frontiers | Research Topics

Experience with quantum annealing computation

Edited by
Catherine McGeoch, Susan Mniszewski, Nicholas Chancellor and David Esteban Bernal Neira

Published in
Frontiers in Computer Science

ISSN 1664-8714
ISBN 978-2-8325-5436-4
DOI 10.3389/978-2-8325-5436-4

About Frontiers

Frontiers is more than just an open access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers journal series

The Frontiers journal series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the *Frontiers journal series* operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

Experience with quantum annealing computation

Topic editors

Catherine McGeoch — D-Wave Systems, Canada
Susan Mniszewski — Los Alamos National Laboratory (DOE), United States
Nicholas Chancellor — Durham University, United Kingdom
David Esteban Bernal Neira — Purdue University, United States

Citation

McGeoch, C., Mniszewski, S., Chancellor, N., Bernal Neira, D. E., eds. (2024). *Experience with quantum annealing computation*. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-8325-5436-4

Table of contents

frontiers | Frontiers in Computer Science

TYPE Editorial
PUBLISHED 03 September 2024
DOI 10.3389/fcomp.2024.1481330

OPEN ACCESS

EDITED AND REVIEWED BY
Sven Schewe,
University of Liverpool, United Kingdom

*CORRESPONDENCE
Catherine C. McGeoch
✉ cmcgeoch@dwavesys.com

†PRESENT ADDRESS
Nicholas Chancellor,
Newcastle University, Newcastle upon Tyne, United Kingdom

RECEIVED 15 August 2024
ACCEPTED 16 August 2024
PUBLISHED 03 September 2024

CITATION
Chancellor N, McGeoch CC, Mniszewski S and Bernal Niera D (2024) Editorial: Experience with quantum annealing computation. *Front. Comput. Sci.* 6:1481330. doi: 10.3389/fcomp.2024.1481330

Editorial: Experience with quantum annealing computation

Nicholas Chancellor[1†], Catherine C. McGeoch[2*], Susan Mniszewski[3] and David Bernal Niera[4]

[1]Department of Physics, Durham University, Durham, United Kingdom, [2]D-Wave Systems, Burnaby, BC, Canada, [3]Computer, Computational and Statistical Sciences (CCS) Division, Los Alamos National Laboratory (LANL), Los Alamos, NM, United States, [4]Purdue University, West Lafayette, IN, United States

KEYWORDS

quantum annealing algorithm, quantum performance analysis, quantum applications, quantum survey, quantum annealing, quantum annealer

Editorial on the Research Topic
Experience with quantum annealing computation

This *Frontiers Research Topic* comprises 13 peer-reviewed papers published in the period August 2023 through June 2024. These papers are cross-listed with *Frontiers in Computer Science*.

The call for submissions invited papers that address all aspects of empirical experience with annealing-based quantum computers, including: best practices in performance tuning; new approaches to hybrid computation (combining classical and quantum methods); estimation of resource usage; new application areas; and software tools and infrastructures. Original research papers as well as tutorials and surveys are within scope.

The submitting authors stepped up to the challenge, submitting works that cover a broad variety of Research Topics and formats. Here is a brief synopsis of each paper.

Early steps toward practical subsurface computations with quantum computing: Greer and O'Malley describe an application of quantum annealing to the problem of seismic inverse analysis, which arises in subsurface hydrology (geoscience). They show how to formulate the problem as a QUBO and present demonstration of the method using two real-world problems. They observe that the approach works well on low-contrast inputs but is challenged by higher precision requirements in problem representation for high-contrast inputs.

Tutorial: calibration refinement in quantum annealing: Chern et al. present a tutorial for calibration refinement, also known as *shimming*, to improve performance of quantum annealers, which are susceptible to crosstalk, device variation, and environmental noise. Explanations and code are supplied showing how to find symmetries for suppressing bias in outputs. Examples are provided for finding small subgraphs in the connectivity graph and finding symmetries of an Ising model via generalized graph automorphism. Other methods, implementations, and limitations are discussed.

Posiform planting: generating QUBO instances for benchmarking: Hahn et al. use planted-solution techniques to develop input generators to be used for benchmarking quantum annealers. Using a technique called posiform planting, they demonstrate that it is possible to construct large QUBO problems with known optimal solutions, thus allowing an evaluation of success probability on large problems. These instances are run on D-Wave quantum annealers to test performance.

Quantum image denoising: a framework via Boltzmann machines, QUBO, and quantum annealing: Kerger and Miyazaki examine applications relating to image de-noising for Boltzmann machines. This work is interesting because it presents two steps where annealers can potentially be used: in the training of the network, and in a separate de-noising step that is formulated as a QUBO. In experimental tests with D-Wave platforms

these methods were indeed found to be successful in reducing the noise in images.

Individual subject evaluated difficulty of adjustable mazes generated using quantum annealing: Ishikawa et al. consider the application of quantum annealing to the problem of generating maze puzzles that are difficult for humans to solve. They develop a cost function Q_{update} that scores problem difficulty for a given individual, and show how to formulate the cost function as a QUBO. They present an analysis of computation time, and describe empirical comparisons to a standard classical approach and a hybrid solver. The quantum annealer is efficient at generating mazes with difficulty tailored to individuals.

Benchmarking quantum annealing with maximum cardinality matching problems: Vert et al. aim to tackle the Maximum Cardinality Matching problem with both quantum and simulated annealing. There exist classical algorithms for this problem that can solve it efficiently, although the authors show that the problem is challenging for both simulated and quantum annealing.

Exploring performance on both original inputs and embedded inputs (after compilation onto the connection topology of the quantum annealer), the authors look at the effects of various parameters such as chain strength and annealing time on performance. These experiments are replicated for quantum annealers with both the Pegasus and Zephyr topologies, highlighting their differences.

Moreover, small-scale classical simulations of the quantum annealing process by solving the Schrodinger equation are used to present the ideal behavior of the quantum processing units. As an interesting observation, the authors note that classical algorithms exhibited worse scaling performance in the embedded version of the problems than the quantum annealer. They show that although this is a native limitation of the quantum hardware, it has less impact on quantum performance than on other solution methods.

Exploration of new chemical materials using black-box optimization with the D-Wave quantum annealer: Doi et al. address an application in screening materials using black-box optimization, which describes cases where a subset of constraints or objectives in an optimization problem cannot be fully encoded as part of the model and must be accessed by an oracle.

One way to address these problems is via Bayesian optimization, where samples are drawn iteratively from a solution space aiming to balance exploration of the black-box function and exploitation toward an optimal solution given by an objective. To balance these two goals and determine which point to sample next, an *acquisition function* is proposed such that the function maximum indicates where to draw the next sample.

The authors apply this approach to the problem of exploring a discrete space of chemical materials to find the binding of substituents to specific sites of the molecular frame as the composition of chemical materials. The objective function is considered black-box as it encodes a Density-Functional Theory (DFT) computation. The authors formulate the acquisition function as a QUBO and tackle it using quantum annealing. Their experiments show how the proposed method varies with respect to the variance of the probability distribution represented by the acquisition function, which aims to represent the exploration-exploitation trade-off.

Software techiques for training restricted Boltzmann machines on size-constrained quantum annealing hardware: Salmenperä and Nurminen study software techniques to train restricted Boltzmann machines using quantum annealers. The main focus of this work is how to make use of hardware with limited size. They describe dropout techniques that allow the annealer to be applied to a subnetwork, and an approach to processing multiple small networks in parallel on the same chip. The authors present results of empirical tests of these methods.

Pneumonia detection by binary classification: classical, quantum, and hybrid approaches for support vector machine (SVM): Guddanti et al. develop a machine learning tool that can accurately classify chest X-ray images as belonging to normal or pneumonia-infected individuals. Classical, quantum (D-Wave), and hybrid annealing methods are explored. The results of these methods are compared and contrasted.

Quantum annealing research at CMU: algorithms, hardware, applications: Tayur and Tenneti present a mini-review of quantum annealing research at Carnegie Mellon University. A highlight of this work is the study of Graver-basis-based hybrid optimization methods using quantum annealers. Other work includes developments of efficient minor-embedding strategies and development of photonic Ising machines. This work includes studies on more abstract benchmark-related problems like Max Cut, as well as real-world problems in cancer genomics.

Adiabatic quantum computing impact on transport optimization in the last-mile scenario: Sales and Araos apply a hybrid quantum-classical approach to transport optimization in the ever-evolving landscape of global trade and supply chain management. They look at the Vehicle Routing Problem (VRP) which is to find an optimal set of routes for multiple vehicles to service a given set of customers. The VRP implementation uses a 2-phase approach: first clustering (grouping the customers), and then finding the optimal routes inside each cluster. This research offers contributions to logistics optimization techniques and their potential for enhancing supply chain efficiency.

Experimenting with D-Wave quantum annealers on prime factorization problems: Ding et al. present a detailed report of experimental decisions that led to the largest prime number factorization (8,219,999 = 32,749 * 251) performed with a pure quantum computational approach reported in the literature to date (doi: 10.1038/s41598-024-53708-7). The authors describe their experiences and experimental results, to develop guidelines for best-use of D-Wave quantum annealers. They address tradeoffs arising in algorithmic questions such as minor-embedding and handling of flux biases. This report provides guidance to other practitioners about improving performance for their own applications.

ILP-based resource optimization realized by quantum annealing for optical wide-area communication networks: Witt et al. apply D-Wave quantum annealers to the problem of resource allocation within communication networks. This work focuses on developing a framework that uses neural networks to refine the solutions returned by the quantum annealer; the authors also perform a detailed study of parameter setting. They observe that improvements using the network were possible in some cases, but not in others.

We are pleased to report that this broad collection of contributions to better understanding of quantum annealing processors is now available to the research community.

Author contributions

NC: Writing – original draft, Writing – review & editing. CM: Writing – original draft, Writing – review & editing. SM: Writing – original draft, Writing – review & editing. DB: Writing – original draft, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Acknowledgments

We thank the authors for their hard work in preparing their submissions. We thank the reviewers for their careful consideration and thoughtful suggestions for improving submissions. We thank the editors and staff of *Frontiers* for their help in navigating the publication process and tools.

Conflict of interest

CM was employed by D-Wave Systems at the time this Research Topic was developed.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The author(s) declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

Publisher's note

frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 17 August 2023
DOI 10.3389/fcomp.2023.1235784

OPEN ACCESS

EDITED BY
Catherine McGeoch,
D-Wave Systems, Canada

REVIEWED BY
Henri Calandra,
Total, France
Amandeep Singh Bhatia,
Purdue University, United States
Matthias Möller,
Delft University of Technology, Netherlands
Travis S. Humble,
Oak Ridge National Laboratory (DOE),
United States

*CORRESPONDENCE
Daniel O'Malley
✉ omalled@lanl.gov

RECEIVED 06 June 2023
ACCEPTED 01 August 2023
PUBLISHED 17 August 2023

CITATION
Greer S and O'Malley D (2023) Early steps toward practical subsurface computations with quantum computing.
Front. Comput. Sci. 5:1235784.
doi: 10.3389/fcomp.2023.1235784

Early steps toward practical subsurface computations with quantum computing

Sarah Greer[1,2] and Daniel O'Malley[1]*

[1]Computational Earth Science (EES-16), Los Alamos National Laboratory, Los Alamos, NM, United States, [2]Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, United States

Quantum computing exploits quantum mechanics to perform certain computations more efficiently than classical computers. Current quantum computers have performed carefully tailored computational tasks that would be difficult or impossible for even the fastest supercomputers in the world. This "quantum supremacy" result demonstrates that quantum computing is more powerful than classical computing in some computational regimes. At present, it is unknown if any computational problems related to the Earth's subsurface fall within these regimes. Here, we describe an approach to performing seismic inverse analysis that combines a type of quantum computer called a quantum annealer with classical computing. This approach improves upon past work on applying quantum computing to the subsurface (via subsurface hydrology) in two ways. First, the seismic inverse problem enables better performance from the quantum annealer because of the Earth's relatively narrow distribution of P-wave velocities compared to the broad distribution of hydraulic conductivities. Second, we develop an iterative approach to quantum-computational inverse analysis, which works with a realistic set of observations. By contrast, the previous method used an inverse method that depended on an impractically dense set of observations. In combination, these two advances significantly narrow the gap a quantum-computational advantage for a practical subsurface geoscience problem. Closing the gap completely requires more work, but has the potential to dramatically accelerate inverse analyses for subsurface geoscience.

KEYWORDS

quantum computing, hydrology, seismic, subsurface, iterative

1. Introduction

Computation has played a critical role in subsurface geoscience for decades. It has been used to simulate ocean circulation (Pinardi et al., 1997), flow in subsurface fractures (Kosakowski and Berkowitz, 1999), mantle flow in stunning detail (Stadler et al., 2010), and has been instrumental to the modern developments in seismic imaging (Bednar, 2005), among many other examples. Computers are also widely used to estimate subsurface properties that are difficult to observe directly using inverse analysis (Khan et al., 2000; Lu and Robinson, 2006). Massive performance improvements have buoyed the widespread use of computers.

Recent computer performance trends indicate that performance improvements are diminishing, suggesting that the rising tide of improving computational performance may be coming to an end. This trend has led to increased use of novel computational methods such as graphical processing unit computing (Fatemi and Poppe, 2018) and machine learning (Gentine et al., 2018), either together or separately. Approaches such as these have the

potential to improve performance significantly, but only by a constant factor. That is, they might provide a speed-up of a factor of, e.g., 10 or 100, but the speed-up does not depend on the problem size. Quantum computing, on the other hand, opens the doors to fundamentally different algorithms that give larger and larger speed-ups as the problem becomes larger. This improved scaling behavior is crucial in subsurface geoscience. Solving equations, such as the groundwater flow equation or the seismic wave equation, might require a highly-refined mesh to resolve heterogeneity. This is because heterogeneities can be very small.

The improved scaling behaviors of certain quantum algorithms is what made possible the recent demonstration of quantum supremacy, where a quantum computer performed a calculation that would push the fastest classical supercomputers beyond their limits (Arute et al., 2019; Zhong et al., 2020; Morvan et al., 2023). While it is known that some problems related to cryptography and quantum chemistry are well-suited to quantum computers, the picture is much less clear for computational science broadly. There are efficient algorithms that could theoretically be used to solve large systems of equations (Harrow et al., 2009; Subaşı et al., 2019), but these efficiencies may be undone by the implementation details needed to use these algorithms for a particular application (Aaronson, 2015). Current work in the geosciences utilizes a more empirical approach—trying different problems and observing the performance (O'Malley, 2018; Sarkar and Levin, 2018; Greer and O'Malley, 2020; Dukalski, 2021; Henderson et al., 2021; Souza et al., 2022; Dukalski et al., 2023). Generally, the performance on current quantum computers lags behind the performance of classical computers using the best algorithms. Nonetheless, it is a critical first step to establish whether or not a quantum computer can solve a problem before trying to establish a performance advantage.

This work, which does not attempt to show any quantum advantage, explores two applications—seismic inverse analysis and hydrologic inverse analysis. It improves upon previous work by enabling the solution of more realistic problems. Our approach enables 2D seismic inverse analysis, whereas previous work focused on a 1D, layered approach (Souza et al., 2022). Past work in hydrology required the use of an unrealistic set of observations (O'Malley, 2018), whereas the approach used here can handle arbitrary, realistic sets of observations. While we study this approach in the context of hydrologic and seismic inverse analysis, it can be applied to other subsurface applications where the goal of the inverse analysis is to segment the subsurface into two separate facies. Traditionally, a seismic inverse problem of this nature could be solved using imaging methods such as reverse time migration (Baysal et al., 1983) or full waveform inversion (Virieux and Operto, 2009). Hydrologic inverse problems are usually solved using variants of the geostatistical approach such as the principal component geostatistical approach (Kitanidis and Lee, 2014) and sometimes more modern techniques leveraging machine learning are used (Kadeethum et al., 2021; Wu et al., 2023).

This work also goes a step beyond the empirical observation of the performance and identifies problem characteristics that enable better performance for the quantum computer. These insights can help guide future work to find an advantage for quantum computers in applications to the subsurface. However, since quantum annealing hardware is still in its early stages, the problems we look at are still relatively simple compared to similar problems solved using classical computing techniques. These problems are scaled according to current quantum-computing hardware's computational size, and advances in hardware will allow for more complex problems to be addressed.

If the long-term goals of this research are successful, subsurface geoscience will be able to exploit the theoretical advances that have been demonstrated for calibrating models to data (Wiebe et al., 2012). Some quantum algorithms show an exponential speed-up (e.g., Harrow et al., 2009; Wiebe et al., 2012) which could be transformational in this context. A quadratic speedup [often built upon Grover's algorithm (Grover, 1996)] is more common and could still be impactful for subsurface inverse analysis, which is often computationally expensive. These algorithms could open doors to solve subsurface geoscience problems with unprecedented resolution and accuracy. After decades of research, the improvements that can be made in classical computational methods are becoming marginal (Shalf, 2020). Novel computational architectures, of which quantum computing is arguably the most promising, remain largely unexplored and have tremendous potential. Now is the time to do this exploration.

The remainder of this manuscript is organized as follows. In Section (2), we describe quantum annealing, which is the quantum algorithm that we leverage, and our approach to formulating inverse analysis as a problem suitable for quantum annealing. Section 3 describes the results of applying this approach to seismic and hydrologic problems. A discussion of various aspects of our approach, including a problem characteristic that improves performance and possibilities with future quantum hardware and methods is presented in Section 4. Finally, concluding remarks are made in Section 5.

2. Methods

2.1. Quantum annealing

Quantum annealing is a heuristic optimization algorithm, similar to simulated annealing (Kirkpatrick et al., 1983), that seeks to find optimal solutions faster than classical methods by exploiting quantum fluctuations (Kadowaki and Nishimori, 1998). There are theoretical guarantees of convergence to the optimal state under certain conditions (Morita and Nishimori, 2008). In practice, these assumptions are generally violated. For example, with the D-Wave quantum annealers (Johnson et al., 2011) that we use here, the anneal process is often performed quickly, whereas the theoretical guarantees generally require the anneal to be performed slowly. In the language of quantum annealing, this means that the adiabaticity is violated, since the fast annealing process means the system will often leave the ground state.

The input to a D-Wave quantum annealer is a vector $\mathbf{h} = (h_i)$ and a matrix $J = (J_{ij})$. The matrix, J, is sparse with a sparsity pattern defined by the connectivity graph associated with the annealer's qubits. The size of the matrix, N, is determined by the size of the problem. For existing D-Wave quantum annealers, this is based on a so-called Chimera graph (Boothby et al., 2016).

From a practical perspective, the quantum annealer can be thought of as minimizing a function of the form

$$g(\mathbf{s}) = \sum_{i=1}^{N} h_i s_i + \sum_{i,j=1}^{N} J_{ij} s_i s_j \,, \tag{1}$$

where each spin, s_i, is either -1 of $+1$, and is called the Ising formulation. On the D-Wave annealer, there are additional sparsity constraints on the matrix J_{ij}. That is, some of the J_{ij} are constrained to be zero. Further, the D-Wave hardware limits the range of values—called the dynamic range—that can be set for h_i and J_{ij}, and problems with coefficient outside these ranges have to be rescaled to fit. This rescaling can have the effect of pushing small coefficients into the hardware's noise range. A more accurate description of the behavior of the annealer is that it is drawing from a distribution that preferentially samples values of $\mathbf{s}$ that make $f(\mathbf{s})$ small. This distribution can often be well-approximated by a Boltzmann distribution where Equation 1 defines the energy subject to a temperature or "effective temperature" that is a characteristic of the hardware. Equation 1 can be formulated as a quadratic unconstrained binary optimization (QUBO) problem,

$$g(\mathbf{q}) = \sum_{i=1}^{N} a_i q_i + \sum_{i,j=1}^{N} b_{ij} q_i q_j \tag{2}$$

where each bit, q_i, is either 0 or 1, and is related to the formulation in Equation 1 via $s_i = 2q_i - 1$. This change, from inverting values of $s_i \in \{-1, 1\}$ as in Equation 1 to $q_i \in \{0, 1\}$ as in Equation 2, is what differentiates the Ising model from the QUBO model. The values of h_i and J_{ij} can also be transformed into the values of a_i and b_{ij} by associating like terms in Equations (1) and (2). Equations (1) and (2) are two equivalent ways of formulating the same problem. We will use the formulation in Equation 2 throughout.

