J Lomonaco. *Quantum Computation: A Grand Mathematical Challenge for the Twenty-first Century and the Millennium: American Mathematical Society, Short Course, January 17-18, 2000, Washington, DC*, volume 58. American Mathematical Soc., 2002.

[108] Samuel J Lomonaco et al. A rosetta stone for quantum mechanics with an introduction to quantum computation. In *Proceedings of Symposia in Applied Mathematics*, volume 58, pages 3–66, 2002.

[109] SJ Lomonaco. Shor's quantum factoring algorithm. In *Proceedings of Symposia in Applied Mathematics*, volume 58, pages 161–180, 2002.

[110] Daniel Lundén and Erik Forsblom. Factoring integers with parallel sat solvers, 2015.

[111] Michael Lustig, David Donoho, and John M Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

[112] Michael Lustig, David L Donoho, Juan M Santos, and John M Pauly. Compressed sensing mri. *IEEE signal processing magazine*, 25(2):72, 2008.

[113] Salvatore Mandra, Gian Giacomo Guerreschi, and Alán Aspuru-Guzik. Faster than classical quantum algorithm for dense formulas of exact satisfiability and occupation problems. *New Journal of Physics*, 18(7):073003, 2016.

[114] Joao Marques-Silva. Practical applications of boolean satisfiability. In *2008 9th International Workshop on Discrete Event Systems*, pages 74–80. IEEE, 2008.

[115] João P Marques-Silva and Karem A Sakallah. Boolean satisfiability in electronic design automation. In *Proceedings of the 37th Annual Design Automation Conference*, pages 675–680. ACM, 2000.

[116] Roman Martoňák, Giuseppe E Santoro, and Erio Tosatti. Quantum annealing by the path-integral monte carlo method: The two-dimensional random ising model. *Physical Review B*, 66(9):094203, 2002.

[117] Ohzeki Masayuki. Breaking limitation of quantum annealer in solving optimization problems under constraints. *Scientific Reports (Nature Publisher Group)*, 10(1), 2020.

[118] Catherine C McGeoch. Theory versus practice in annealing-based quantum computing. *Theoretical Computer Science*, 2020.

[119] Xiangming Meng and Jiang Zhu. Bilinear adaptive generalized vector approximate message passing. *IEEE Access*, 7:4807–4815, 2018.

[120] Ilya Mironov and Lintao Zhang. Applications of sat solvers to cryptanalysis of hash functions. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 102–115. Springer, 2006.

[121] Mohit Mishra, Vaibhav Gupta, Utkarsh Chaturvedi, Kaushal K Shukla, and Roman V Yampolskiy. A study on the limitations of evolutionary computation and other bio-inspired approaches for integer factorization. In *SCSE*, pages 603–610, 2015.

[122] Qun Mo and Song Li. New bounds on the restricted isometry constant $\delta$2k. *Applied and Computational Harmonic Analysis*, 31(3):460–468, 2011.

[123] Ashley Montanaro. Quantum walk speedup of backtracking algorithms. *arXiv preprint arXiv:1509.02374*, 2015.

[124] Ashley Montanaro. Quantum algorithms: an overview. *npj Quantum Information*, 2:15023, 2016.

[125] Gary J Mooney, Sam UY Tonetto, Charles D Hill, and Lloyd CL Hollenberg. Mapping np-hard problems to restricted adiabatic quantum architectures. *arXiv preprint arXiv:1911.00249*, 2019.

[126] Seyedahmad Mousavi and Jinglai Shen. Solution uniqueness of convex piecewise affine functions based optimization with applications to constrained $\ell_1$ minimization. *arXiv preprint arXiv:1711.05882*, 2017.

[127] Seyedahmad Mousavi, Mohammad Mehdi Rezaee Taghiabadi, and Ramin Ayanzadeh. A survey on compressive sensing: Classical results and recent advancements. *arXiv preprint arXiv:1908.01014*, 2019.