2.2. Inverse approach

We consider an inverse approach where the goal is to divide the subsurface into two different materials with constitutive properties based on measurements obtained on the boundary of the domain, which aligns well with the quantum annealer's ability to perform binary optimization. Using standard variable names from seismic inversion, the objective function used in the inverse analysis takes the form

$$\mathbf{F}(\mathbf{c}) = \sum_{i=1}^{N} |[\mathbf{U}(\mathbf{c})]_i - \hat{u}_i|^2 \,, \tag{3}$$

where $\mathbf{U}$ is a non-linear forward modeling operator such that $\mathbf{U}(\mathbf{c}) = \mathbf{u}$ is a solution to the relevant governing equation (wave equation for the seismic problem, groundwater flow equation for the hydrology problem), $\mathbf{c}$ is the subsurface model, $\hat{\mathbf{u}}$ is the measurements (wavefield in the case of the seismic problem and hydraulic head in the hydrology problem), and i represents the index of the N different observations. Note that each component of $\mathbf{c}$ can take only two values, either c_{low} or c_{high}, which will correspond to the low and high values of the relevant parameter field (P-wave velocity for the seismic problem, or hydraulic conductivity for the hydrology problem). We attempt to minimize this global objective function using an iterative process, where each iteration involves solving a related optimization problem that is suited to the quantum annealer. We begin with an initial guess, $\mathbf{c}^{(0)}$, and iteratively produce $\mathbf{c}^{(k+1)}$ from $\mathbf{c}^{(k)}$.

During an iteration, we find the model update by creating a quadratic unconstrained binary optimization (QUBO) problem that approximates Equation (3). This QUBO problem can then be solved with the quantum annealer. The QUBO objective function takes the form

$$\mathbf{F}(\mathbf{q}) = \sum_{n=1}^{N} |[\tilde{\mathbf{U}}(\mathbf{q})]_n - \hat{\mathbf{u}}_n|^2 \tag{4}$$

where $\tilde{\mathbf{U}}(\mathbf{q})$ is a linear approximation to $\mathbf{U}$ at the current best estimate $\mathbf{c}^{(k)}$. This estimate is given by

$$\tilde{\mathbf{U}}(\mathbf{q})^{(k)} = \mathbf{U}(\mathbf{c}^{(k)}) + \sum_{m=1}^{M} q_m \mathbf{B}_m \tag{5}$$

where $\mathbf{q}$ is a binary vector that indicates how the model should be updated, and M is the size of the model, which is the number of parameters that we estimate. The vector $\mathbf{B}_m$ is defined to be

$$\mathbf{B}_m = \begin{cases} \mathbf{U}(\mathbf{c}^{(k)} + [c_{low} - c_{high}]\mathbf{e}_i) - \mathbf{U}(\mathbf{c}^{(k)}) & \text{if } c_i^{(k)} = c_{high} \\ \mathbf{U}(\mathbf{c}^{(k)} + [c_{high} - c_{low}]\mathbf{e}_i) - \mathbf{U}(\mathbf{c}^{(k)}) & \text{if } c_i^{(k)} = c_{low} \end{cases} \tag{6}$$

where $\mathbf{e}_i$ is the i^{th} standard basis vector of size M consisting of all zeros except for a 1 in the i^{th} component. Essentially, $\mathbf{B}_m$ is an operator that flips the value of its input and then forward models it. This makes the computational cost of each iteration equal to the cost of M forward model runs. When updating the model, if $q_i = 0$ then $c_i^{(k+1)}$ takes the same value as $c_i^{(k)}$. On the other hand, if $q_i = 1$ then $c_i^{(k+1)}$ takes the same opposite value as $c_i^{(k)}$. That is, if $q_i = 1$ then

$$c_i^{(k+1)} = \begin{cases} c_{high} & \text{if } c_i^{(k)} = c_{high} \text{ and } q_i = 0 \\ c_{low} & \text{if } c_i^{(k)} = c_{low} \text{ and } q_i = 0 \\ c_{low} & \text{if } c_i^{(k)} = c_{high} \text{ and } q_i = 1 \\ c_{high} & \text{if } c_i^{(k)} = c_{low} \text{ and } q_i = 1 \end{cases} \tag{7}$$

The least-squares objective function for iteration k is then equivalent to

$$\mathbf{F}(\mathbf{q}) = ||\mathbf{U}(\mathbf{c}^{(k)}) - \hat{\mathbf{u}}||_2^2 + 2(\mathbf{U}(\mathbf{c}^{(k)}) - \hat{\mathbf{u}})^T \mathbf{B}^T \mathbf{q} + \mathbf{q}^T \mathbf{B}\mathbf{B}^T \mathbf{q} \,. \tag{8}$$

Note that this equation has a constant term, a term that is linear in $\mathbf{q}$, and a term that is quadratic in $\mathbf{q}$. The constant term is irrelevant to the optimization process, and neglecting it results in a function of the form in Equation (2). After creating the QUBO for a given iteration (Equation 8), Los Alamos National Laboratory's D-Wave 2000Q quantum annealer does the (forward) annealing and returns 1,000 possible solutions. We analyze the first several solutions that minimize the local objective function (Equation 8), select the update among those that minimizes the global objective function (3), and update the model accordingly. The selected update, $\mathbf{q} \in \{0, 1\}^M$, is used to update the model, $\mathbf{c}^{(k+1)}$, using Equation (7). We used default values for the annealing time, thermalization time, and post-processing. D-Wave's heuristic embedder was used to embed the problem graph on the D-Wave, which generally involves embedding a complete graph.

2.2.1. Hydrologic inverse analysis

We study the steady-state groundwater flow equation,

$$\nabla \cdot \left(k(\mathbf{x})\nabla h\right) = f \tag{9}$$

where h represents the hydraulic head, $k(\mathbf{x})$ denotes the heterogeneous hydraulic conductivity, and f represents fluid sources or sinks. Note that throughout, we will assume that $f = 0$ (i.e., there are no fluid sources/sinks) and that $k(\mathbf{x})$ is either k_h or k_l where k_h is a high conductivity value and k_l is a low conductivity value. The inverse analysis's goal is, given a set of hydraulic head observations, to infer the spatially variable hydraulic conductivity, $k(\mathbf{x})$. That is, to determine at each location, $\mathbf{x}$, whether $k(\mathbf{x}) = k_h$ or $k(\mathbf{x}) = k_l$. This process corresponds to determining where two different materials exist in the aquifer. If the highly conductive material were sand and the low conductivity material were clay, the inverse analysis would answer the questions: where is the sand? and where is the clay?

To obtain a hydrologic inverse problem, we generate two hydraulic conductivity fields based on two real-world examples where the hydraulic conductivity at each location is either k_h or k_l. We first look at the example from Lu and Robinson (2006), which includes two low-permeability zoned embedded in a high permeability background medium. Our second hydrology example includes a slurry wall in our domain of interest. The hydraulic conductivities are distributed on a grid that is coarse compared to the finite volume grid on which the Equation (9) is solved. This use of these two grids reduces the number of hydraulic conductivity variables so that they can all be fit on the quantum annealer and the finite volume grid remains sufficiently resolved to produce a physically accurate simulation. Given these hydraulic conductivities, Equation (9) is solved to obtain a set of hydraulic head observations. These observations and the current estimate of the hydraulic conductivities are then used to formulate a QUBO using the previously discussed approach. Effectively, the PDE solver used to solve Equation (9) provides the function $\mathbf{F}(\mathbf{c})$ in Equation (3). After approximations, this results in the QUBO given in Equation (8). The quantum annealer is then used to optimize the QUBO, providing an updated estimate of the hydraulic conductivities. This iterative process continues until the convergence criteria are satisfied.

We have a 700 × 700 meter domain in this application, with seven discrete permeability blocks in the x-direction and seven in the y-direction. We place 24 receivers across the surface of the domain in a checkerboard pattern. Our computational mesh grid is $dx = dy = 10$ meters. Our goal is to invert for the locations of the two different facies, which each have permeabilities of k_l or k_h. Given this problem geometry, there are $2^{49} \approx 5.6 \times 10^{14}$ possible solutions. The exact model we use is shown in Figure 1.

2.2.2. Seismic inverse analysis

In this application, our goal is to find the distribution of P-wave velocity values that give rise to a set of wavefield measurements from receivers on the surface of the domain. We study the acoustic wave equation,

$$\left(\frac{1}{c(\mathbf{x})^2}\frac{\partial^2}{\partial t^2} - \nabla\right) u = f\,, \tag{10}$$

where $c(\mathbf{x})$ is the P-wave velocity, u is the measured wavefield, and f is the forcing, or source term. Similar to the hydrology example, we assume that $c(\mathbf{x})$ is either c_{high} or c_{low}, so this problem can be thought of as locating two different materials in the subsurface. We look at two different examples: one is a salt body in an constant

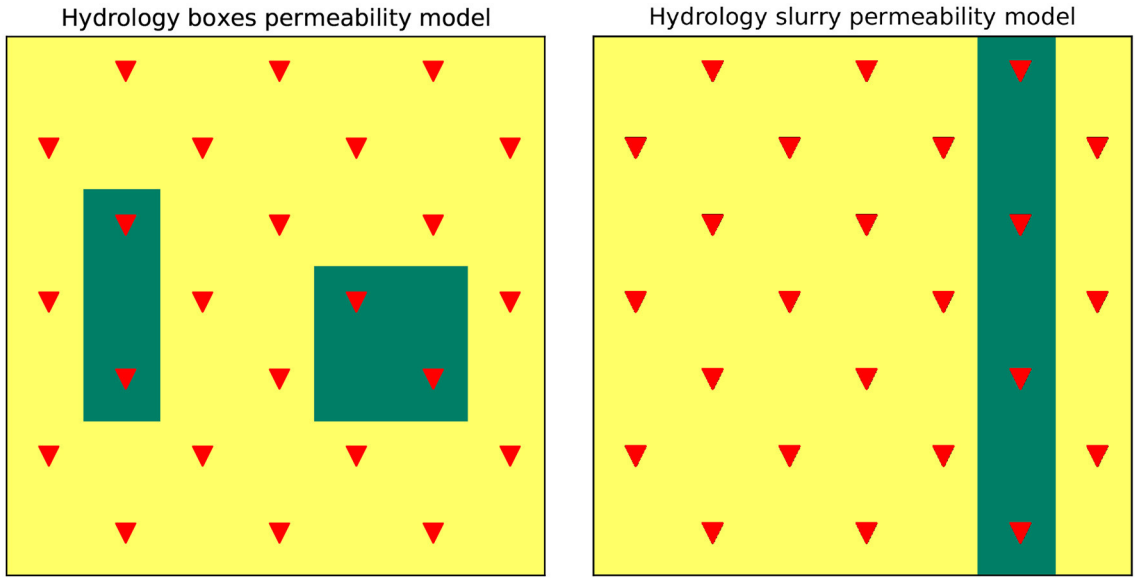


FIGURE 1
The two hydrology permeability models used in this experiment, where yellow locations are high permeability and green locations are for low permeability values. We place seven discrete permeability block locations in the x-direction and seven in the y-direction. The domain is 700 × 700 m, so each discrete permeability block is 100 × 100 m. The computational mesh grid is dx = dy = 10 meters. We have 24 receivers, denoted by red triangles, which are spread in a checkerboard pattern across the top of the domain.

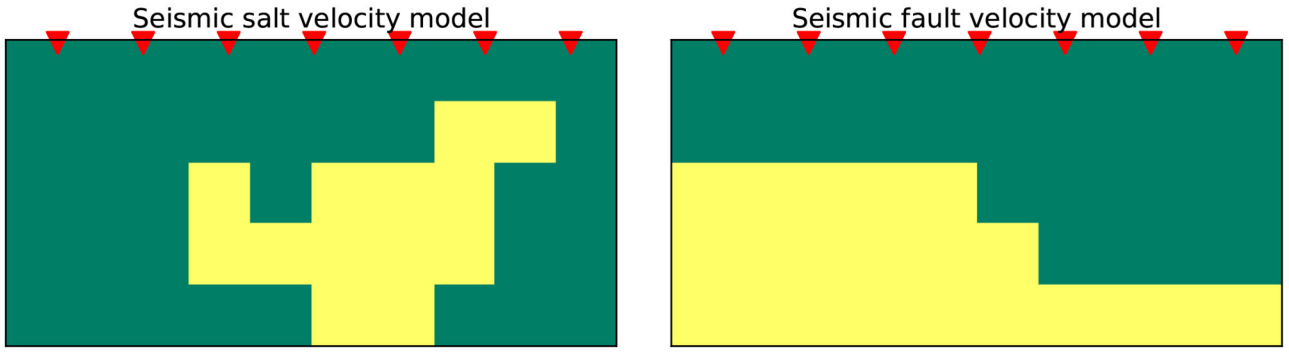


FIGURE 2
The seismic velocity models used in this experiment, where yellow locations are high velocity and green locations are for low velocity values. We place ten discrete velocity block locations in the x-direction and five in the z-direction. The domain is 1,000 m wide and 500 m deep, so each discrete velocity block is 100 × 100 m. The computational mesh grid is $dx = dz = 10$ meters. We have seven receivers, denoted by red triangles, which are spread evenly across the top of the domain. The source is located at the top center of the domain.

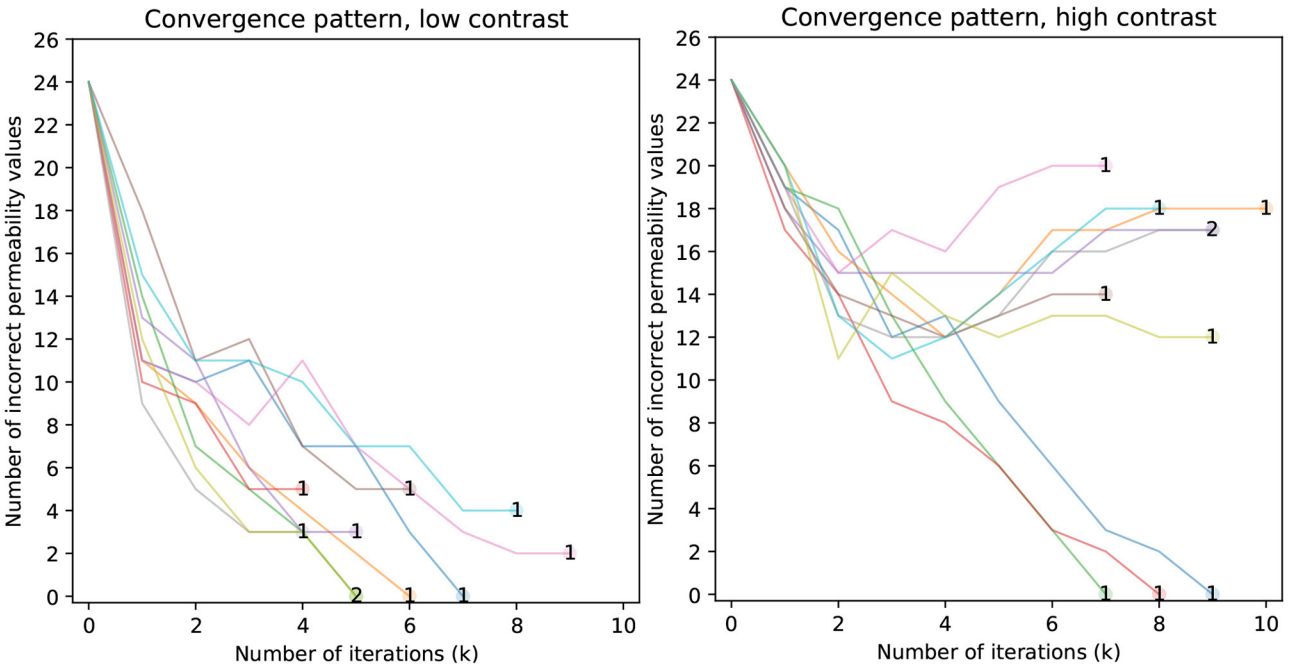


FIGURE 3
The convergence of our quantum annealing inverse approach for 10 low **(left)** and high **(right)** contrast hydrologic inverse problems is shown using the same initial value of $c_0 = k_{low}$. The numbers at the end convergence point represent the number of model runs that converged to that model value. The approach works well in the low contrast regime, but there are problems in the high contrast regime due to the quantum annealer's limited dynamic range. Realistic hydrologic inverse problems have a high contrast, so this is problematic.

background medium, and the other is a two-layer faulted example. In this application, our domain of interest is 1 kilometer wide and 0.5 km deep and includes 50 possible velocity value locations: 5 in the vertical direction and 10 in the horizontal direction. To keep computational costs similar to that of the hydrologic inverse analysis, our experiment only uses one source, and we choose our source f to be a 25 Hz Ricker wavelet at the top center of the domain. We also spread seven receivers evenly across the surface of the domain to record the wavefield measurements. We use a 1 millisecond sampling interval and record for 0.4 s at all sensor locations for the wavefield measurements. We use the observations recorded at the receiver locations to formulate a QUBO using the method discussed in this paper. As in the hydrology problem, the PDE solver used to solve Equation (10) provides the function $\mathbf{F}(\mathbf{c})$ in Equation (3). After approximations, this results in the QUBO given in Equation (8). The exact velocity model we use is in Figure 2. Given this problem geometry, there are $2^{50} \approx 1.1 \times 10^{15}$ possible solutions.

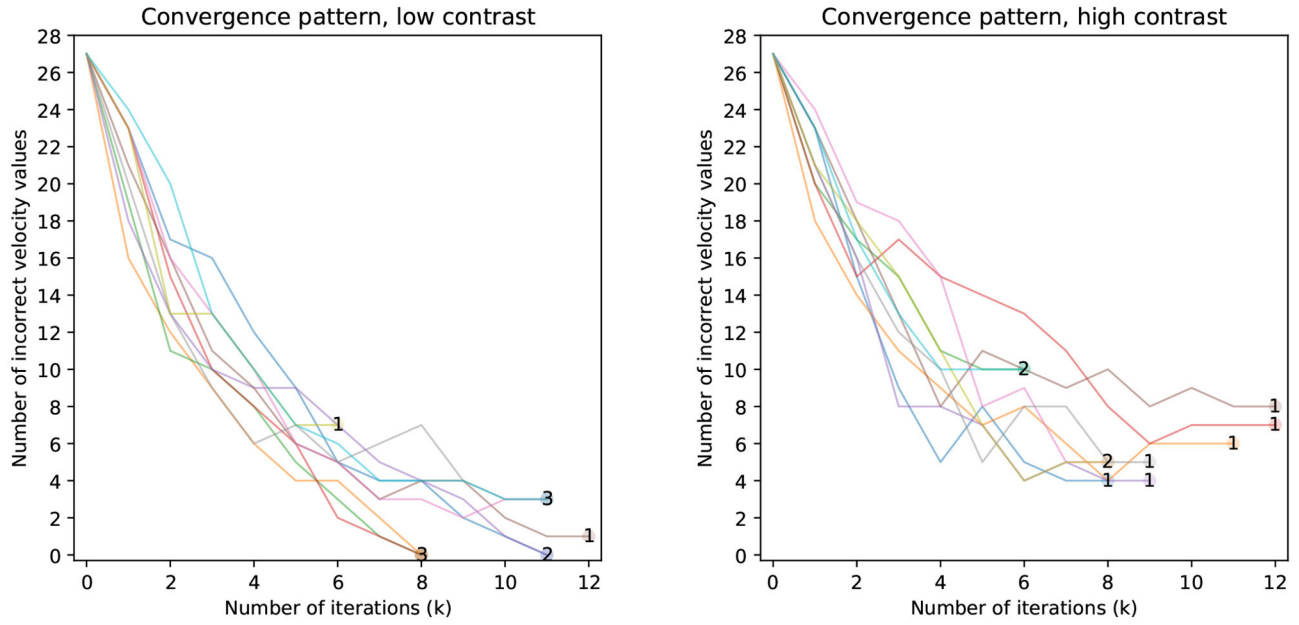


FIGURE 4
The convergence of our quantum annealing inverse approach for 10 low **(left)** and high **(right)** contrast seismic inverse problems is shown using the same initial value of $c_0 = c_{low}$. The numbers at the end convergence point represent the number of model runs that converged to that model value. The approach works well in the low contrast regime, but there are problems in the high contrast regime due to the quantum annealer's limited dynamic range. Realistic seismic inverse problems have a low contrast, so these problems are well-suited to the quantum annealer.

3. Results

We applied the approach to inverse analysis previously discussed to seismic and hydrologic problems. For each of these physical problems, we consider a case where c_{high}/c_{low} is large and another where c_{high}/c_{low} is relatively small. For the seismic problem, the case where c_{high}/c_{low} is relatively small is realistic. On the other hand, the case where c_{high}/c_{low} is large is realistic for the hydrologic problem.

Figure 3 shows the convergence behavior of the inverse analyses for the hydrologic problem. The left panel shows the convergence pattern when there is low contrast between the high hydraulic conductivity and low conductivity, while the right panel shows the convergence pattern when there is a large contrast. For the low contrast case, we use $k_{low} = 1 \times 10^{-3} m/s$ and $k_{high} = 2 \times 10^{-3} m/s$, and for the high contrast case, we use $k_{low} = 5 \times 10^{-8} m/s$ and $k_{high} = 5 \times 10^{-3} m/s$. In both cases, the initial model $c_0 = k_{low}$. The stopping criteria for iterations is when the same model output was selected for two iterations in a row. There is a strong tendency to converge to a good result when the contrast is low. In the low contrast setting, the inverse approach gets all the hydraulic conductivities correct in four analyses and at most 5 incorrect in the remaining six analyses. On the other hand, there is a strong tendency to converge to a lackluster result in the high contrast case. In the high contrast setting, the inverse approach gets all the hydraulic conductivities correct three times, but gets 12 or more incorrect in the remaining seven analyses. The performance in these two settings indicates that the high contrast case is more challenging for the inverse method than the low contrast case.

TABLE 1 Variability in permeability in unconsolidated sediments (Fetter, 2018).

Materials	Hydraulic conductivity (m/s)
Clay	10^{-11}–10^{-8}
Silt, sandy silts, clayey sands, till	10^{-8}–10^{-6}
Silty sands, fine sands	10^{-7}–10^{-5}
Well-sorted sands, glacial outwash	10^{-5}–10^{-3}
Well-sorted gravel	10^{-2}–10^{-4}

Note the extreme variability between sediment types (nine orders of magnitude) and the variability within sediment types (typically two orders of magnitude).

Figure 4 shows the convergence behavior of the inverse analyses for the seismic problem. The left panel shows the convergence pattern when there is a low contrast between the high velocity and low velocity, while the right panel shows the convergence pattern when there is a large contrast.

For the low contrast case, we use the same velocity values and contrast as used in the initial example: $c_{low} = 4,250 m/s$ and $c_{high} = 4,750 m/s$. In the high contrast case, we use $c_{low} = 2,000 m/s$ and $c_{high} = 5,000 m/s$. In both cases, the initial model $c_0 = c_{low}$. The stopping criteria for iterations is when the same model output was selected for two iterations in a row. Like the hydrologic inverse analysis, the low contrast problem shows better convergence behavior than the high contrast problem. In the low contrast setting, the inverse approach gets all the velocities correct five times, between one and four velocities incorrect four times,

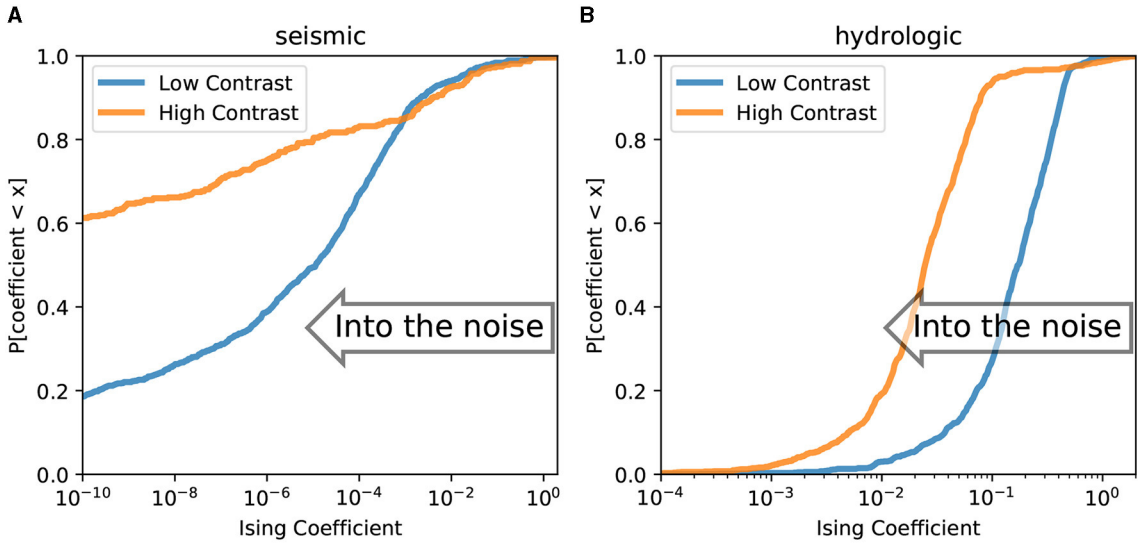


FIGURE 5
The cumulative distribution function for an Ising problem in **(A)** the seismic inverse analysis and **(B)** the hydrologic inverse analysis are shown. Note that the high contrast problem is more likely to have small coefficients that get lost in the noise of the quantum annealer.

and seven velocities incorrect once. Again, the performance of the inverse approach declines as the contrast increases. In the high contrast setting, the inverse analysis never obtains all the velocities correctly. The number of incorrect velocity values tends to cluster around seven, which was the worst result in the low contrast case. Due to the inconsistent nature of the convergence pattern in low vs. high contrast cases and seismic vs. hydrologic examples, we choose to not provide an algebraic form for the convergence patterns.

4. Discussion

Inverse analysis in subsurface flow problems is challenging for a variety of reasons. One source of challenges is the high variability in hydraulic conductivity and permeability found in the Earth's subsurface. Table 1 shows hydraulic conductivity ranges for a set of unconsolidated sediments (Fetter, 2018). Note that even within one class of materials, the hydraulic conductivity can vary by several orders of magnitude. Since different materials often coexist in the same region of the subsurface, the variability can be even larger than this. For example, the widely used SPE 10 model exhibits variation in the permeability over 8–12 orders of magnitude from 10^{-7} milliDarcy to 10^{4} milliDarcy (Lie, 2019).

This extreme variability adds to the challenges for the quantum annealer because high contrasts in the parameters result in more variability in the QUBO coefficients. Figure 5 shows the distribution of the coefficients in one iteration of the seismic and hydrologic inverse analysis for both the low and high contrast cases. The coefficients that are small in magnitude tend to be lost in the noise associated with the quantum annealer (Golden and O'Malley, 2021). This is caused by the rescaling that is necessary to fit the coefficients within the dynamic range allowed by the quantum annealer. The quantum annealer has little or no information about qubits whose linear and quadratic coefficients are below the hardware's analog noise level. Since the iterative model updates are determined by solving this QUBO problem, which may be inaccurate in high contrast cases due to its coefficients being below the threshold of hardware noise, the model updates selected from this method may diverge from minimizing the original problem's objective function, as seen in Figure 3.

While different geologic units may have large contrasts in hydraulic conductivity, there is much less variability in P-wave velocity values both between and within rock types than in hydraulic conductivity, as seen in Table 2. In general, the P-wave velocity of a given unit will increase with depth since the material becomes more compact. Because of this, units close to each other in depth are likely to have a more similar velocity than units far from each other in depth. This allows for a lower impedance contrast under the assumption of a constant-density acoustic model. The high-contrast case, where the performance is not as good, would be uncommon in real examples of subsurface velocity models with the notable exception in areas with salt bodies, such as in the Gulf of Mexico. Because of the difficulties with QUBO coefficients being lost in the noise, quantum annealing appears more suitable for seismic inverse analysis than hydrologic inverse analysis using the method we propose.

One of the most notable limitations of the current work is that the resolution of the subsurface parameterization is limited. The hydrologic inverse problem had a 7×7 grid of hydraulic conductivities, and the seismic problem had a 5×10 grid of P-wave velocities. This work was done with D-Wave's older generation 2000Q hardware. D-Wave's current generation Advantage hardware uses a Pegasus graph (Dattani et al., 2019). This hardware significantly increases the number of qubits on the quantum annealer chip and the number of connections per qubit. The effect of this will be to approximately triple the number of parameters that can be calibrated. Approximately tripling the number of parameters will significantly increase the resolution of

TABLE 2 Variability in P-wave velocity in common rock types in exploration seismology (Bourbié et al., 1987).

Materials	P-wave velocity (m/s)
Wet sands	1,500 – 2,000
Saturated shales and clays	1,100 – 2,500
Porous and saturated sandstones	2,000 – 3,500
Limestones	3,500 – 6,000
Salt	4,500 – 5,500

Note that there is much less variability in P-wave velocity values both between and within rock types than in Hydraulic Conductivity of unconsolidated sediments in Table 1.

the inverse model that the quantum annealer can handle. The Advantage hardware also reduces the noise on the system, which could improve the performance of high contrast problems.

It should also be noted that further methodological developments could improve the resolution of the inverse model. The full domain could be explored by moving through the domain in a tiling fashion. Another possibility would be to use an alternative to Equation (4) that has some natural sparsity. For example, parameters associated with regions that are physically distant from each other might tend to have a small quadratic term, which could be neglected in some cases. Many possibilities cannot be explored here – this is just the beginning of using quantum computing for subsurface applications.

5. Conclusion

We have considered the application of noisy, intermediate-scale quantum computing to subsurface geoscience. In particular, we have used a quantum annealer to solve seismic and hydrologic inverse problems. We found that the seismic inverse problem is better suited to the quantum annealer than the hydrologic inverse problem. This is because the ratio between a fast P-wave velocity and a slow P-wave velocity is small compared to the ratio between a high hydraulic conductivity and a low hydraulic conductivity. This ratio ultimately influences the variability of the coefficients in the Hamiltonian used to program the quantum annealer, with a large ratio resulting in higher variability. High variability in the Hamiltonian coefficients leads to poor performance because the small coefficients effectively get lost in the noise.

In addition to identifying a subsurface problem that is well-suited to the quantum annealer, we also developed methods that enable the quantum annealer to solve inverse problems with a realistic set of observations. This is a significant step forward because previous work, which focused on the hydrologic inverse problem, was limited to an unrealistic set of observations. In particular, it required that the hydraulic head be observed at every point on the computational grid. This was consistent with early methods that were used in computational hydrology—called direct inverse methods (Yeh, 1986). The transition to an iterative approach for inverse analysis with quantum annealing brings it in line with modern methods for inverse analysis that also use iterative methods and can handle realistic observation sets.

By transitioning from hydrology to seismology and from a direct inverse method to an iterative inverse method, we have taken two significant steps toward enabling the use of quantum annealing for practical applications in subsurface geoscience. One significant hurdle remains, and that is increasing the resolution of the subsurface image that the quantum annealer can handle. This would be aided by adding additional qubits to the quantum annealer and increasing the qubits' connectivity, both anticipated in D-Wave's next quantum annealer.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: Greer (2020).

Author contributions

DO'M conceived of the presented idea, with contributions from SG. SG implemented the method and performed the computations with support from DO'M. All authors discussed the results and contributed to the final manuscript. All authors contributed to the article and approved the submitted version.

Funding

SG acknowledges support from the United States Department of Energy through the Computational Science Graduate Fellowship (DOE CSGF) under grant number DE-SC0019323. DO'M acknowledges support from the National Nuclear Security Administration's Advanced Simulation and Computing program.

Acknowledgments

We would like to acknowledge John Golden for useful discussions.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Aaronson, S. (2015). Read the fine print. *Nat. Phys.* 11, 291–293. doi: 10.1038/nphys3272

Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., et al. (2019). Quantum supremacy using a programmable superconducting processor. *Nature* 574, 505–510. doi: 10.1038/s41586-019-1666-5

Baysal, E., Kosloff, D. D., and Sherwood, J. W. (1983). Reverse time migration. *Geophysics* 48, 1514–1524. doi: 10.1190/1.1441434

Bednar, J. B. (2005). A brief history of seismic migration. *Geophysics* 70, 3MJ–20MJ. doi: 10.1190/1.1926579

Boothby, T., King, A. D., and Roy, A. (2016). Fast clique minor generation in chimera qubit connectivity graphs. *Quant. Inf. Process.* 15, 495–508. doi: 10.1007/s11128-015-1150-6

Bourbié, T., Coussy, O., and Zinszner, B. (1987). *Acoustics of Porous Media*. Editions TECHNIP.

Dattani, N.S., Szalay, S., and Chancellor, N. (2019). Pegasus: The second connectivity graph for large-scale quantum annealing hardware. *arXiv [Preprint]*. arXiv: 1901.07636.

Dukalski, M. (2021). "Toward an application of quantum computing in geophysics," in *Fifth EAGE Workshop on High Performance Computing for Upstream*, Vol. 2021 (European Association of Geoscientists & Engineers), 1–5.

Dukalski, M., Rovetta, D., van der Linde, S., Möller, M., Neumann, N., and Phillipson, F. (2023). Quantum computer-assisted global optimization in geophysics illustrated with stack-power maximization for refraction residual statics estimation. *Geophysics* 88, V75–V91. doi: 10.1190/geo2022-0253.1

Fatemi, S., and Poppe, A. R. (2018). Solar wind plasma interaction with asteroid 16 psyche: implication for formation theories. *Geophys. Res. Lett.* 45, 39–48. doi: 10.1002/2017GL073980

Fetter, C. W. (2018). *Applied Hydrogeology*. Long Grove, IL: Waveland Press.

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., and Yacalis, G. (2018). Could machine learning break the convection parameterization deadlock? *Geophys. Res. Lett.* 45, 5742–5751. doi: 10.1029/2018GL078202

Golden, J. K., and O'Malley, D. (2021). Pre- and post-processing in quantum-computational hydrologic inverse analysis. *Quantum Inf. Process* 20, 176. doi: 10.1007/s11128-021-03115-y

Greer, S. (2020). *sygreer/QuantumAnnealingInversion.jl: First release of QuantumAnnealingInversion.jl (v1.0.0)*. Zenodo. Available online at: https://zenodo.org/record/4313834

Greer, S., and O'Malley, D. (2020). "An approach to seismic inversion with quantum annealing," in *SEG Technical Program Expanded Abstracts (SEG)*, 2845–2849. doi: 10.1190/segam2020-3424413.1

Grover, L. K. (1996). "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery), 212–219.

Harrow, A. W., Hassidim, A., and Lloyd, S. (2009). Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* 103, 150502. doi: 10.1103/PhysRevLett.103.150502

Henderson, J. M., O'Malley, D., and Viswanathan, H. S. (2021). "Interrogating the performance of quantum annealing for the solution of steady-state subsurface flow," in *2021 IEEE High Performance Extreme Computing Conference (HPEC)* (IEEE), 1–6.

Johnson, M. W., Amin, M. H., Gildert, S., Lanting, T., Hamze, F., Dickson, N., et al. (2011). Quantum annealing with manufactured spins. *Nature* 473, 194–198. doi: 10.1038/nature10012

Kadeethum, T., O'Malley, D., Fuhg, J. N., Choi, Y., Lee, J., Viswanathan, H. S., et al. (2021). A framework for data-driven solution and parameter estimation of pdes using conditional generative adversarial networks. *Nat. Comp. Sci.* 1, 819–829. doi: 10.1038/s43588-021-00171-3

Kadowaki, T., and Nishimori, H. (1998). Quantum annealing in the transverse ising model. *Phys. Rev. E* 58, 5355. doi: 10.1103/PhysRevE.58.5355

Khan, A., Mosegaard, K., and Rasmussen, K. L. (2000). A new seismic velocity model for the moon from a monte carlo inversion of the apollo lunar seismic data. *Geophys. Res. Lett.* 27, 1591–1594. doi: 10.1029/1999GL008452

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science* 220, 671–680. doi: 10.1126/science.220.4598.671

Kitanidis, P. K., and Lee, J. (2014). Principal component geostatistical approach for large-dimensional inverse problems. *Water Resour. Res.* 50, 5428–5443. doi: 10.1002/2013WR014630

Kosakowski, G., and Berkowitz, B. (1999). Flow pattern variability in natural fracture intersections. *Geophys. Res. Lett.* 26, 1765–1768. doi: 10.1029/1999GL900344

Lie, K. A. (2019). *An Introduction to Reservoir Simulation Using MATLAB/GNU Octave: User Guide for the MATLAB Reservoir Simulation Toolbox (MRST)*. Cambridge: Cambridge University Press.

Lu, Z., and Robinson, B. A. (2006). Parameter identification using the level set method. *Geophys. Res. Lett.* 33, L06404. doi: 10.1029/2005GL025541.1

Morita, S., and Nishimori, H. (2008). Mathematical foundation of quantum annealing. *J. Math. Phys.* 49, 125210. doi: 10.1063/1.2995837

Morvan, A., Villalonga, B., Mi, X., Mandrá, S., Bengtsson, A., Klimov, P. V., et al. (2023). Phase transition in random circuit sampling. *arXiv [Preprint]*. arXiv: 2304.11119. doi: 10.48550/arXiv.2304.11119

O'Malley, D. (2018). An approach to quantum-computational hydrologic inverse analysis. *Sci. Rep.* 8, 1–9. doi: 10.1038/s41598-018-25206-0

Pinardi, N., Korres, G., Lascaratos, A., Roussenov, V., and Stanev, E. (1997). Numerical simulation of the interannual variability of the mediterranean sea upper ocean circulation. *Geophys. Res. Lett.* 24, 425–428. doi: 10.1029/96GL03952

Sarkar, R., and Levin, S. A. (2018). "Snell tomography for net-to-gross estimation using quantum annealing," in *SEG International Exposition and Annual Meeting* (SEG).

Shalf, J. (2020). The future of computing beyond moore's law. *Philos. Transact. R. Soc.* 378, 20190061. doi: 10.1098/rsta.2019.0061

Souza, A. M., Martins, E. O., Roditi, I., Sá, N., Sarthour, R. S., and Oliveira, I. S. (2022). An application of quantum annealing computing to seismic inversion. *Front. Phys.* 9, 748285. doi: 10.3389/fphy.2021.748285

Stadler, G., Gurnis, M., Burstedde, C., Wilcox, L. C., Alisic, L., and Ghattas, O. (2010). The dynamics of plate tectonics and mantle flow: from local to global scales. *Science* 329, 1033–1038. doi: 10.1126/science.1191223

Subaçı, Y., Somma, R. D., and Orsucci, D. (2019). Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing. *Phys. Rev. Lett.* 122, 060504. doi: 10.1103/PhysRevLett.122.060504

Virieux, J., and Operto, S. (2009). An overview of full-waveform inversion in exploration geophysics. *Geophysics* 74, WCC1–WCC26. doi: 10.1190/1.3238367

Wiebe, N., Braun, D., and Lloyd, S. (2012). Quantum algorithm for data fitting. *Phys. Rev. Lett.* 109, 050505. doi: 10.1103/PhysRevLett.109.050505

Wu, H., Greer, S. Y., and O'Malley, D. (2023). Physics-embedded inverse analysis with algorithmic differentiation for the earth?s subsurface. *Sci. Rep.* 13, 718. doi: 10.1038/s41598-022-26898-1

Yeh, W. W.-G. (1986). Review of parameter identification procedures in groundwater hydrology: the inverse problem. *Water Resour. Res.* 22, 95–108. doi: 10.1029/WR022i002p00095

Zhong, H.-S., Wang, H., Deng, Y.-H., Chen, M.-C., Peng, L.-C., Luo, Y.-H., et al. (2020). Quantum computational advantage using photons. *Science* 370, 1460–1463. doi: 10.1126/science.abe8770

frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 15 September 2023
DOI 10.3389/fcomp.2023.1238988

OPEN ACCESS

EDITED BY
Susan Mniszewski,
Los Alamos National Laboratory (DOE),
United States

REVIEWED BY
Jaka Vodeb,
Helmholtz Association of German Research Centres (HZ), Germany
Alejandro Lopez,
Los Alamos National Laboratory (DOE),
United States
Andrea Delgado,
Oak Ridge National Laboratory (DOE),
United States

*CORRESPONDENCE
Andrew D. King
✉ aking@dwavesys.com

RECEIVED 12 June 2023
ACCEPTED 17 August 2023
PUBLISHED 15 September 2023

CITATION
Chern K, Boothby K, Raymond J, Farré P and King AD (2023) Tutorial: calibration refinement in quantum annealing.
Front. Comput. Sci. 5:1238988.
doi: 10.3389/fcomp.2023.1238988

Tutorial: calibration refinement in quantum annealing

Kevin Chern, Kelly Boothby, Jack Raymond, Pau Farré and Andrew D. King*

D-Wave, Burnaby, BC, Canada

Quantum annealing has emerged as a powerful platform for simulating and optimizing classical and quantum Ising models. Quantum annealers, like other quantum and/or analog computing devices, are susceptible to non-idealities including crosstalk, device variation, and environmental noise. Compensating for these effects through calibration refinement or "shimming" can significantly improve performance but often relies on *ad-hoc* methods that exploit symmetries in both the problem being solved and the quantum annealer itself. In this tutorial, we attempt to demystify these methods. We introduce methods for finding exploitable symmetries in Ising models and discuss how to use these symmetries to suppress unwanted bias. We work through several examples of increasing complexity and provide complete Python code. We include automated methods for two important tasks: finding copies of small subgraphs in the qubit connectivity graph and automatically finding symmetries of an Ising model via generalized graph automorphism. We conclude the tutorial by surveying additional methods, providing practical implementation tips, and discussing limitations and remedies of the calibration procedure. Code is available at: https://github.com/dwavesystems/shimming-tutorial.

KEYWORDS

quantum computing, quantum annealing, D-Wave, calibration, quadratic unconstrained binary optimization, Ising

1. Background

1.1. Introduction to quantum annealing

Quantum annealing (QA; Kadowaki and Nishimori, 1998; Johnson et al., 2011) is a computing approach that physically realizes a system of Ising spins in a transverse magnetic field. A common application of QA is to find low-energy spin states of the Ising problem Hamiltonian as follows:

$$\mathcal{H}_P = \sum_i h_i \sigma_i^z + \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z. \tag{1}$$

Here, $\{\sigma_i^z\}_{i=1}^N \in \{-1, 1\}^N$ is a set of Pauli z-operators, which can be thought of as a vector of classical ± 1 Ising spins; h_i denotes a longitudinal field (bias) on spin i, and J_{ij} (used interchangeably with $J_{i,j}$ depending on context) denotes a coupling (quadratic interaction) between spins i and j. Minimizing $\mathcal{H}_P$ is intractable, i.e., NP-hard (Barahona, 1982).