[128] Shanmugavelayutham Muthukrishnan et al. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1(2):117–236, 2005.

[129] Ukash Nakarmi and Nazanin Rahnavard. Bcs: Compressive sensing for binary sparse signals. In *MILCOM 2012-2012 IEEE Military Communications Conference*, pages 1–5. IEEE, 2012.

[130] Kumpati S Narendra and Mandayam AL Thathachar. *Learning automata: an introduction*. Courier Corporation, 2012.

[131] Nina Narodytska and Fahiem Bacchus. Maximum satisfiability using core-guided maxsat resolution. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[132] Nasser M Nasrabadi. Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4):049901, 2007.

[133] Grey Stephen Nearing, Milton Halem, David R Chapman, and CS Pelissier. Data assimilation on a quantum annealing computer: Feasibility and scalability. *AGUFM*, 2014:IN11A–3592, 2014.

[134] Hai Nguyen, Matthias Katzfuss, Noel Cressie, and Amy Braverman. Spatio-temporal data fusion for very large remote sensing datasets. *Technometrics*, 56(2):174–185, 2014.

[135] Nga Thi Thuy Nguyen and Garrett Kenyon. Comparing deep learning with quantum inference on the d-wave 2x. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2018.

[136] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[137] Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. Solving sat and sat modulo theories: from an abstract davis–putnam–logemann–loveland procedure to dpll (t). *Journal of the ACM (JACM)*, 53(6):937–977, 2006.

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

[139] Eugene Nudelman, Kevin Leyton-Brown, Holger H Hoos, Alex Devkar, and Yoav Shoham. Understanding random sat: Beyond the clauses-to-variables ratio. In *International Conference on Principles and Practice of Constraint Programming*, pages 438–452. Springer, 2004.

[140] Shuntaro Okada, Masayuki Ohzeki, Masayoshi Terabe, and Shinichiro Taguchi. Improving solutions by embedding larger subproblems in a d-wave quantum annealer. *Scientific reports*, 9(1):2098, 2019.

[141] Daniel O'Malley, Velimir V Vesselinov, Boian S Alexandrov, and Ludmil B Alexandrov. Nonnegative/binary matrix factorization with a d-wave quantum annealer. *PloS one*, 13(12):e0206653, 2018.

[142] Jason T Parker, Volkan Cevher, and Philip Schniter. Compressive sensing under matrix uncertainties: An approximate message passing approach. In *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 804–808. IEEE, 2011.

[143] Craig Pelissier, Jacqueline Le Moigne, Gyorgy Fekete, and Milton Halem. Quantum assisted learning for registration of modis images. *AGUFM*, 2017:IN12C–04, 2017.

[144] Craig Pelissier, Jacqueline Le Moigne, Milton Halem, and Troy Ames. Learning image registration and quantum annealing statistics. In *AGU Fall Meeting Abstracts*, 2018.

[145] Elijah Pelofske, Georg Hahn, and Hristo Djidjev. Optimizing the spin reversal transform on the d-wave 2000q. *arXiv preprint arXiv:1906.10955*, 2019.

[146] WangChun Peng, BaoNan Wang, Feng Hu, YunJiang Wang, XianJin Fang, XingYuan Chen, and Chao Wang. Factoring larger integers with fewer qubits via quantum annealing with optimized parameters. *SCIENCE CHINA Physics, Mechanics & Astronomy*, 62(6):60311, 2019.

[147] Alejandro Perdomo-Ortiz, Neil Dickson, Marshall Drew-Brook, Geordie Rose, and Alán Aspuru-Guzik. Finding low-energy conformations of lattice protein models by quantum annealing. *Scientific reports*, 2:571, 2012.

[148] Alejandro Perdomo-Ortiz, Joseph Fluegemann, Sriram Narasimhan, Rupak Biswas, and Vadim N Smelyanskiy. A quantum annealing approach for fault detection and diagnosis of graph-based systems. *The European Physical Journal Special Topics*, 224(1):131–148, 2015.