QA adds to $\mathcal{H}_P$ to an initial driving Hamiltonian as follows:

$$\mathcal{H}_D = -\sum_i \sigma_i^x. \tag{2}$$

The ground state of $\mathcal{H}_D$, which is a uniform quantum superposition of all classical states, is easy to prepare. QA guides a time-dependent Hamiltonian $\mathcal{H}(s)$ from $\mathcal{H}_D$ to $\mathcal{H}_P$ by linearly combining $\mathcal{H}_D$ and $\mathcal{H}_P$ as follows:

$$\mathcal{H}(s) = \Gamma(s)\mathcal{H}_D + \mathcal{J}(s)\mathcal{H}_P, \tag{3}$$

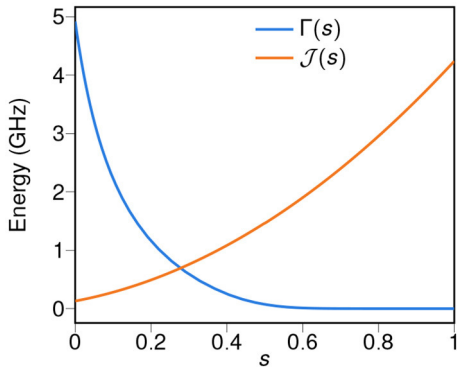


FIGURE 1
Annealing schedule for Hamiltonian (3) in a D-Wave™ Advantage™ processor. $\Gamma(s)$ and $\mathcal{J}(s)$ control the magnitude of quantum fluctuations and the Ising energy scale, respectively. These values vary from one processor to another.

where s is a unitless annealing parameter ranging from 0 to 1. Unless stated, s is simply t/t_a: time normalized by annealing time. The functions $\Gamma(s)$ and $\mathcal{J}(s)$ define the *annealing schedule*: $\Gamma(s)$ decreases toward 0 as a function of s, and $\mathcal{J}(s)$ increases as a function of s; $\Gamma(0) \gg \mathcal{J}(0)$. Units are GHz, convertible to Joules by multiplication by $\hbar$ (reduced Planck constant). An example is shown in Figure 1.

1.2. Calibration imperfections and refinement

Quantum processing units (QPUs, in this case quantum annealers) are typically made available with a single one-size-fits-all calibration. Non-idealities in the calibration can arise from a number of sources. For example, small fluctuations in the magnetic environment can bias qubits in one direction or the other. Moreover, *crosstalk*, in which a Hamiltonian term, e.g., a programmed coupler J_{ij}, can cause an undesired perturbation in another Hamiltonian term corresponding to a physically nearby device, e.g., a bias field h_i.

In short, no calibration is perfect. Oftentimes, in-depth studies of a single system (Ising model) or ensemble of systems (e.g., a set of realizations of a spin-glass model) can be improved by suppressing crosstalk and other non-idealities. This is achieved by "shimming:" inferring statistical features of an ideal annealer and tuning the Hamiltonian to produce these features. An ideal annealer, in this study, is defined simply as one that respects symmetries in the Hamiltonian—each qubit behaves identically and each coupler behaves identically.

Variations on the methods described herein have been used in many studies (King et al., 2018, 2021a,b,c, 2022, 2023; Kairys et al., 2020; Nishimura et al., 2020). Often, when behavior of the system relies on precise maintenance of energy degeneracy between states, or energy splitting from the transverse field, the results are highly sensitive to these tunings. Particularly for the simulation of exotic magnetic phases, calibration refinement is an essential ingredient of successful experiments. However, so far the discussion of these methods has mostly been relegated to Supplementary material. Here, our aim is to provide an accessible guide that will encourage the use of these powerful but simple methods.

Specific visual demonstrations of the benefit of these methods "in the wild" include:

- Frustrated 2D lattice, King et al. (2018), Extended Data Figure 7.
- Diluted ferromagnet, Nishimura et al. (2020), Figures 34–35.
- 1D quantum Ising chain, King et al. (2022), Supplementary Figures S3, S4.
- 3D quantum spin glasses, King et al. (2023), Supplementary Figures S8–S9.

The tutorial is organized as follows. In the remainder of this section, we introduce concepts that form the bases of the QPU calibration procedure. In Section 2, we illustrate the essence of our approach through a toy example. In Section 3, we extend the method and improve calibration efficiency by exploiting symmetries in a given model. In Section 4, we introduce a non-trivial system to demonstrate additional concepts useful for realistic applications. Collectively, these sections provide a comprehensive walkthrough of the calibration procedure. In Section 5, we survey additional methods for narrower use cases of the QPU. Finally, we provide practical tips and considerations in Section 6 and conclude the tutorial in Section 7.

1.3. Inferring statistical features: qubit and coupler orbits

The approach described in this tutorial can be stated simply and generically. In theory, two observables of a QPU output are expected to be identical due to symmetries in the Ising model being studied. In experiment, they can differ systematically. We tune Hamiltonian terms to reduce these differences. In theory, symmetries in an Ising model admit identical expectation values of observables.[1] However, empirical averages over many realizations of these observables (from a QPU) may differ systematically.[2] We tune Hamiltonian terms to reduce the discrepancy between the expected value and observed averages. For example, given a Hamiltonian consisting of a single qubit with no bias, $\mathcal{H}_P = 0(s_1)$, we expect to observe a mean spin of 0 for s_1. However, this observed quantity may deviate from 0 systematically; we, thus, attempt to correct this deviation by perturbing the Hamiltonian.

In this study, we only consider one- and two-spin observables—spin magnetizations and frustration probabilities—in part because they can be fine-tuned easily using the available programmable terms in the QPU. A call to the QPU typically results in a number of classical samples, which we set to 100 for all examples. From these

1 An observable is a quantity that one can physically measure and observe. For example, the spin of a qubit.

2 Here, "differ systematically" refers to discrepancies between the expected value and the observed average as a result of biases in the physical system and not discrepancies due to finite samples.

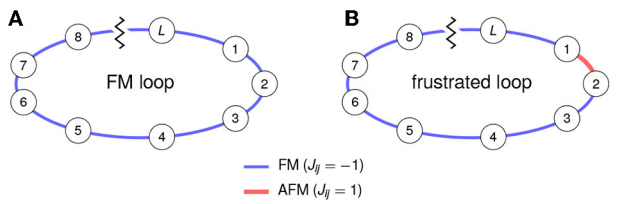


FIGURE 2
(A) Ferromagnetic loop (periodic 1D chain) on L spins. **(B)** Frustrated loop, with one antiferromagnetic coupler. The FM loop has a 2-fold degenerated ground state (all spins up or all spins down) with no frustration; the frustrated loop has $2L$ ground states, each with one frustrated bond. When $h = 0$, all qubits trivially have zero average magnetization in an ideal annealer.

samples, we can compute a magnetization as follows:

$$m_i = \langle s_i \rangle \in [-1, 1] \tag{4}$$

For each spin s_i, a *frustration probability* is as follows:

$$f_{i,j} = \frac{1 + \text{sign}(J_{i,j})\langle s_i s_j \rangle}{2} \in [0, 1] \tag{5}$$

For each coupler $J_{i,j}$; $f_{i,j}$ is the observed probability of the coupler having a positive contribution to the energy in $\mathcal{H}_P$.

This raises the first question: how do we identify observables that should be identical in expectation? The answer is through symmetries of the Ising model under spin relabelling and gauge transformation (discussed below).[3] We understand and formalize these symmetries—and automate their detection—through *graph isomorphisms* (especially automorphisms) and generalizations (Godsil and Royle, 2001). We understand and formalize these symmetries—and automate their detection—through *graph isomorphisms* (especially automorphisms) and generalizations. We briefly introduce the concept of graph automorphisms and *orbits* below (see Godsil and Royle, 2001 for a more complete treatment). Notably, the symmetries we find and exploit here are a subset of all possible symmetries.

Given a graph $G = (V, E)$ with vertex and edge sets V, E, a graph automorphism is a mapping $\pi : V \mapsto V$ such that $(\pi(u), \pi(v)) \in E$ if and only if $(u, v) \in E$. Intuitively, a graph automorphism is an adjacency-preserving relabelling of vertices. Two vertices $u, v \in V$ are said to belong in the same *vertex orbit* if there exists an automorphism mapping u to v (or v to u). If $u, v \in V$ belong in the same vertex orbit, the edges incident to u or v also belong to the same *edge orbit*.

We now relate the definitions of graph automorphisms and orbits back to our goal of detecting and exploiting symmetries. These symmetries admit two types of equivalence relations on an Ising model $\mathcal{H}_P$: one on the qubits and the other on the couplers. We call the equivalence classes *qubit orbits* and *coupler orbits*, respectively. We use notation $\mathcal{O}(s_i)$ for a qubit orbit containing spin s_i, and $\mathcal{O}(s_i, s_j)$ for a coupler orbit containing coupler (s_i, s_j).

3 A gauge transformation is also known as a *spin reversal transformation*, in which a subset of spins have their sign flipped.

We define them as having the following properties guaranteed by symmetry in an ideal annealer:

- All qubits in the same orbit have the same expected magnetization.
- All couplers in the same orbit have the same frustration probabilities.

Formally, a set $\mathcal{O}$ is said to be a qubit orbit if $s_i, s_j \in \mathcal{O}$, then $m_i = m_j$. Similarly, a set $\mathcal{O}$ is said to be a coupler orbit if $(s_i, s_j), (s_k, s_l) \in \mathcal{O}$, then $f_{i,j} = f_{k,l}$.

For example, consider the Hamiltonian $\mathcal{H}_P = hs_1 - hs_2 + hs_3$ for $h \neq 0$. The two *independent* spins s_1, s_3 can be trivially relabeled (permuted) by each other, thus the two qubits belong to the same qubit orbit; s_2 belongs in its own qubit orbit as it does not have the same magnetization as s_1, s_3. Now, let us consider the Hamiltonian $\mathcal{H}_{P,2} = \mathcal{H}_P + Js_1s_2 + Js_2s_3$ for $J \neq 0$. In this case, (s_1, s_2) and (s_2, s_3) exist in the same coupler orbit because s_1, s_3 can be swapped while preserving the couplers in $\mathcal{H}_{P,2}$; the permutation preserves adjacency structures. As a non-example, let us consider the Hamiltonian $\mathcal{H}_{P,3} = \mathcal{H}_P + Js_1s_2 - 2Js_2s_3$. In this case, (s_1, s_2) and (s_2, s_3) no longer exist in the same coupler orbit because swapping s_1, s_3 no longer preserves the couplers in $\mathcal{H}_{P,3}$.

Due to spin-flip symmetries, or *spin reversal transformations (SRTs; described below)*, each qubit and coupler orbit can additionally have up to one non-empty orbit that is *opposite*.

- If qubit orbits $\mathcal{O}(s_i)$ and $\mathcal{O}(s_j)$ are opposite,
  - We write $\mathcal{O}(s_i) = -\mathcal{O}(s_j)$ and $-\mathcal{O}(s_i) = \mathcal{O}(s_j)$.
  - If $\mathcal{O}(s_i) = -\mathcal{O}(s_j)$, $h_i = -h_j$ and, in an ideal annealer, $m_i = -m_j$.
- If coupler orbits $\mathcal{O}(s_i, s_j)$ and $\mathcal{O}(s_k, s_\ell)$ are opposite,
  - We write $\mathcal{O}(s_i, s_j) = -\mathcal{O}(s_k, s_\ell)$ and $-\mathcal{O}(s_i, s_j) = \mathcal{O}(s_k, s_\ell)$.
  - $J_{i,j} = J_{k,\ell}$ and, in an ideal annealer, $f_{i,j} = f_{k,\ell}$. $J_{i,j} = -J_{k,\ell}$ and, in an ideal annealer, $f_{i,j} = f_{k,\ell}$.

An SRT, as its name suggests, flips the sign of a spin. For example, consider the Hamiltonian with a single qubit $\mathcal{H}_P = hs$. An SRT transforms on s yields an identical Hamiltonian $\mathcal{H}_P = -h\tilde{s}$ where $\tilde{s} = -s$. Similarly, for a Hamiltonian consisting of both biases and coupling terms such as $\mathcal{H}_P = h_1s_1 + h_2s_2 + J_{1,2}s_1s_2$, we can apply an SRT on one (or multiple) variable(s) to obtain $\mathcal{H}_P = -h_1\tilde{s}_1 + h_2s_2 - J_{1,2}\tilde{s}_1s_2 = h_1s_1 - h_2\tilde{s}_2 - J_{1,2}s_1\tilde{s}_2 = -h_1\tilde{s}_1 - h_2\tilde{s}_2 + J_{1,2}\tilde{s}_1\tilde{s}_2$, where $\tilde{s}_1 = -s_1, \tilde{s}_2 = -s_2$.

In the earlier example $\mathcal{H}_P = hs_1 - hs_2 + hs_3$, qubit s_2 belongs to the orbit opposite of s_1, s_3's orbit. We will sometimes overload notation, conflating $\mathcal{O}(s_i)$ with $\mathcal{O}(i)$ and $\mathcal{O}(s_i, s_j)$ with $\mathcal{O}(i, j)$.

Qubit and coupler orbits are related to, but not identical to, automorphism orbits of an auxiliary graph. In particular, qubit and coupler orbits are not unique: putting each qubit and each coupler

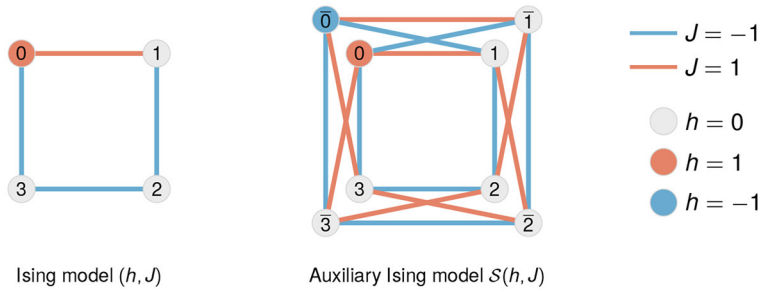


FIGURE 3
Construction of signed Ising model. To detect exploitable symmetries, we search for automorphisms of an auxiliary Ising model in which each spin is duplicated into itself and its negation; each coupler is, then, expanded to four copies of itself, two of them negated. Automorphisms of the auxiliary Ising model can be detected by conversion into an equivalent automorphism-finding problem on an edge-labeled graph. Here, vertex labels indicate the identities of spins and show how each spin is duplicated for the signed Ising model.

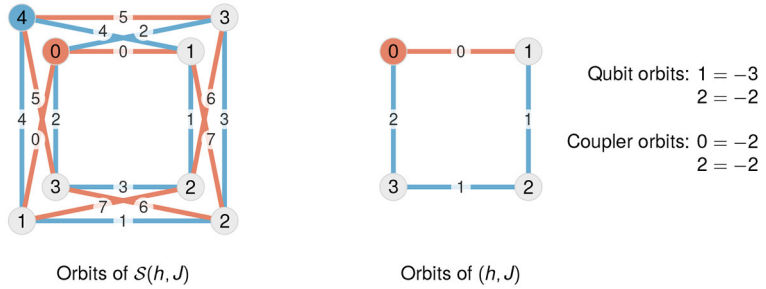


FIGURE 4
Orbits of signed and original Ising model. By computing automorphism groups of the edge- and vertex-labeled graph of the signed Ising model [Figure3 **(right)**], we can construct orbits of qubits and couplers that should behave identically by symmetry in $\mathcal{S}(h, J)$ **(left)**. Here, vertex and edge labels indicate orbits. By identifying equivalent orbits (e.g., coupler orbits 0 and 5) and reducing back to the original Ising model (h, J), we determine effective qubit and coupler orbits of (h, J) and their opposite relations **(right)**.

in a separate orbit is sufficient to meet the definition but does not provide any useful information. We seek large orbits that satisfy the requirements.

Notably, in the commonly arising situation where $h_i = 0$ on all qubits, each qubit orbit is its own opposite, so all qubits have $m_i = 0$. The analogous situation does not exist for couplers because we do not consider symmetries between pairs of qubits with zero coupling between them. Two simple examples are shown in Figure 2: a frustrated loop and an unfrustrated loop. In each case, all qubits are expected to have magnetization and all couplers are expected to have the same probability of frustration, but this is less obvious in the frustrated case than in the ferromagnetic case. In each case, all qubits and couplers have, respectively, identical magnetization and frustration probabilities. The unfrustrated case is trivially true. The frustrated scenario is less obvious but can be verified by computing frustration probabilities for each edge.

Having defined qubit and coupler orbits, we now consider how to find them.

1.3.1. Automorphisms of the signed Ising model

We proposed a strategy for identifying exploitable symmetries for calibrating a QPU by finding qubit and coupler orbits. We now introduce a method for identifying these orbits and begin by defining the *signed Ising model*.

Let (h, J) denote an Ising model with fields $h = \{h_i | v_i \in V\}$ and $J = \{J_{i,j} | e_{i,j} \in E\}$, with an underlying graph $G = (V, E)$ with vertex and edge sets V and E. We construct a signed Ising model $\mathcal{S}(h, J)$ as follows:

- For each spin $v_i \in V$, $\mathcal{S}(h, J)$ has two spins v_i and $\bar{v}_i$, with fields h_i and $-h_i$ respectively.
- For each coupler $(v_i, v_j) \in E$, $\mathcal{S}(h, J)$ has four couplers: two couplers (v_i, v_j) and $(\bar{v}_i, \bar{v}_j)$ with coupling $J_{i,j}$ and two couplers $(\bar{v}_i, v_j)$ and $(v_i, \bar{v}_j)$ with coupling $-J_{i,j}$.

Informally, we simply replace each spin with two: itself and its negation and replace each coupler with four couplers with appropriate parity-based sign flipping. Figure 3 shows an example of this construction applied to a four-spin Ising model.

Our aim is to find large qubit and coupler orbits for (h, J), and we will begin by finding the automorphism group of $\mathcal{S}(h, J)$, which can be considered as a vertex- and edge-labeled graph. Our aim is to find qubit and coupler orbits. Because automorphisms of the underlying graph of an Ising model are symmetries of the Ising model, we can generate qubit and coupler orbits by considering the graph symmetries alone. In other words, finding

graph automorphisms of the Ising model effectively give us qubit and coupler orbits. We can stop here and perform calibration based on these qubit and coupler orbits extracted from these symmetries (spin relabelling symmetries). However, we can similarly extract and exploit spin-flip symmetries by finding the automorphisms of the graph of $\mathcal{S}(h, J)$. These additional automorphisms give rise to qubit and coupler orbits opposite to the original. Intuitively, $\mathcal{S}(h, J)$ enumerates and concatenates all SRT configurations to the original Ising model. As a consequence, by finding its automorphisms, we are able to further identify symmetries due to SRTs. In other words, if a negated vertex is in the same orbit as a non-negated vertex, they exhibit symmetries through an SRT. In short, the automorphism group of $\mathcal{S}(h, J)$ naturally generates one equivalence relation defining qubit orbits and another equivalence relation defining coupler orbits (see Figure 4).[4] Our orbits of $\mathcal{S}(h, J)$ immediately give us orbits of (h, J) and constructed by simply discarding the qubits and couplers that do not exist in (h, J).

There is more usable information held in the orbits of $\mathcal{S}(h, J)$. First, we can combine coupler orbits of $\mathcal{S}(h, J)$ such that for each coupler $e_{i,j} \in E$, $(\bar{v}_i, v_j)$ and $(v_i, \bar{v}_j)$ are in the same orbit, and (v_i, v_j) and $(\bar{v}_i, \bar{v}_j)$ are in the same orbit. Second, we can, then, easily derive opposite orbits: $\mathcal{O}(v_i) = -\mathcal{O}(\bar{v}_i)$, and $\mathcal{O}(v_i, v_j) = -\mathcal{O}(\bar{v}_i, \bar{v}_j)$.

These orbits are already very useful, but we can combine some to make even larger orbits. As demonstrated in the example in Figure 3, in $\mathcal{S}(h, J)$, the couplers between pairs $(\bar{v}_i, v_j)$ and $(v_i, \bar{v}_j)$ are not necessarily automorphic. However, they are clearly equivalent under a flip of all spins. Thus, we combine the coupler orbits containing these two couplers. Likewise, the same applies to couplers between pairs $(\bar{v}_i, \bar{v}_j)$ and (v_i, v_j). This is all demonstrated in the accompanying code `example0_1_orbits.py` and shown in Figure 4.

We now consider how to exploit orbits to improve performance in quantum annealers, building up a set of tools in the following worked examples.

2. Worked example: ferromagnetic loop

Code reference: `example1*.py`.

For our first example of calibration refinement, we study the ferromagnetic (FM) loop (Figure 5) in which each coupling $J_{1,2} = J_{2,3} = \cdots = J_{N,1}$ is equal and each field h_i is zero. In this case, by rotation, it is obvious that all qubits are in the same orbit and all couplers are in the same orbit. Furthermore, the orbit containing all qubits is its own opposite. Thus, we will perform two refinements. First, we will balance each qubit at zero magnetization $m_i \approx 0$. Second, we will balance the couplings so that each coupler is frustrated with approximately equal probability.

Since the FM loop has no frustrated bonds in the ground state, the latter condition is only interesting if we sample excited states.

4 Since the automorphism-finding code `nauty`, McKay and Piperno (2014) only handles vertex-labeled graphs and not edge-labeled graphs, and we need to construct a vertex-labeled graph G'' from $\mathcal{S}(h, J)$, which gives us the appropriate automorphism group.

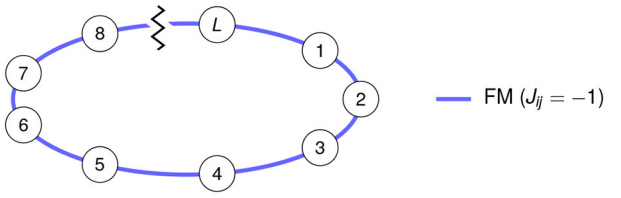


FIGURE 5
Ferromagnetic loop.

To ensure abundant excitations, we study a reasonably long loop with weak couplings: $L = 64$ and $J_{ij} = -0.2$.

2.1. Finding multiple embeddings of a small Ising model

Code reference: `embed_loops.py`.

The first task is to find a copy of the FM loop in the qubit connectivity graph A_{QPU} of the QPU being used. This is an *embedding*—a mapping of spins of an Ising model to qubits in a QPU. In an Advantage processor, a 64-qubit loop can be embedded many times on disjoint sets of qubits, so we can run many copies in parallel for a richer and larger set of measurements.

To find these embeddings, we use the Glasgow graph solver (McCreesh et al., 2020), which has been incorporated into the embedding finding module `minorminer` (D-Wave, 2023). To make the embedding search faster, we raster-scan across 2×2 blocks of unit cells in the QPU's Pegasus graph (Boothby et al., 2020) and then greedily construct a large set of non-intersecting embeddings. The file `embed_loops.py` provides a code example that finds multiple disjoint copies of a 64-qubit loop in A_{QPU}.

2.2. Balancing qubits at zero

Code reference: `example1_1_fm_loop_balancing.py`.

We will use simple parameters for the experiment, running 1 µs anneals forward anneals (where s increases linearly in time as t/t_a) and drawing 100 samples for each QPU call. We set `auto_scale=False` to ensure that the QPU will not automatically magnify the energy scale.

In D-Wave's annealing QPUs, each qubit s_i can be biased toward -1 or $+1$ in two ways: first, with a programmable longitudinal field h_i as in Equation 1; second, with a programmable flux-bias offset (FBO) Φ_i (Harris et al., 2009; D-Wave, 2022). In the quantum annealing Hamiltonian (3), the bias conferred by the h_i term is scaled by $\mathcal{J}(s)$, meaning that it changes as a function of s. The FBO Φ_i, in contrast, confers a constant bias that is independent of s. We prefer to mitigate biases using FBOs, in part, because they are programmed independently of h_i.

We employ an iterative gradient descent method for minimizing $|m_i|$ with a step size α_Φ. For a given iteration, we consider the observed magnetization $m_i = \langle s_i \rangle$. If $m_i < 0$, we adjust the FBO to push s_i toward $+1$; if $m_i > 0$, we adjust the FBO to push s_i toward -1. This is done by updating as follows:

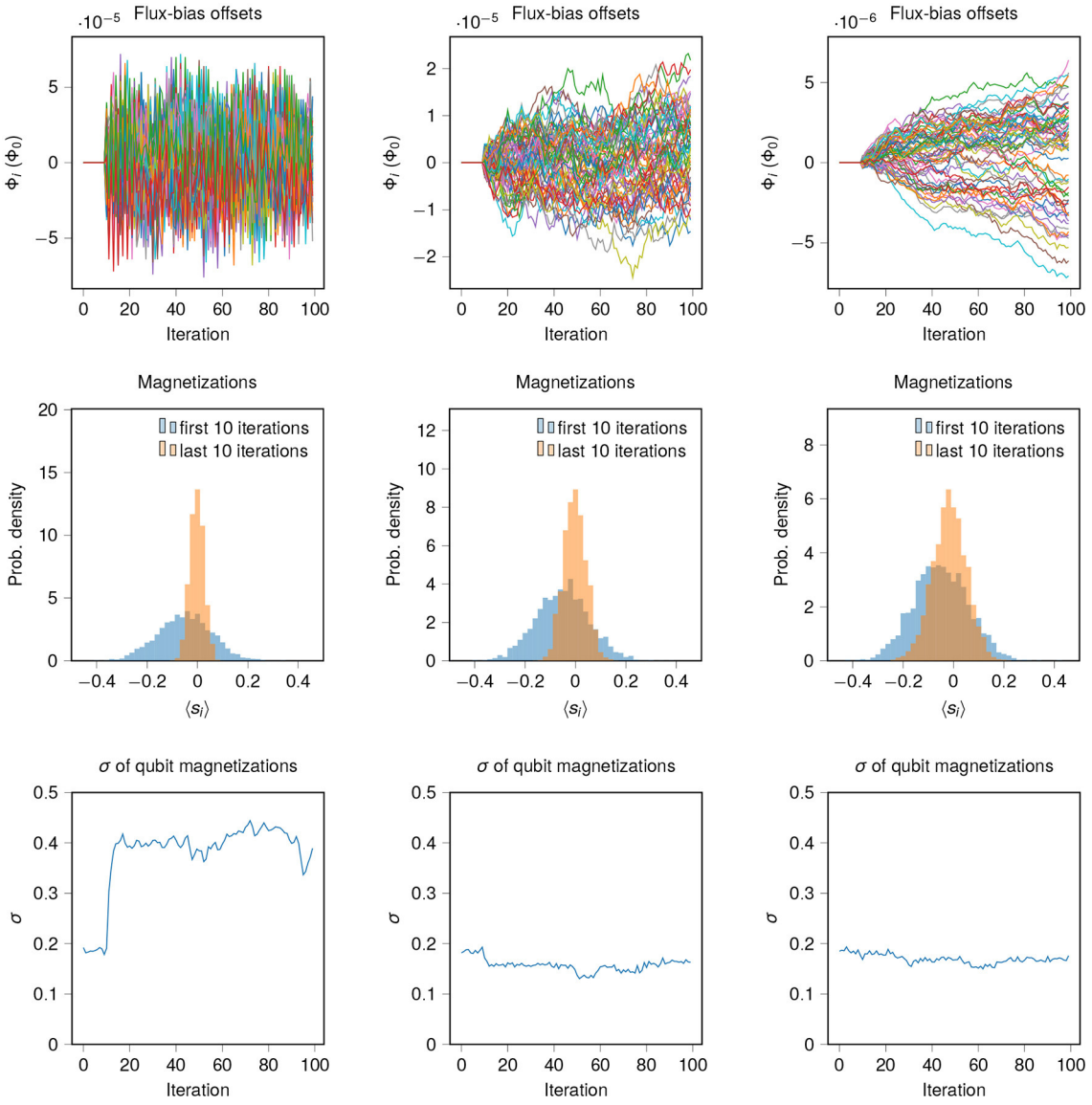


FIGURE 6
Balancing qubits in a FM chain with flux-bias offsets. Iterative correction of qubit biases is demonstrated using three step sizes α_Φ for 100 iterations: 10^{-4} **(left)**, 10^{-5} **(middle)**, and 10^{-6} **(right)**. Step size is set to zero for the first 10 iterations. **(Top)** Evolution of flux-bias offsets for 64 qubits in an FM chain. **(Middle)** Qubit magnetization averaged over first 10 iterations and last 10 iterations. **(Bottom)** Standard deviation of qubit magnetizations per iteration.

$$\Phi_i \leftarrow \Phi_i - \alpha_\Phi(m_i - \bar{m}) \tag{6}$$

where $\bar{m}$ is the average observed magnetization across all qubits. In this case, we can simply replace $\bar{m}$ with 0 since $h_i = 0$ for all qubits.

The choice of a step size α_Φ has a strong influence on the convergence of the calibration procedure. In Figure 6, we show the resulting FBOs for a single copy of the 64-qubit chain, as well as magnetization statistics. We show experiments for three choices of α_Φ. One (flux 1×10^{-4}, in units of Φ_0) is too large and creates oscillations in Φ_i and m_i. One (1×10^{-6}) is too small and takes many iterations to converge. One (1×10^{-5}) is in between and performs well. The choice of step size is a common concern in gradient descent applications, and we will consider automatic tuning of α_Φ in a later section (Section 4.4). For best results, we should ensure:

- The calibration refinement appears to have converged to the vicinity of a fixed point.
- The parameters do not oscillate wildly.

When seeking evid ence that qubit bias is improved by the FBOs, we should not just look at qubit statistics over a single QPU call, since fluctuations can be large. Rather, we should look at the average magnetization of a qubit over multiple calls, which indicates systematic bias. The middle row of Figure 6 shows the average magnetization of each qubit across the first and last 10 iterations. For each step size, the shim results in a significant improvement in variation of m_i from one qubit to another. However, the standard deviation among qubit magnetizations for *individual* iterations shows that the case $\alpha_\Phi = 1 \times 10^{-4}$ causes

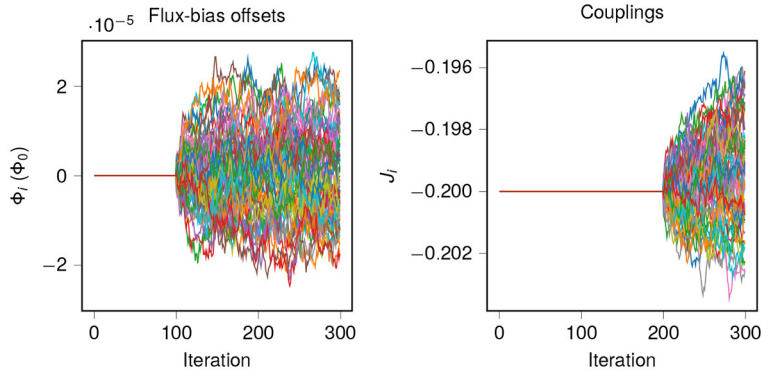


FIGURE 7
Balancing qubits and couplers in an FM chain with flux-bias offsets and coupler adjustments. This experiment is similar to that shown in Figure 6 but with $\alpha_J > 0$ for the last 100 iterations. Couplers remain distributed about the average value of $J = -0.2$.

broad spreading of biases, so we need to be careful with our step sizes.

2.3. Balancing spin-spin correlations

Code reference:
`example1_2_fm_loop_correlations.py`.

Having balanced qubits at zero with linear terms with an FBO shim, we now address homogenizing the spin-spin correlations on adjacent qubits, which by symmetry should be equal for all coupled pairs. The couplings $J_{i,i+1}$ are all nominally -0.2; we will fine-tune the couplings in the vicinity of this value. This is similar to how we fine-tuned the FBOs but with the added constraint that we do not change the average coupling.

For a given iteration, we take the observed probability $f_{i,i+1}$ of the coupler being frustrated as follows:

$$f_{i,i+1} = (1 + \text{sign}(J_{i,i+1})\langle s_i s_{i_1}\rangle)/2. \tag{7}$$

Let $\bar{f}$ denote the average frustration across all couplers in all disjoint embeddings of the chain, in general, we will compute $\bar{f}$ across all couplers in the union of a coupler's orbit and its opposite orbit. We, then, adjust couplings based on the *residual frustration* $f_{i,i+1} - \bar{f}$:

$$J_{i,i+1} \leftarrow J_{i,i+1}(1 + \alpha_J(f_{i,i+1} - \bar{f})). \tag{8}$$

Figure 7 shows data for the same experiment as Figure 6 but with the "coupler shim" added, with $\alpha_J = 0.001$. To show the effect of the two shims, we run 100 iterations with $\alpha_\Phi = 0$ and $\alpha_J = 0$, then 100 iterations with $\alpha_\Phi = 1 \times 10^{-5}$ and $\alpha_J = 0$, then 100 with $\alpha_\Phi = 1 \times 10^{-5}$ and $\alpha_J = 0.001$. In this particular case, the coupler shim is small but some systematic signals can be observed. We will show more impactful cases later in the tutorial.

3. Worked example: frustrated loop

Code reference: `example2*.py`.

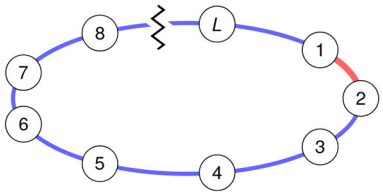


FIGURE 8
Frustrated loop.

Taking the ferromagnetic loop considered in the previous example, the sign is flipped of a single coupler $J_{1,2}$ (Figure 8). It is again obvious that all spins should have zero average magnetization, since there is no symmetry-breaking field (i.e., $h_i = 0$ everywhere). Less obvious is the fact that we can have two coupler orbits: one containing all FM couplers and one containing the AFM coupler, and they are opposite. Consequently, every coupler should be frustrated with equal probability in an ideal annealer.

3.1. Finding orbits

Code reference:
`example2_1_frustrated_loop_orbits.py`.

We can derive this fact as follows. Flipping the sign of s_2, and the sign of both couplers incident to it, is a gauge transformation and, as such, will not change the probability of any coupler being frustrated in an ideal annealer. The result of this gauge transformation is again a frustrated loop with a single AFM coupler $J_{2,3}$; this is equivalent to the original loop by a cyclic shift of qubit labels. From this, we can infer that $J_{1,2}$ and $J_{2,3}$ should have the same frustration probability; repeating this argument tells us that all couplers should have the same frustration probability.

For more complicated examples, we would prefer to find such statistical identities programmatically as described in Section 1.3. We do this in the file

`example2_1_frustrated_loop_orbits.py`

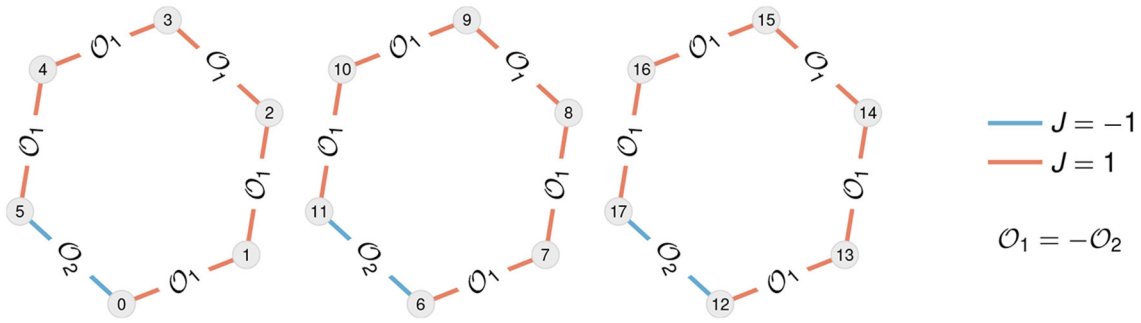


FIGURE 9
Coupler orbits of frustrated loops. The code `example2_1_frustrated_loop_orbits` constructs three disjoint frustrated loops and programmatically generates qubit and coupler orbits. All qubits are in the same orbit. There are two signed coupler orbits, $\mathcal{O}_1$ and $\mathcal{O}_2$, and in this example, they form an opposite pair, meaning that a coupler in $\mathcal{O}_1$ and a coupler in $\mathcal{O}_2$ have opposite sign ($J = -1$ and $J = 1$ in this case) but equal probability of frustration in an ideal annealer.

by computing automorphisms of an auxiliary graph. The result is a mapping $\mathcal{O}$ of qubits and couplers to orbits. If spins s_i and s_j satisfy $\mathcal{O}(s_i) = \mathcal{O}(s_j)$, then in an ideal annealing experiment $m_i = m_j$. Similarly, if couplers $s_i s_j$ and $s_k s_\ell$ satisfy $\mathcal{O}(s_i s_j) = \mathcal{O}(s_k s_\ell)$, they have identical frustration probabilities $f_{i,j} = f_{k,\ell}$. The code also gives us a mapping of orbits to "opposite" orbits, such that if spins s_i and s_j are in opposing orbits, $m_i = -m_j$, and if couplers $s_i s_j$ and $s_k s_\ell$ are in opposing orbits, $f_{i,j} = f_{k,\ell}$ and $J_{i,j} = -J_{k,\ell}$.

Running the code on three disjoint copies of a frustrated six-qubit loop tells us that all AFM couplers are in one coupler orbit $\mathcal{O}_1 = \{s_i, s_j \mid \mathcal{O}(s_i s_j) = 1\}$, and all FM couplers are in its opposite, $\mathcal{O}_2 = -\mathcal{O}_1$ (see Figure 9).

We point out an obvious but useful fact: If we are using multiple embeddings of an Ising model, all copies of a given qubit are in the same orbit, and all copies of a given coupler are in the same orbit. Here, we use disjoint embeddings, but they need not be disjoint: the embeddings could overlap, and be annealed in separate calls to the QPU.

3.1.1. Shimming

We can now approach the frustrated loop similarly to the unfrustrated loop: all qubits should have average magnetization zero, and all couplers should be frustrated with the same probability. Again, tuning FBOs and individual couplings helps to reduce bias in the system. This example shows how to exploit orbits for our shim.

There is one detail worth pointing out. In Equations 6, 8, the terms $\bar{m}$ and $\bar{f}$ can be computed as averages over an orbit. If we are dealing with opposing qubit orbits $\mathcal{O}_q$ and $-\mathcal{O}_q$, we can simply use $\bar{m} = 0$, as we do in the first example. For opposing coupler orbits $\mathcal{O}_c$ and $-\mathcal{O}_c$, we can compute $\bar{f}$ across the union of the two orbits. In this case, that means that $\bar{f}$ is the average frustration probability across all couplers.

Figure 10 shows the results of shimming FBOs and couplings for 165 parallel embeddings of a 16-qubit frustrated loop, using nominal coupling strength $|J_i| = 0.9$. Here, both components of the shim show a marked improvement of statistical homogeneity. Taking moving means for 10 iterations at a time, we see that both σ_m (standard deviation of qubit magnetization) and σ_f (standard deviation of coupler frustration probability) decrease as a result of turning on the FBO shim and the coupler shim, respectively.

3.1.2. Finding orbits of an arbitrary Ising model

Code reference: `example2_3_buckyball_orbits.py`.

Here we present an example of an Ising model that is read from a text file and run through our orbit-finding code. The user may want to edit this code to analyze other Ising models of interest.

Consider another antiferromagnetic Ising model ($J_{ij} = 1$) with a *Buckyball graph* as its underlying structure and no linear fields ($h_i = 0$). We apply the same methodology described in Section 1.3 to find its orbits. Figure 11 visualizes the Buckyball model with its orbits labeled by text, as well as its signed Ising counterpart with coupling values encoded by color.

4. Worked example: triangular antiferromagnet

Code reference: `example3*.py`.

In the previous examples we demonstrated several key methods:

- Finding qubit and coupler orbits.
- Homogenizing magnetizations with FBOs.
- Homogenizing frustration by tuning couplers.

We can now apply these tools to a non-trivial system: the triangular antiferromagnet (TAFM; Figure 12). This is a classic example of a frustrated 2D spin system. Moreover, the addition of a transverse field to a TAFM leads to order-by-disorder at low temperature (Moessner and Sondhi, 2001; Isakov and Moessner, 2003). For this and other reasons, including qualitative similarity to real materials, the TAFM has been simulated extensively using quantum annealers (King et al., 2018, 2022). We will use it as an example to showcase several concepts in calibration refinement for quantum simulation:

- Truncating and renormalizing Hamiltonian terms.

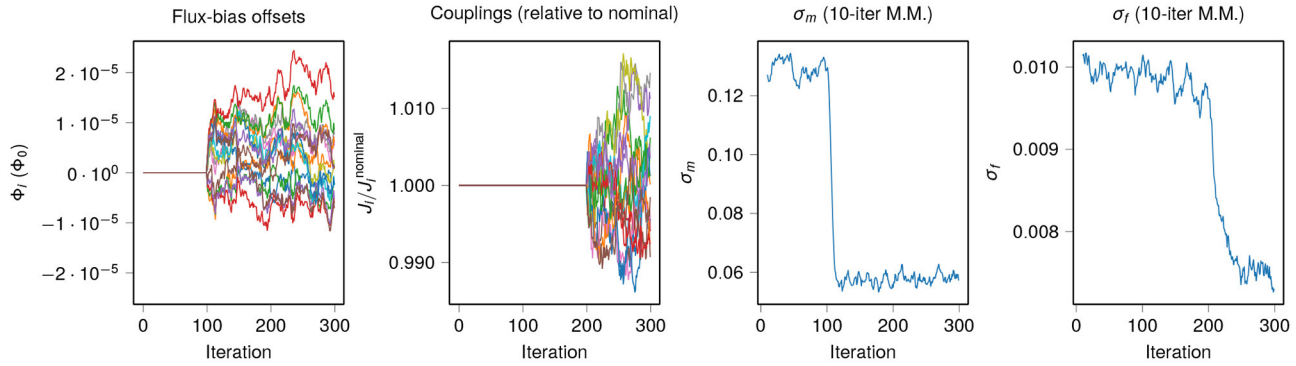


FIGURE 10
Shimming a frustrated loop. Three hundred iterations are performed. A flux-bias offset shim is used after iteration 100, and a coupler shim is used after iteration 200. Nominal couplings are ±0.9. The third panel shows the standard deviation of qubit magnetizations taken as a moving mean over 10 iterations, σ_m. The fourth shows the corresponding quantity σ_f for frustration probability.