[149] John G Proakis and Dimitris G Manolakis. Digital signal processing, 2013.

[150] Kristen L Pudenz, Tameem Albash, and Daniel A Lidar. Quantum annealing correction for random ising problems. *Physical Review A*, 91(4):042302, 2015.

[151] A Ramezanpour. Enhancing the efficiency of quantum annealing via reinforcement: A path-integral monte carlo simulation of the quantum reinforcement algorithm. *Physical Review A*, 98(6):062309, 2018.

[152] Carlos Ramirez, Vladik Kreinovich, and Miguel Argaez. Why l1 is a good approximation to l0: A geometric explanation. *Journal of Uncertain Systems*, 7(3):203–207, 2013.

[153] Meenu Rani, SB Dhok, and RB Deshmukh. A systematic review of compressive sensing: Concepts, implementations and applications. *IEEE Access*, 6:4875–4894, 2018.

[154] Pulak Ray, Bikas K Chakrabarti, and Arunava Chakrabarti. Sherrington-kirkpatrick model in a transverse field: Absence of replica symmetry breaking due to quantum fluctuations. *Physical Review B*, 39(16):11828, 1989.

[155] Eleanor G Rieffel, Davide Venturelli, Bryan O'Gorman, Minh B Do, Elicia M Prystay, and Vadim N Smelyanskiy. A case study in programming a quantum annealer for hard operational planning problems. *Quantum Information Processing*, 14(1):1–36, 2015.

[156] Jussi Rintanen. Madagascar: Scalable planning with sat. *Proceedings of the 8th International Planning Competition (IPC-2014)*, 21, 2014.

[157] Francesca Rossi, Peter Van Beek, and Toby Walsh. *Handbook of constraint programming*. Elsevier, 2006.

[158] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[159] Rayan Saab, Rick Chartrand, and Ozgur Yilmaz. Stable sparse approximations via nonconvex optimization. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3885–3888. IEEE, 2008.

[160] Tara N Sainath, Sameer R Maskey, Bhuvana Ramabhadran, and Dimitri Kanevsky. Sparse representations for text classification, October 22 2013. US Patent 8,566,270.

[161] Giuseppe E Santoro, Roman Martoňák, Erio Tosatti, and Roberto Car. Theory of quantum annealing of an ising spin glass. *Science*, 295(5564):2427–2430, 2002.

[162] Giuseppe E Santoro and Erio Tosatti. Quantum to classical and back. *Nature Physics*, 3(9):593–594, 2007.

[163] Stefan Schoenmackers and Anna Cavender. Satisfy this: An attempt at solving prime factorization using satisfiability solvers. 2004.

[164] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[165] Roberto Sebastiani and Michele Vescovi. Automated reasoning in modal and description logics via sat encoding: the case study of k (m)/alc-satisfiability. *Journal of Artificial Intelligence Research*, 35:343–389, 2009.

[166] Bart Selman, Henry A Kautz, Bram Cohen, et al. Local search strategies for satisfiability testing. *Cliques, coloring, and satisfiability*, 26:521–532, 1993.

[167] Bart Selman, David G Mitchell, and Hector J Levesque. Generating hard satisfiability problems. *Artificial intelligence*, 81(1-2):17–29, 1996.

[168] Jinglai Shen and Seyedahmad Mousavi. Least sparsity of $p$-norm based optimization problems with $p > 1$. *SIAM Journal on Optimization*, 28(3):2721–2751, 2018.

[169] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

[170] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[171] Jennifer Sleeman, John Dorband, and Milton Halem. A hybrid quantum enabled rbm advantage: Convolutional autoencoders for quantum image compression and generative learning. *arXiv preprint arXiv:2001.11946*, 2020.

[172] Niklas Sorensson and Niklas Een. Minisat v1. 13-a sat solver with conflict-clause minimization. *SAT*, 2005(53):1–2, 2005.