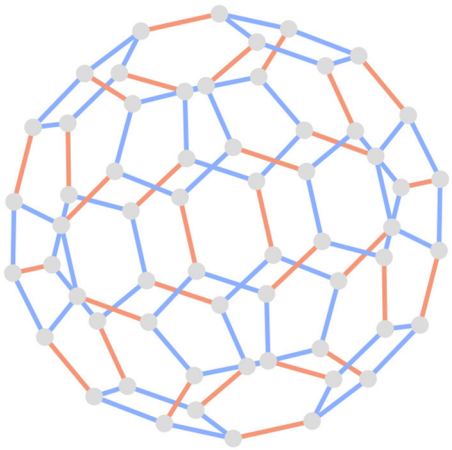


FIGURE 11
An antiferromagnetic Ising model with a Buckyball graph structure. The node and edge colors encode the resulting qubit and edge orbits respectively. All qubits are in the same orbit since the graph is vertex-transitive. There are only two coupler orbits: those couplers sitting between two hexagons, and those sitting between a hexagon and a pentagon.

- Simulating logical vs. embedded systems.
- Simulating an infinite system vs. faithfully simulating boundary conditions. When simulating an infinite system, we use the same geometry but suppress effects of any open boundaries, which otherwise cause statistics such as nearest-neighbor correlations to vary depending on distance from the boundary. To do this we determine our qubit and coupler orbits assuming an inifinite lattice, instead of computing them from the finite lattice at hand.

4.1. Embedding as a square lattice

Code reference: `example3_1_tafm_get_orbits.py`.

In D-Wave's Advantage systems, we can minor-embed the TAFM using two-qubit FM chains. First, we will embed a 12 × 12 square lattice with cylindrical boundary conditions, then we ferromagnetically couple pairs of qubits with a strong coupling J_{FM}. The cylindrical boundaries are very helpful in providing rotational symmetries that we can exploit in our calibration refinement methods (as in the 1D chains already studied).

The provided code uses the Glasgow subgraph solver to find embeddings of the 12 × 12 square lattice, but note that this can take several hours. For larger square lattices, up to 32 × 32 or even larger depending on the location of inoperable qubits, one can inspect embeddings of smaller lattices and generalize the structure, since subgraph solvers are unlikely to be efficient at that size. We proceed with 10 disjoint 12 × 12 embeddings generated by the code.

In this example we will set AFM couplers to $J_{\text{AFM}} = 0.9$, and all FM couplers to $J_{\text{FM}} = -2 * J_{\text{AFM}}$. Since FM couplers are very rarely frustrated in this system, we will only shim the AFM couplers.

4.2. Annealing with and without shimming

Code reference: `example3_2_tafm_forward_anneal.py`.

As in the previous example, we will compare performance of three methods: no shim, FBO shim only, and FBO and coupler shims together. We perform 800 iterations, turning on the FBO shim after 100 iterations and the coupler shim after 300 iterations. Figure 13 shows data for this experiment, and we can see that as with the frustrated loop example, shimming improves statistical homogeneity of magnetizations and frustration. Note, however, that there is no appreciable impact on the average magnitude of the order parameter $\langle|\psi|\rangle$. This will change when we vary boundary conditions (see Figure 16).

4.3. Manipulating orbits to simulate an infinite system

Code reference: `example3_2_tafm_forward_anneal.py`.

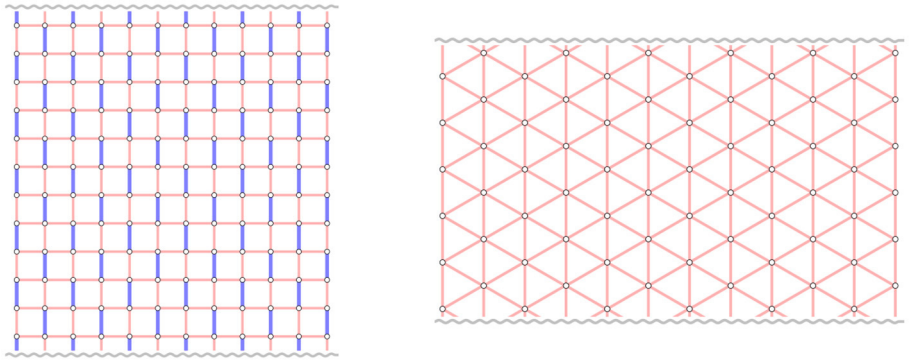


FIGURE 12
A 12 × 12 square lattice with cylindrical boundary conditions (periodic top/bottom). Contracting two-qubit FM chains into single spins results in a triangular antiferromagnet.

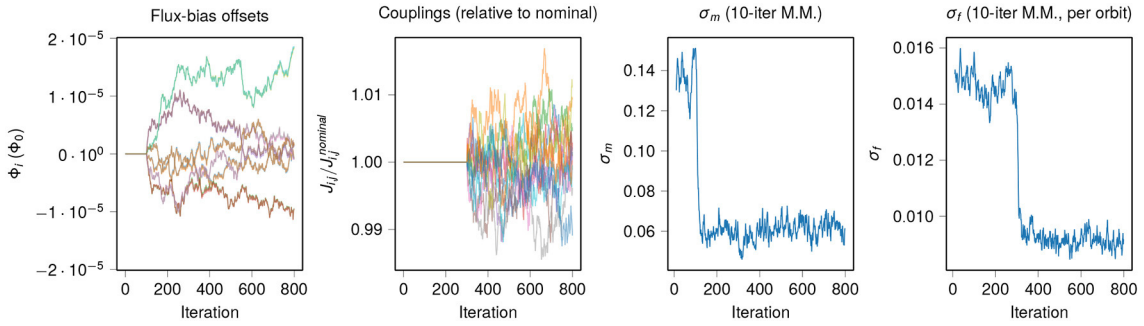


FIGURE 13
Shimming an embedded cylindrical triangular antiferromagnet. Eight hundred iterations are performed. A flux-bias offset shim is used after iteration 100, and a coupler shim is used after iteration 300. For clarity, we only show FBOs for 12 qubits, and couplings for 12 couplers in the same orbit. Standard deviation of frustration probabilities, σ_f, is computed for the couplers in each orbit, and the average over all orbits is taken.

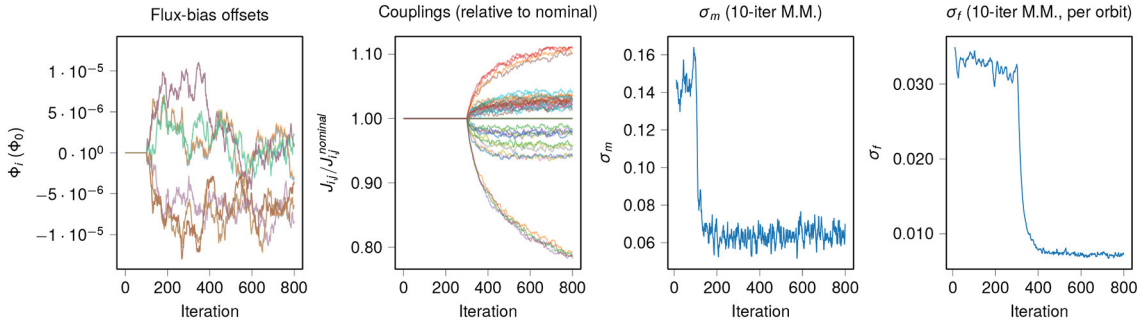


FIGURE 14
Shimming an isotropic, infinite triangular antiferromagnet. The experiment from Figure 13 is repeated, but with all AFM couplers placed in the same orbit. For clarity, we only show FBOs for 12 qubits, and every 5th coupling from the AFM orbit.

The shim shown in Figure 13 used coupler orbits for the square lattice with cylindrical boundaries, which are naturally different for couplers that are different distances from the boundary, or different orientations with respect to the boundary (and to FM chains). But what if we want to simulate, to the extent possible, an infinite TAFM? In that system, a coupler's probability of frustration is independent of its orientation and position, unlike in the square-lattice embedded system. We can simulate this case by putting all AFM couplers in one orbit, and all FM couplers in a second orbit, and proceeding as before. The coupler orbits no longer reflect the structure of the programmed Ising model, but rather the structure of the Ising model we wish to simulate.

Results for the "infinite triangular" shim are shown in Figure 14. This experiment is performed just like the previous one, but with

the parameter

```
shim['type']='triangular_infinite'
```

instead of

```
shim['type']='embedded_finite'.
```

The coupler shim deviates significantly from nominal values (note axis scale), and has not converged even after 500 iterations.

4.3.1. Truncating and renormalizing couplers

In this code example (and others) we use an important method in the coupler shim: truncation. Programmed couplings must be in the range $[-2, 1]$, so AFM couplers must remain < 1, which is $1.11 * J_{\text{AFM}}$. Therefore, when couplers go out of range, we truncate them to within the range. To avoid persistent shrinking of the couplings due to truncation, we renormalize to the correct average coupling value (0.9) before truncation—this prevents cumulative shrinkage over many iterations.

4.3.2. Better initial conditions

Looking at the data, we can see that the most reduced couplers are those on the boundary. This suggests that if we want to simulate the infinite TAFM, we should start with a thoughtful setting of couplers. In this case, setting the AFM couplers on the boundary to $J_{\text{AFM}}/2$ reduces the need to shim enormously. This makes sense, since doing so maximizes the ground-state degeneracy of the classical system, as previously noted (King et al., 2018).

This shim is shown in Figure 15. The experiment is performed just like the previous one, but with the parameter

```
param['halve_boundary_couplers']=True
```

instead of

```
param['halve_boundary_couplers']=False.
```

We can see that now, the coupler shim only deviates a few percent from nominal, at most.

4.3.3. Complex order parameter

Order in the TAFM can be characterized by a complex order parameter ψ, which we define now. Let $c : S \rightarrow \{0, 1, 2\}$ be a 3-coloring of the spins of the TAFM, mapping them onto three sublattices so that no two coupled spins are in the same sublattice (this coloring is unique, up to symmetries). Then for a spin state S we can define

$$\psi(S) = \frac{\sqrt{3}}{N} \sum_{\ell}^{N} \left(s_\ell e^{c_\ell 2\pi/3} \right), \tag{9}$$

where $c_i = c(s_i)$ and $i = \sqrt{-1}$. Due to symmetries among the sublattices arising from the cylindrical boundary condition, as well as up-down symmetry of spins since $h = 0$, we expect sixfold rotational symmetry (among other symmetries) in the distribution of ψ in an ideal annealer. Thus ψ can serve as a good indicator of any biases in the system, as well as global ordering.

We can use ψ to compare the "embedded finite" shim and "triangular infinite" shim, as seen in Figure 16. Although we are simply forward-annealing the system, and therefore not sampling from the mid-anneal Hamiltonian, we expect the same characteristic ring histogram—without a peak near $\psi = 0$—that is seen in the quantum system (cf. King et al., 2018, Figure 3C). This is seen only after the "triangular infinite" shim. We mainly attribute this to the halving of the boundary couplings. In all cases, the shim improves the theoretically expected sixfold rotational symmetry of ψ.

4.4. Adaptive step sizes

Code reference: `example3_2_tafm_forward_anneal.py`.

It is often difficult or impractical to determine appropriate step sizes *a priori*. Here we demonstrate a simple method for adapting step sizes based on statistics of the shim. Note that due to noise in the QPU's surrounding environment, there is no well-defined asymptote or steady state for a shim. However, we act as though such a state exists: we expect high-frequency fluctuations in the environment to be small compared to low-frequency fluctuations and static cross-talk.

If the step size is sufficiently small and we are sufficiently close to the steady state, we can expect fluctuations of the Hamiltonian terms (FBOs, couplers, or fields) to behave like unbiased random walks. In an unbiased random walk with position $x(t)$ at time $t = 0, 1, \ldots$, the probability distribution of $x(t)$ approaches the normal distribution with mean 0 and variance t.

If the shim is far from the steady state and has a relatively small step size, the random walks will be biased in one direction, and thus the variance of fluctuations will grow superlinearly in t. Finally, if the step size is very large, then it will tend to overshoot the steady state, and oscillate. This leads to variance of fluctuations growing sublinearly in t. Thus we can periodically adjust the step size of a shim as follows, using a 20-iteration lookback and a tuning term $\varepsilon = 0.1$:

1. For $d \leq 20$, $x(t) - x(t - d)$ is the difference between the current shim value for a term (e.g. FBO) and the value d iterations previous. Let X_d be the set of all $x(t) - x(t - d)$ for all x being tuned.
2. Find a best-fit exponent b describing $\text{var}(X_d) \propto d^b$.
3. If $b > 1.1$, multiply the step size α by $1 + \varepsilon$.
4. If $b < 0.9$, divide the step size α by $1 + \varepsilon$.

In the example code `example3_2_tafm_forward_anneal.py`, this method is applied by setting

```
adaptive_step_size = True.
```

This check is done every iteration, but this is not necessary.

Adaptive step sizes are so far a largely unexplored research area, and various approaches could be taken. Using different step sizes for each orbit is certainly worth exploring; note in Figure 14 that different coupler orbits have hugely varying deviations from

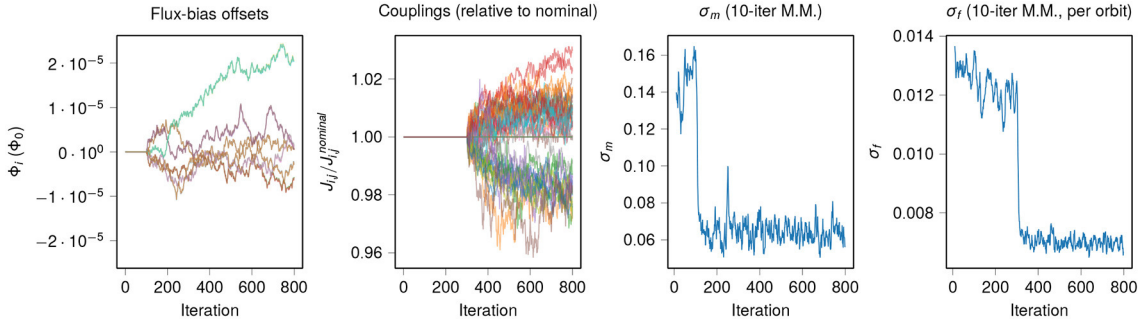


FIGURE 15
Shimming an isotropic, infinite triangular antiferromagnet, starting with halved boundary couplers. The experiment from Figure 14 is repeated, but with all AFM couplers on the boundary halved (to $J_{AFM}/2 = 0.45$) as an initial condition. For clarity, we only show FBOs for 12 qubits, and every 5th coupling from the AFM orbit.

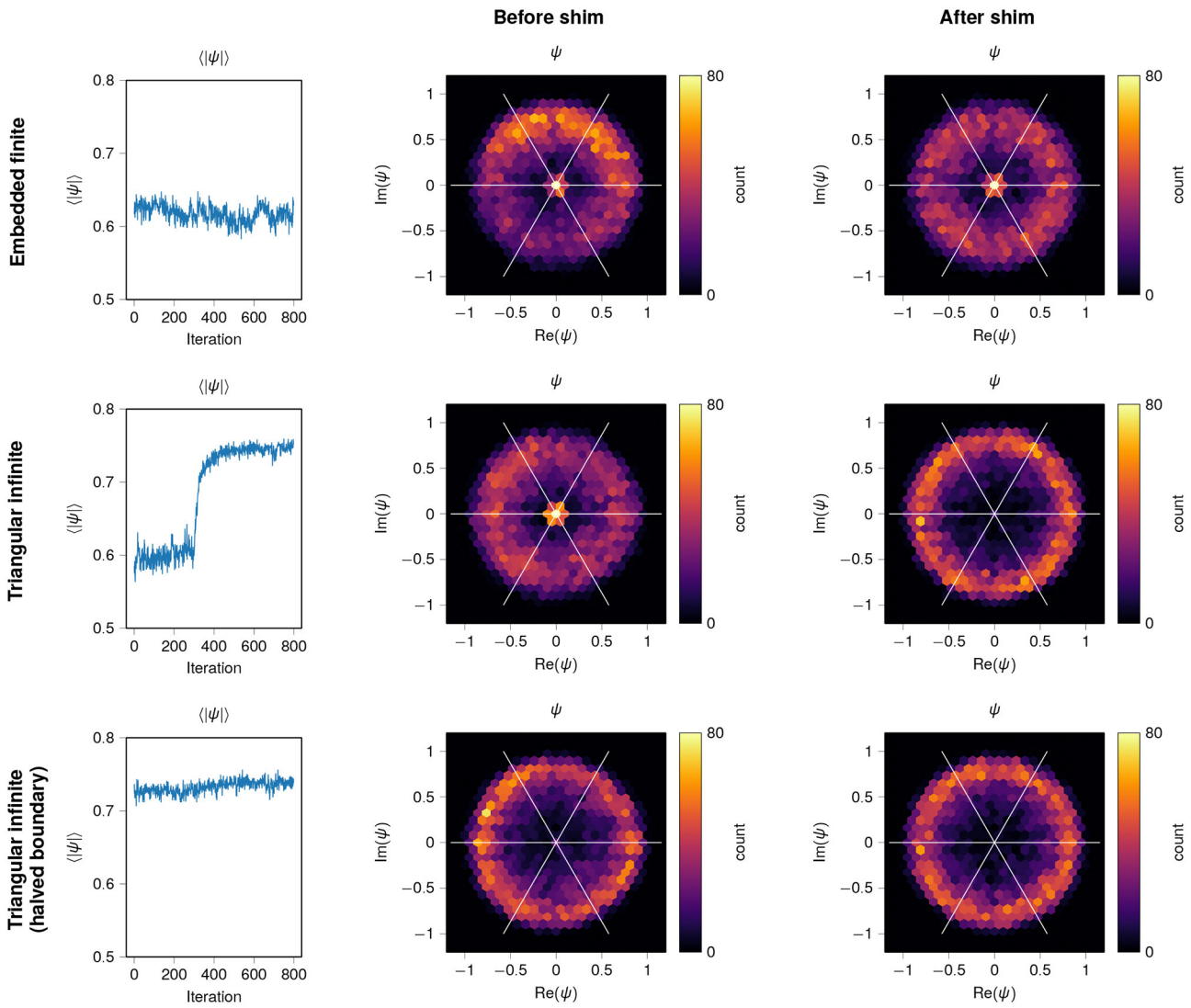


FIGURE 16
Complex order parameter ψ. For the three shims shown in Figures 13–15, we plot the evolution of the average magnitude $\langle\psi\rangle$, as well as complex histograms of ψ (showing only data for one of the ten embeddings) before and after shimming.

the mean. More general frameworks like "Adam" (Kingma and Ba, 2014) could also be useful in this context.

5. A survey of additional methods

We have provided detailed demonstrations and free-standing Python implementations for several worked examples. These cover the basics of calibration refinement. Here we discuss some additional methods that have been used successfully in recent works.

5.1. Shimming a system in a uniform magnetic field

Certain Ising models in a uniform magnetic field are of interest to physicists, and these have been simulated in quantum annealers both at equilibrium (Kairys et al., 2020) and out of equilibrium (King et al., 2021a). If we want to simulate an infinite system, we would ideally study a large system with no missing spins, and with fully periodic boundaries. However, this is often not possible, so we wish to make the magnetization m_i independent of the spin's position relative to the boundary (although it may depend on the spin's position in a unit cell of the lattice being simulated). In a typical experiment, we want to measure a system under an average field $\bar{h}$ for each value in an increasing set of equally spaced values $\{\bar{h}^{(1)}, \bar{h}^{(2)}, \ldots, \bar{h}^{(m)}\}$.

To deal with this, we can shim individual longitudinal field terms, h_i, such that all spins of a given type (i.e., in the same position of the unit cell) are in the same orbit. We can then shim all h_i terms for each simulated field magnitude $\bar{h}^{(k)}$ that we want to study, and denote each individual term $h_i^{(k)}$. After each iteration we renormalize the fields so the average value $\frac{1}{N}\sum_{i=1}^{N} h_i^{(k)}$ remains equal to $\bar{h}^{(k)}$ throughout the shim, perhaps with an adjustment arising from boundary spins.

To shim the case $\bar{h} = 0$, we use FBOs (as in the worked examples) instead of tuning h_i. To shim the case $\bar{h}^{(k)} = 0$, we use FBOs (as in the worked examples) to set a zero point instead of tuning $h_i^{(k)}$. In doing so, we can compensate for time-dependent flux drift, particularly when determining the location (in $\bar{h}$) of a phase transition.

We can additionally ensure that each h_i is a locally smooth function of $\bar{h}$ by adding a smoothing term. For example, if h_i has values h_i^- and h_i^+ for the next lower and higher values of $\bar{h}$ being simulated, we can make the adjustment.

We can additionally ensure that each sequence $\{h_i^{(1)}, \ldots, h_i^{(m)}\}$ is a smooth function of $\bar{h}$ by adding a smoothing term.[5] For example, we can add make an adjustment in two steps. First, set all

$$\tilde{h}_i^{(k)} \leftarrow (1-\varepsilon)h_i^{(k)} + \epsilon\left(h_i^{(k-1)} + h_i^{(k+1)}\right)/2, \tag{10}$$

for intermediate values of k and some smoothing constant $0 < \epsilon < 1$. Second, set all $h_i^{(k)} \leftarrow \tilde{h}_i^{(k)}$.

5.2. Shimming an Ising model with no symmetries

In King et al. (2021b), a qubit spin ice was implemented using a checkerboard Ising model. The system had open boundary conditions and missing spins due to inoperable qubits, so no geometric symmetries were available. However, due to the rich automorphism group of the qubit connectivity graph (ignoring unused qubits), it was possible to generate many distinct embeddings of the same system, using different mappings of qubits to spins. Therefore we could simulate a collection of distinct embeddings (in this case, 20) and shim in the same way we did in the worked examples. The only difference is that in the qubit spin ice example, the embeddings are not disjoint and therefore must be sampled from using separate calls to the QPU. However, once we have a set of samples from each embedding, we can analyze the data as though the embeddings are disjoint, whether or not this is actually the case. The benefit remains the same: by simulating with 20 distinct embeddings, we get qubit and coupler orbits of size at least 20.

5.3. Shimming a collection of random inputs

In King et al. (2023), shimming was used to study *spin-glass ensembles*—collections of random problems with certain parameters. As we have seen, we can spend hundreds of iterations shimming a single problem, and this becomes impractical when studying ensembles of thousands of instances.

The approach used was to exploit a common symmetry: all problems in the ensembles had $h = 0$. Shimming the couplers was abandoned as being impractical for such a large set of inputs. Shimming FBOs, however, is straightforward. By cycling through 300 spin-glass realizations using the same set of qubits and couplers, simulating each realization several times, it is possible to combine the work and arrive at a good set of FBOs that mitigates the majority of systematic offsets.

5.4. Shimming anneal offsets for fast anneals

As described in the Supplementary material to King et al. (2022), D-Wave quantum annealing processors have recently demonstrated the capacity to anneal much faster than currently generally available, at an anneal time of 10 nanoseconds or less (King et al., 2022, 2023). This speed exceeds the ability of the control electronics to synchronize the annealing lines (eight in the Advantage processor, four in D-Wave 2000Q^{TM}) satisfactorily. Therefore, frustration statistics can be used to infer which lines are out of sync with the others, and in which direction. Anneal offsets, which allow individual qubits to be annealed slightly ahead

5 A smooth function is desirable because a shim is a perturbation of the Hamiltonian intended to compensate for small non-idealities in the quantum annealer.

of or behind other qubits, were used to synchronize the qubits on each annealing line. These fast anneals are not currently generally available, but they may be in the future.

6. Additional tips

6.1. Limitations

The shimming techniques we have discussed in this tutorial are efficient to perform and are simple to justify theoretically. However, we note a couple of its limitations here. First, there exists a potential computational bottleneck in the full shimming paradigm. The method partly relies on identifying qubit and coupler orbits. Identifying these orbits can be reduced to the problem of identifying the automorphism group of a graph, which can usually be done quickly in practice even though no general polynomial-time algorithm is known. Second, shims inherently exhibit time-dependent fluctuations due to noise in the environment. These fluctuations tend to be smaller than the shim terms themselves, so do not significantly diminish the potential benefit of calibration refinement.

6.2. Making calibration refinement more efficient

As we have seen, shimming can take many iterations to converge. Naively repeating the process across many combinations of parameters (e.g., annealing time, energy scale, etc.) can be extremely time consuming. However, there are ways to improve the efficiency of the process. Here we outline some important things to bear in mind.

6.2.1. Adjustments are often continuous functions of other parameters

If we determine a set of adjustments for a given experiment, then slightly vary some parameters of the experiment, we can generally expect that the adjustments will not change much. For example, FBOs and coupling adjustments are expected to vary smoothly as functions of annealing time, energy scale, and various perturbations to the system (for example the ratio between FM and AFM couplers in an embedded triangular antiferromagnet). This assumption is natural outside the vicinity of a phase transition, and near a phase transition we adhere to the principle that we should not make discontinuous compensations to a simulator which is itself under smooth parametric modulation. An important example of a smoothly tuned parameter is the annealing parameter s, in cases where we simulate a system at $0 \ll s \ll 1$ (King et al., 2018, 2021a, 2022, etc.).

As an example of how this can help speed up a shim, if we double the annealing time, FBOs and coupling adjustments will remain relatively stable. Thus, rather than starting our shim anew from the nominal Hamiltonian, we can start from an adjusted Hamiltonian that was determined using similar parameters. One could go further than this, and extrapolate or interpolate based on multiple values.

6.2.2. Predictable adjustments should be programmed into the initial Hamiltonian

As shown in Figures 14, 15, starting with halved boundary couplings can immediately bring the couplings close to their converged values. If we are aware of such adjustments, using them as initial conditions can make shims converge far faster.

6.3. Damping shim terms

It is sometimes useful to gently encourage a shim to remain close to the nominal values, for example to prevent drifting Hamiltonian terms. This issue can be particularly important near a phase transition, where statistical fluctuations can be very large. Drift can be suppressed by adding a damping term to the shim. For example, we can set a damping constant $0 \leq \rho \leq 1$, and after every iteration we can move each coupler J_{ij} toward its nominal value $\hat{J}_{ij}$:

$$J_{ij} \to J_{ij} - \rho(J_{ij} - \hat{J}_{ij}). \tag{11}$$

Doing this can discourage random fluctuations, but can also lead to under compensation of biases. It is only recommend to use damping when the shim is otherwise badly behaved. In practice, a suitable value for ρ is determined through trial and error, i.e., by assessing whether shims have converged (see Section 2 for discussion on shim convergence).

7. Conclusions

In this document we have presented several basic examples that introduce the value of calibration refinement or "shimming" in quantum annealing processors. These methods should be applied to any detailed study of quantum systems in a quantum annealer, and will generally provide a significant improvement to the results. Depending on the sensitivity of the system under study, these methods can mean the difference between an unsuccessful experiment and an extremely accurate simulation.

We have provided fully coded examples in Python, which should be easy to generalize and adapt. As part of these examples, we include methods for embedding many copies of a small Ising model in a large quantum annealing processor. This is a valuable and straightforward practice that can enormously improve both the quantity and the quality of results drawn from a single QPU programming.

Another important perspective, which has been introduced here for the first time, is the notion of constructing an auxiliary Ising model and using automorphisms of it to infer qubit and coupler orbits automatically. We encourage users to experiment with this method and report on any challenges or benefits found.

The examples in this document are written for use in an Advantage processor, but are not specific to that model, or even to D-Wave quantum annealers in general. These results may prove useful in diverse analog Ising machines, both quantum and classical.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

AK and JR devised the shimming methods. All authors contributed to the writing and revision of the paper and code examples.

Funding

This tutorial was supported by D-Wave Systems Inc.

Acknowledgments

The authors are grateful to Ciaran McCreesh for help with the Glasgow Subgraph Solver, to Hanjing Xu, Alejandro Lopez-Bezanilla, and Joel Pasvolsky for comments on the manuscript.

Conflict of interest

KC, KB, JR, PF, and AK are employees of D-Wave Systems Inc.

The reviewer AL declared a past co-authorship with the author AK to the handling editor.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2023.1238988/full#supplementary-material

References

Barahona, F. (1982). On the computational complexity of Ising spin glass models. *J Phys A*. 15, 3241–3253.

Boothby, K., Bunyk, P., Raymond, J., and Roy, A. (2020). *Next-Generation Topology of D-Wave Quantum Processors*. Available online at: https://arxiv.org/abs/2003.00133

D-Wave (2022). *D-Wave System Documentation: "Flux-Bias Offsets"*. Available online at: https://docs.dwavesys.com/docs/latest/c_qpu_error_correction.html#qpu-error-fix-fbo (accessed January 3, 2023).

D-Wave (2023). *Minorminer*. Available online at: https://github.com/dwavesystems/minorminer (accessed January 23, 2023).

Godsil, C., and Royle, G. (2001). *Algebraic Graph Theory, Vol 207 of Graduate Texts in Mathematics*. New York, NY: Springer.

Harris, R., Lanting, T., Berkley, A. J., Johansson, J., Johnson, M. W., Bunyk, P., et al. (2009). Compound Josephson-junction coupler for flux qubits with minimal crosstalk. *Phys. Rev. B* 80, e052506. doi: 10.1103/PhysRevB.80.052506

Isakov, S. V., and Moessner, R. (2003). Interplay of quantum and thermal fluctuations in a frustrated magnet. *Phys. Rev. B* 68, 104409. doi: 10.1103/PhysRevB.68.104409

Johnson, M. W., Amin, M. H., Gildert, S., Lanting, T., Hamze, F., Dickson, N., et al. (2011). Quantum annealing with manufactured spins. *Nature* 473, 194–198. doi: 10.1038/nature10012

Kadowaki, T., and Nishimori, H. (1998). Quantum annealing in the transverse Ising model. *Phys. Rev. E* 58, 5355.

Kairys, P., King, A. D., Ozfidan, I., Boothby, K., Raymond, J., Banerjee, A., et al. (2020). Simulating the Shastry-Sutherland ising model using quantum annealing. *PRX Quantum* 1, e020320. doi: 10.1103/PRXQuantum.1.020320

King, A. D., Batista, C. D., Raymond, J., Lanting, T., Ozfidan, I., Poulin-Lamarre, G., et al. (2021a). Quantum annealing simulation of out-of-equilibrium magnetization in a spin-chain compound. *PRX Quantum* 2, e030317. doi: 10.1103/PRXQuantum.2.030317

King, A. D., Carrasquilla, J., Raymond, J., Ozfidan, I., Andriyash, E., Berkley, A., et al. (2018). Observation of topological phenomena in a programmable lattice of 1,800 qubits. *Nature* 560, 456–460. doi: 10.1038/s41586-018-0410-x

King, A. D., Nisoli, C., Dahl, E. D., Poulin-Lamarre, G., and Lopez-Bezanilla, A. (2021b). Qubit spin ice. *Science* 373, 576–580. doi: 10.1126/science.abe2824

King, A. D., Raymond, J., Lanting, T., Harris, R., Zucca, A., Altomare, F., et al. (2023). Quantum critical dynamics in a 5,000-qubit programmable spin glass. *Nature* 617, 61–66. doi: 10.1038/s41586-023-05867-2

King, A. D., Raymond, J., Lanting, T., Isakov, S. V., Mohseni, M., Poulin-Lamarre, G., et al. (2021c). Scaling advantage over path-integral Monte Carlo in quantum simulation of geometrically frustrated magnets. *Nat. Commun.* 12, 1113. doi: 10.1038/s41467-021-20901-5

King, A. D., Suzuki, S., Raymond, J., Zucca, A., Lanting, T., Altomare, F., et al. (2022). Coherent quantum annealing in a programmable 2,000 qubit Ising chain. *Nat. Phys.* 18, 1324–1328. doi: 10.1038/s41567-022-01741-6

Kingma, D. P., and Ba, J. (2014). *Adam: A Method for Stochastic Optimization*. Available online at: https://arxiv.org/abs/1412.6980

McCreesh, C., Prosser, P., and Trimble, J. (2020). "The Glasgow subgraph solver: using constraint programming to tackle hard subgraph isomorphism problem variants." in *Graph Transformation*, eds F. Gadducci and T. Kehrer (Cham: Springer International Publishing), 316–324.

McKay, B. D., and Piperno, A. (2014). Practical graph isomorphism, II. *J. Symbol. Comput.* 60, 94–112. doi: 10.1016/j.jsc.2013.09.003

Moessner, R., and Sondhi, S. L. (2001). Ising models of quantum frustration. *Phys. Rev. B* 63, 1–19. doi: 10.1103/PhysRevB.63.224401

Nishimura, K., Nishimori, H., and Katzgraber, H. G. (2020). Griffiths-McCoy singularity on the diluted Chimera graph: Monte Carlo simulations and experiments on quantum hardware. *Phys. Rev. A* 102, e042403. doi: 10.1103/PhysRevA.102.042403

frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 16 October 2023
DOI 10.3389/fcomp.2023.1286591

OPEN ACCESS

EDITED BY
Nicholas Chancellor,
Durham University, United Kingdom

REVIEWED BY
Jessica Park,
University of York, United Kingdom
Yu He,
Southern University of Science and Technology, China

*CORRESPONDENCE
Ilmo Salmenperä
✉ ilmo.salmenpera@helsinki.fi

RECEIVED 31 August 2023
ACCEPTED 25 September 2023
PUBLISHED 16 October 2023

CITATION
Salmenperä I and Nurminen JK (2023) Software techniques for training restricted Boltzmann machines on size-constrained quantum annealing hardware.
Front. Comput. Sci. 5:1286591.
doi: 10.3389/fcomp.2023.1286591

Software techniques for training restricted Boltzmann machines on size-constrained quantum annealing hardware

Ilmo Salmenperä* and Jukka K. Nurminen

Department of Computer Science, University of Helsinki, Helsinki, Finland

Restricted Boltzmann machines are common machine learning models that can utilize quantum annealing devices in their training processes as quantum samplers. While this approach has shown promise as an alternative to classical sampling methods, the limitations of quantum annealing hardware, such as the number of qubits and the lack of connectivity between the qubits, still pose a barrier to wide-scale adoption. We propose the use of multiple software techniques such as dropout method, passive labeling, and parallelization techniques for addressing these hardware limitations. The study found that using these techniques along with quantum sampling showed comparable results to its classical counterparts in certain contexts, while in others the increased complexity of the sampling process hindered the performance of the trained models. This means that further research into the behavior of quantum sampling needs to be done to apply quantum annealing to training tasks of more complicated RBM models.

KEYWORDS

machine learning, quantum annealing, restricted Boltzmann machines, quantum sampling, dropout method

1. Introduction

When training a well-known machine learning model called restricted Boltzmann machine (RBM), the gradient estimation process for the weights and biases requires the taking samples from a probability distribution called the Boltzmann distribution. While there are classical methods for this process, such as the Contrastive Divergence (CD) algorithm, they are known to grow computationally expensive as the model grows in size (Adachi and Henderson, 2015). An interesting alternative for this classical sampling process is generating these samples using quantum computation devices called quantum annealers (Hauke et al., 2020). While most of the contemporary use cases for these devices are focused on finding low-energy states for quantum systems, these devices have shown promise for sampling data points from the Boltzmann distribution of Hamiltonian energy functions (Adachi and Henderson, 2015; Dixit et al., 2021). This feature of quantum annealing devices has wide applicability in training of classical machine learning models, such as RBM (Restricted Boltzmann Machine) or layer-wise pretraining of more complicated deep learning algorithms. While these models are not on par with the leading industry-level machine learning models, they provide a task where it is quite simple to compare the performance of these quantum techniques with classical techniques, which are of high academic interest.

Quantum sampling have some advantages, such as being faster on large layer sizes or showing improved performance on learning tasks, over the conventional sampling algorithms, such as Gibbs sampling or the contrastive divergence algorithm (Hinton, 2002). These algorithms, especially Gibbs sampling, are relatively slow and do not produce accurate

estimations of the underlying probability distribution (Carreira-Perpiñán and Hinton, 2005). While these algorithms have been deemed good enough for classical use cases, it is still vital to compare them with novel quantum sampling-based approaches to determine whether the switch from classical to quantum can be deemed practical.

The quantum sampling approach does have its own set of issues as follows: (1) The accuracy of the technique is highly dependent on device parameters related to the annealing process, and no known way of determining these parameters exists properly yet; (2) it is not known whether the technique can even produce proper samples from the Boltzmann distribution; and (3) size limitations imposed on the machine learning model by the hardware itself cause the problem space to be limited to toy examples, instead of actually useful real-world problems.

This article will focus mostly on the last issue and proposes and evaluates several techniques to circumvent some obstacles caused by hardware limitations. First of these is the use of extreme rates of unit dropout to reduce the effective layer width of RBMs during the sampling process. The second technique is to use passive labeling schemes to reduce the total width of the visible layer, by disabling all labeling units during the training and adding their influence to the hidden layer as a modifier to the bias of the hidden unit during sampling. Finally, the article will take a look into the inherent parallelism of the quantum annealing device and provide insight into how this technique can have wide use cases on quantum sampling. It is important to note that this last technique does not allow training our models in smaller hardware, but it shows ways that RBMs could take advantage of hypothetical future hardware, especially in tandem with the unit dropout method.

The study shows that while classical methods require fewer epochs for well-behaving models, the end result after a longer period of training can be closely the same, or sometimes even better, which is in line with previous research. The unit dropout method further accentuates this effect and, in our experiments, performs demonstrably worse compared with classical dropout techniques. The reasons for this are analyzed in the Section 7 of the article. The parallelization schemes seem to somewhat lower the performance of the training but decrease the time-to-solution of each round of estimating the model distribution drastically. Finally, the passive labeling strategy shows promise for evaluating the performance of quantum sampling, without any hardware-related costs.

The key contributions of this article are as follows:

- Proposing these techniques for alleviating the presented hardware-related issues and evaluating the effects and the limitations to use a theoretical setting (Section 4).
- Developing an experimental setup to evaluate how these techniques perform when training RBMs against classical methods in similar contexts and showing their benefits and restrictions (Sections 5, 6).
- Providing discussion on the results and how current generation quantum annealing hardware needs to scale to be usable in these sampling tasks (Section 7).

2. Related research

Restricted Boltzmann machine has been studied extensively for a very long time (Hinton and Sejnowski, 1983), but their usefulness has become more apparent in the last decade (Hinton, 2012). Research on classical sampling methods gained traction when the Contrastive Divergence algorithm was discovered, which allowed RBMs to be trained more efficiently compared with the older sampling methods (Carreira-Perpiñán and Hinton, 2005). The dropout algorithm featured in this article has been researched quite extensively, showing improvements on performance and also working as a weight regularization method for many different machine learning models (Srivastava et al., 2014).

The use of quantum annealing in sampling tasks has been researched widely, and it has shown some advantages over classical sampling methods, despite the stated issues. In the study by Adachi and Henderson (2015), quantum annealing was used to pretrain a deep belief network, which showed increased performance over classical sampling methods on a Bars and Stripes dataset. In the study by Dixit et al. (2021) quantum annealing was shown to be as effective as classical sampling methods when training the RBM on a cybersecurity ISCX dataset. Pelofske et al. (2022) presented the technique for parallelizing QUBO problems for quantum annealing devices, which is particularly useful for training RBMs as presented in this study.

There is also a study conducted on a purely quantum version of the more general Boltzmann Machines that are called QBMs (Quantum Boltzmann Machines) (Amin et al., 2018). There are also Quantum Born Machines, which have shown quite a bit of promise in various generative machine learning tasks, that share a lot of their underlying math with Boltzmann Machines (Coyle et al., 2020). It is important to note that these quantum machine learning models are most often implemented in gate-based quantum hardware, as opposed to quantum annealing hardware.

3. Theoretical background

Restricted Boltzmann Machines are simple neural networks that can be applied to various machine learning tasks (Hinton, 2012). In practice, they are mostly used in the pretraining phase of more complex machine learning models such as deep belief networks (Hinton et al., 2006). They are characterized by a visible and a hidden layer of units connected bilaterally, and the units are activated using the sigmoid function.

These models are based on the Ising model: a mathematical representation of ferromagnetic system, where the stochastic behavior of the system is governed by a Hamiltonian energy function E. With this function, the probability P, often also referred to as the Boltzmann distribution, of a system being in a certain configuration can be computed using the following equations:

$$E(\boldsymbol{v},\boldsymbol{h}) = -\sum_i b_i h_i - \sum_i c_i v_i - \sum_{i,j} w_{i,j} v_i h_j \quad (1)$$

$$P(\boldsymbol{v},\boldsymbol{h}) = Z^{-1} e^{E(\boldsymbol{v},\boldsymbol{h})/T} \quad (2)$$

where $\boldsymbol{\sigma}$ is the collection of units in an RBM with possible states $\{0,1\}$. b_i and c_i are the bias values of the hidden and visible units

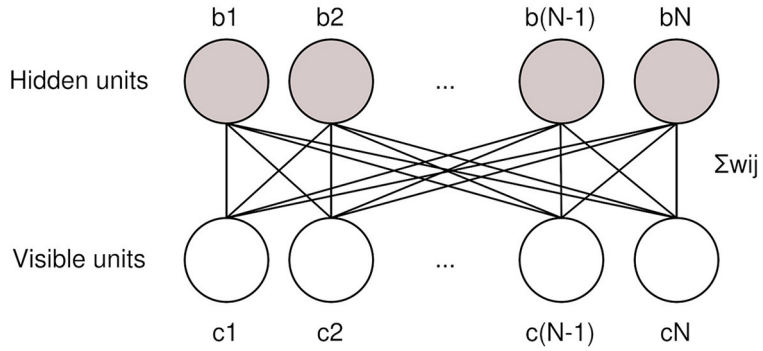


FIGURE 1
Restricted Boltzmann machines have a hidden and a visible layer of units with biases b_i and c_i, connected by weights w_{ij}.

h_i and v_i. $w_{i,j}$ is the weight of the connection between units v_i and h_j. Z is the partition function over all possible combinations of σ that normalizes the probability to be between 0 and 1. T is the temperature of the system, which is often normalized as 1. The structure of a RBM is visualized in the Figure 1.