[173] Niklas Sörensson and Niklas Eén. Minisat 2.1 and minisat++ 1.0-sat race 2008 editions. *SAT*, page 31, 2009.

[174] Juexiao Su, Tianheng Tu, and Lei He. A quantum annealing approach for boolean satisfiability problem. In *Proceedings of the 53rd Annual Design Automation Conference*, page 148. ACM, 2016.

[175] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[176] Andreas M Tillmann and Marc E Pfetsch. The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing. *IEEE Transactions on Information Theory*, 60(2):1248–1259, 2013.

[177] Tony T Tran, Minh Do, Eleanor G Rieffel, Jeremy Frank, Zhihui Wang, Bryan O'Gorman, Davide Venturelli, and J Christopher Beck. A hybrid quantum-classical approach to solving scheduling problems. In *Ninth annual symposium on combinatorial search*, 2016.

[178] Saeed V Vaseghi. *Advanced digital signal processing and noise reduction*. John Wiley & Sons, 2008.

[179] Davide Venturelli, Dominic JJ Marchand, and Galo Rojo. Quantum annealing implementation of job-shop scheduling. *arXiv preprint arXiv:1506.08479*, 2015.

[180] Walter Vinci and Daniel A Lidar. Non-stoquastic hamiltonians in quantum annealing via geometric phases. *npj Quantum Information*, 3(1):38, 2017.

[181] Yakir Vizel, Georg Weissenbacher, and Sharad Malik. Boolean satisfiability solvers and their applications in model checking. *Proceedings of the IEEE*, 103(11):2021–2035, 2015.

[182] Tomas Vyskocil and Hristo Djidjev. Simple constraint embedding for quantum annealers. In *2018 IEEE International Conference on Rebooting Computing (ICRC)*, pages 1–11. IEEE, 2018.

[183] Tomáš Vyskočil, Scott Pakin, and Hristo N Djidjev. Embedding inequality constraints for quantum annealing optimization. In *International Workshop on Quantum Technology and Optimization Problems*, pages 11–22. Springer, 2019.

[184] Philip Walther, Kevin J Resch, Terry Rudolph, Emmanuel Schenck, Harald Weinfurter, Vlatko Vedral, Markus Aspelmeyer, and Anton Zeilinger. Experimental one-way quantum computing. *Nature*, 434(7030):169, 2005.

[185] Haoze Wu. Improving sat-solving with machine learning. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, pages 787–788. ACM, 2017.

[186] Roman V Yampolskiy. Application of bio-inspired algorithm to the problem of integer factorisation. *International Journal of Bio-Inspired Computation*, 2(2):115–123, 2010.

[187] Igor Yanovsky, Ali Behrangi, Mathias Schreier, Van Dang, Berry Wen, and Bjorn Lambrigtsen. Fusion of microwave and infrared data for enhancing its spatial resolution. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 2625–2628. IEEE, 2017.

[188] Anocha Yimsiriwattana and Samuel J Lomonaco Jr. Distributed quantum computing: A distributed shor algorithm. In *Quantum Information and Computation II*, volume 5436, pages 360–372. International Society for Optics and Photonics, 2004.

[189] Gesen Zhang, Shuhong Jiao, Xiaoli Xu, and Lan Wang. Compressed sensing and reconstruction with bernoulli matrices. In *The 2010 IEEE International Conference on Information and Automation*, pages 455–460. IEEE, 2010.

[190] Hao Zhu, Geert Leus, and Georgios B Giannakis. Sparsity-cognizant total least-squares for perturbed compressive sampling. *IEEE Transactions on Signal Processing*, 59(5):2002–2016, 2011.

[191] Marko Žnidarič. Scaling of the running time of the quantum adiabatic algorithm for propositional satisfiability. *Physical Review A*, 71(6):062305, 2005.

[192] Hui Zou, Trevor Hastie, Robert Tibshirani, et al. On the degrees of freedom of the lasso. *The Annals of Statistics*, 35(5):2173–2192, 2007.