Training these models requires finding a set of parameters θ, which makes the model distribution P mimic an unknown data distribution Q that characterizes the problem. These parameters can be found by minimizing the Kullback-Leibler divergence between these two distributions, which, in turn, can be approximated by minimizing the average negative log-likelihood of the model distribution (Joyce, 2011; Hinton, 2012). This ultimately results in the following training rules for weights and biases which can be used for gradient descent:

$$\partial w_{ij} = \eta \left(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \right) \quad (3)$$

$$\partial b_i = \eta \left(\langle h_i \rangle_{data} - \langle h_i \rangle_{model} \right) \quad (4)$$

$$\partial c_i = \eta \left(\langle v_i \rangle_{data} - \langle v_i \rangle_{model} \right) \quad (5)$$

where η is the learning rate and $< \ldots >_{data}$ and $< \ldots >_{model}$, respectively, are the data and model distributions of the system.

The important thing to notice here is the fact that estimating the data distribution of an RBM can be done easily using classical algorithms, but estimating the model distribution is considered to be analytically intractable. This is due to the partition function, which requires the algorithm to compute the total energy of the system for all possible configurations of σ. This requires $O(2^n)$ computations where n is the number of units in the system, which means that alternative methods for estimating this distribution are needed.

Instead of computing an exact solution for the model distribution, sampling methods are used to get an estimate of the model distribution. If it is possible to draw accurate samples from the probability distribution P, the average of these samples can form a proper estimate of the model distribution. This is usually done using the Contrastive Divergence (CD) algorithm, where the states of the visible and hidden layers are inferred repeatedly from each other, starting from the initial data vector v assigned to the visible units (Carreira-Perpiñán and Hinton, 2005). The number of cycles in this process can influence the accuracy of the resulting model depending on the problem at hand. Even one iteration has been found to converge toward the correct solution, more iterations can result in improved accuracy of the resulting model (Carreira-Perpiñán and Hinton, 2005). Increasing the number of cycles is a very expensive process, which is why more efficient sampling methods can provide more benefits in tasks that require training RBMs. Contrastive divergence is often marked by appending the number of cycles after the CD abbreviation, i.e., contrastive divergence with one cycle becomes CD-1.

3.1. Sampling from the Boltzmann distribution using quantum annealing

Quantum annealing is a novel alternative to universal quantum computing, where, instead of using gate operations to modify the states of the qubits in the device, it implements a physical system that corresponds to the Ising Model (Hauke et al., 2020). Mathematically, the quantum annealing process implements a Hamiltonian function as follows:

$$H(\tau) = A(\tau)H_D + B(\tau)H_P \quad (6)$$

$$H_D = -\sum_{i \in \mathcal{V}} \sigma_i^x \quad (7)$$

$$H_P = \sum_{ij \in \mathcal{E}} J_{ij} \sigma_i^z \sigma_j^z + \sum_{i \in \mathcal{V}} h_i \sigma_i^z \quad (8)$$

where H_D is the initial Hamiltonian of the system, and H_P is the target Hamiltonian which describes the problem at hand. σ_i^x and σ_i^z are Pauli matrices localized to qubit i, $A(\tau)$ and $B(\tau)$ are time-dependent monotonic functions, which describe the schedule in which H_D is transformed into H_P, when normalized annealing time τ moves from 0 to 1. $J_{i,j}$ and h_i are the parameters that describe the interactions between the qubits of the system.

Quantum annealing devices are capable of finding ground states for Hamiltonian systems, due to the adiabatic theory of quantum mechanics. While this has been contested before, after years of research, it has become quite evident that these devices can be also used to sample from the Boltzmann distribution of the given Hamiltonian (Benedetti et al., 2016). This could benefit the process of training RBMs drastically, as the model distribution of an RBM can be approximated by an average of samples taken from the Boltzmann distribution of the model.

Sampling from the Boltzmann distribution of the model requires small changes to the quantum annealing process. The control parameters of the system have to be scaled down by a parameter called the effective temperature B_{eff}, which allows the system to thermalize more freely during the annealing (Benedetti et al., 2016). Choosing the correct value for this parameter can be difficult, as it seems to be dependent on multiple factors, like the size of the system and the parameters of the system itself. This choice is often done before the training process by evaluating the performance of the parameter against classical methods on similarly sized models and keeping it constant during the training process.

There is also research suggesting that only using B_{eff} to scale the parameters of the model can be insufficient while using alternative annealing schedules provided by current generation quantum annealing devices can help to alleviate these issues (Marshall et al., 2019). For example, pausing the annealing in the middle of the process can improve the accuracy sampling process, provided that the pause happens on a correct region, which is again dependent on the model parameters. The process of reverse annealing has also shown promise for improving the sampling accuracy.

While the sampling capabilities of quantum annealing devices are promising, the limited device sizes and the constraint they impose on the layer sizes of RBMs are still the key limiting factors on applying quantum annealing to machine learning problems (Dumoulin et al., 2013). As the connectivity between qubits is very limited in the current generation quantum annealing devices, embedding fully connected RBMs requires chaining qubits together. This imposes a maximum layer width on the trained RBMs, which is still far away from conventionally used layer sizes, which can have easily over 1,000 units in a single layer. The quantum annealing device DWave 2000Q has a theoretical maximum layer width of 64 units, and while DWave Advantage does not yet have a known theoretical maximum layer width, the modern embedding heuristics are capable of finding embeddings with a layer width of 128 units.

4. Materials and methods

This section describes various methods which can be used to circumvent limitations that arise due to maximum layer sizes imposed by the small qubit counts and the effects of limited topology of current generation quantum annealing hardware.

4.1. The unit dropout method

Unit Dropout is a widely adopted weight regularization method for neural networks, originally developed for RBMs (Srivastava et al., 2014). In this method, during training, units from the model are dropped out with probability p, usually referred to as the dropout rate. It is also possible to keep the amount of dropped-out units constant, in which case we can describe the dropout process using a variable called S_{max}, which is the amount of units kept in the RBM Layer. This process is presented in Figure 2. The training will, then, resume for the pruned network for the duration of a single batch, and the parameter updates will be computed for the pruned network. After this, the units that were dropped out are returned to normal, and the process can repeat until the training has been completed. This has been shown to regularize the weights very efficiently and to be resilient against overfitting during training (Srivastava et al., 2014).

This method is very convenient for the purpose of training Restricted Boltzmann Machines using quantum annealing, as it automatically prunes the model to a smaller subset of the original one. This means that the new model will be easier to fit inside a contemporary quantum annealing device. The dropout rate can also be tweaked to control the size of the model that will be embedded into the quantum annealing device, allowing for a lot of control over the resulting model.

When using this method in tandem with quantum annealing, small modifications need to be made to the original algorithm to take into account the limits imposed by the quantum annealing device. Instead of using a probabilistic dropout rate p, constant S_{max} number of units should be picked from the model with uniform probability. In this way, it is easier to ensure that the model can still be embedded into the device, and it also allows us to reuse the same embedding scheme for the duration of the training, which is useful as computing an embedding scheme for a problem is an expensive process (Cai et al., 2014). If $S_{max}/N_{units} \leq 0.5$, multiple subsets of size S_{max} can be chosen from the units of the model, making the training more efficient, as these models can be sampled in parallel. Existing research places the optimal value for the dropout rate approximately 0.5, but this rate can be pushed further to allow larger layer sizes to be trained using existing quantum annealing devices, as shown in Section 5.

4.2. Passive labeling

While RBMs are often used for unsupervised learning tasks, they are also capable of supervised learning by adding predictive label units to the hidden layer of the network. Because these additional units can be treated as additional visible units in the system, it is often convenient to use different activation functions, like the softmax activation function, for them, as this can improve the predictive capabilities of the network. Though this works quite well for classical sampling algorithms, the core assumption of quantum annealing assumes the likelihood of a unit coming from the Boltzmann distribution. This means that alternative activation functions are not viable for quantum-sampled RBMs.

Adding labeling units into the RBM is useful, as they provide a clear metric for the fitness of the training process, as opposed to measuring the reconstruction error of the model or evaluating the generative capabilities of the model by eye. For quantum sampled RBMs, this can be difficult, as adding labels to the system takes valuable space in the embedding map, increases total chain length of the system, and breaks the symmetry of the total area required by the model. We have developed a novel technique of adding labels to RMBs called the passive labeling technique to address these inconveniencies.

In passive labeling, an average influence of the label units on the hidden units is computed classically before the sampling starts using any activation function. This influence can, then, be added

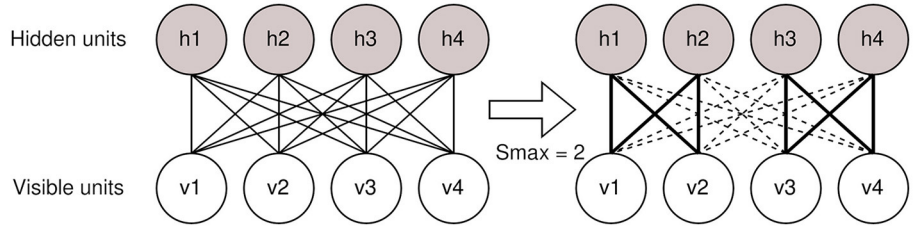


FIGURE 2
Example of the dropout process. Here, the total layer width is 4 units and before sampling half of the units are dropped out from the model. The remaining weights are shown using bolded lines between the units.

to the bias of the hidden unit for the duration of the sampling procedure, while all the labeling units are kept out from the sampling process.

$$h_i^{passive} = h_i + \text{softmax}(\sum_{k \in \mathbf{l}} l_k w_{i,k}^{label}) \quad (9)$$

where $\mathbf{l}$ is the set of labeling units, and $w_{i,k}^{label}$ is the weight associated with the hidden unit h_i and labeling unit l_k.

After the states of the units are sampled, the states of the labeling units can be inferred from the hidden states classically, and these states can be used to compute the parameter gradients of the label units. This should cause only a slight cost for the accuracy of the learning process, with no requirements imposed on the sampling compared with the unsupervised learning methods. If the purpose of the labeling is to evaluate the effectiveness of the sampling techniques for the quantum annealing algorithms, this cost should be more than reasonable, compared with the apparent cost of adding multiple label units to the system.

4.3. Inherent parallelism of quantum annealing

Whenever quantum annealing is used for sampling from systems, it is possible that many of the qubits that are not connected to the embedded model are left unused during the sampling. This is especially wasteful when the problem size is much smaller than the maximum allowed. As shown in the study mentioned in the reference (Pelofske et al., 2022), smaller problems can be embedded into quantum annealing device multiple times, as shown in Figure 3, which reduces the time-to-solution of the problem greatly. This technique is especially interesting for quantum sampling, as using novel annealing control techniques, such as mid-annealing pauses, can increase the overall sampling time by a large margin. This means that the overhead of embedding the problem multiple times into the annealing device will become quite negligible, as the time of taking each sample can increase from the default value of $20\mu s$ to even $1,000\mu s$.

There are two main ways in which this parallelism technique can help in the process of quantum sampling. The first one is reducing the number of samples to $1/N$ of the original size, where N is the number of times in which the problem will fit into the sampling device (Pelofske et al., 2022). The other way is an intersection between using the dropout technique and the inherent parallelism of the quantum annealing device, taking the pruned networks from the dropout process, embedding them all into the quantum annealing device, and producing samples for them in parallel. This method of parallelism should outperform the original one in relation to time, as the time-consuming calls to the quantum sampling device will be reduced to the $1/N$ of the original amount, negating a lot of unnecessary networking overhead while also increasing the amount of work that can be now done in parallel by the classical processes.

While this technique does not address the issue of limited hardware, a reasonable assumption is that if these techniques become viable in future, the growth of the possible hardware will allow us to further take advantage of the computational resources we have. Even on current generation hardware, this technique managed to save a lot of computational resources and time, as shown in the Section 6 of the article.

5. Experimental setup

The techniques presented were evaluated by training restricted Boltzmann machines on a custom-made generated bars and stripes dataset, which is presented in Figure 4. This allowed for strict control over the overall size $N_{problem}$ of the dataset and the difficulty of the machine learning task itself, as a variable amount of noise was introduced to the dataset to make the task more difficult. Using these rules, a labeled training set of 10,000 images, a prediction set of 2,000 images, and an evaluation set of 2,000 images were created. The training dataset is, then, divided into 20 batches for training, and the relatively large batch size was chosen to save computational resources. Two distinct datasets were created for evaluating the different qualities of the algorithm: the 64-pixel dataset and the 256-pixel dataset.

The 64-pixel bars and stripes problem was formulated for testing out how embedding one RBM multiple times into the quantum annealing device compares to embedding it a single time performance-wise. This dataset allowed us to also test how parallel embedding of RBMs affects the performance of the training algorithm.

The 256-pixel bars and stripes problem was formulated for looking into the effects of drastic rates of dropout used in tandem with quantum annealing. Multiple RBMs were trained with various rates of unit dropout, using the CD-1 sampling and quantum

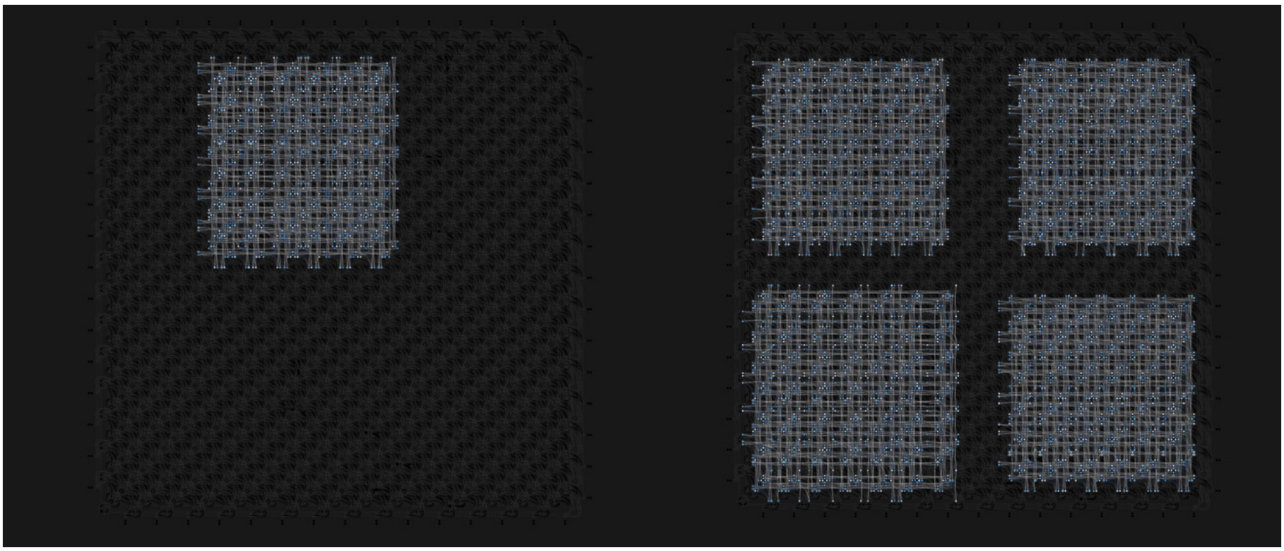


FIGURE 3
Example of problem parallelization on quantum annealing devices: On the **left**, a single 64 × 64 RBM is embedded into the device, leaving many of the qubits unused during sampling. On the **right**, the 64 × 64 RBM has been embedded into the device four times, allowing these RBMs to be sampled simultaneously. These RBMs can be identical or distinct from one another depending on the use case. An image is created using DWave visualization tools.

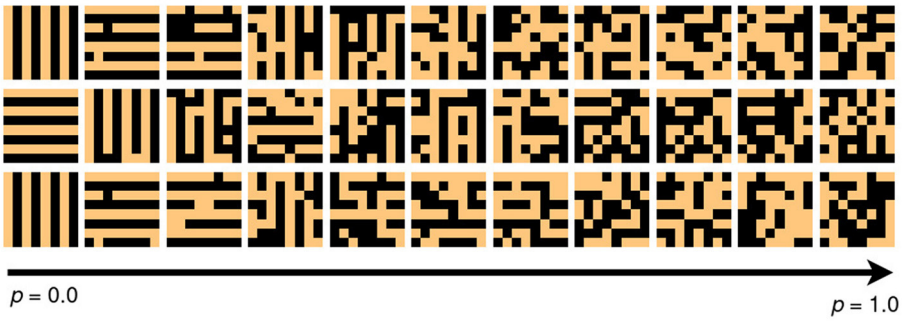


FIGURE 4
Examples 8x8 images generated for testing the learning methods of this study from noise level $p = 0.0$ to $p = 1.0$. The dataset is divided into images of bars (vertical stripes) and stripes (horizontal stripes), after which noise is introduced to the image by randomizing each pixel with the probability of p. The choice of this probability p determines how difficult this learning task will be. These images for the study were created with $p = 0.7$.

annealing. The effects of embedding multiple RBMs into the device and sampling them at the same time were also tested.

The RBM implementation was written in python, and the quantum sampling was implemented using the APIs of the DWave Leap platform and AWS platform. The quantum sampling implementation targeted the DWave Advantage quantum annealing device, for which the embedding schemes were precomputed using the DWave MinorMiner tool (Cai et al., 2014). The parameters of the annealing procedure were chosen manually by evaluating the L1 distance between the gradients of the quantum sampling approach and classical Gibbs sampling with 1,000 cycles. Additional evaluation of techniques was done by classical means.

An effective temperature of 1.0 was chosen for annealing by evaluating the accuracy of the gradient estimation for different values. A pause of $10\mu s$ was introduced in the middle of the annealing process, which improved the sampling accuracy by a sizeable margin. Five spin reversal transforms were used to ensure that the device-specific errors would not affect the learning process that much. Each gradient update was computed from 100 samples taken from the annealer. Finally, the strength of the chain between logically coupled qubits was set to 1, which was essential for achieving well-trained models during the training.

Classical machine learning parameters were chosen by training various models classically and picking the best one for quantum sampling approaches. This was hardly the ideal method for choosing parameters, as there is no guarantee that the ideal classical parameters for the quantum sampling method mirror the classical methods, but as training the models using quantum annealing was very time-consuming and expensive, this way was chosen due to convenience.

The experimental setup was affected by the fact that while the process of sampling from the quantum annealer itself is very

fast, the API calls to the cloud platforms for each of the quantum annealing tasks were very slow, as most of the time was spent on queues waiting to get access to the annealing device.

6. Results

Figure 5 shows results for training multiple RBMs using different sampling approaches. As can be seen, quantum sampling performs similarly or worse than classical sampling approaches. It has to be noted that this performance could be improved with more careful choices for the annealing parameters.

In the 64-pixel dataset, quantum annealing managed to achieve higher prediction rates compared to the classical approach, but the training process required more epochs. This result is in line with previous findings on the performance of the quantum annealing in these sampling tasks (Adachi and Henderson, 2015; Benedetti et al., 2016; Dixit et al., 2021). Initially, the parallel and non-parallel sampling approaches were in line with one another, but in the end, the non-parallel approach outperformed the parallel one. In the classical training case, the passive labeling scheme was completely identical compared to the traditional sampling approach, but in other more complex problems, it showed consistent slight decreases in accuracy. Parallelizing sampling reduced the time that it took to generate the samples from around 98 ms to 84 ms, which is not a huge decrease, but this gap could widen, if advanced annealing control schemes would be used during the annealing.

In the 256-pixel dataset, the classical sampling methods outperformed the quantum ones quite consistently. Only one of the classical sampling methods with the largest rate of the dropout was in line with the quantum sampling approach. The results of these quantum-sampled RBMs also were a lot noisier compared to classical RBMs. The effects of dropout on the prediction rates were quite consistent with existing research for about halfway into the training (Srivastava et al., 2014), as the lower dropout rates seemed to outperform the higher ones until the quantum sampling approaches seemed to converge into the same region of prediction rates. The parallel and non-parallel quantum sampling approaches were again very similar to the 64-pixel dataset until the difference converged in the end. Here the real difference is in the time-to-solution of the parallel and non-parallel sampling methods, which is quite drastic. Taking 100 samples for four different RBMs at the same time would take about 110 ms, compared to about 390 ms when taken subsequently. This does not take into account the overhead from networking-related tasks with the communication with the classical computer and quantum platforms, which could take anywhere from a couple of seconds to a couple of minutes of real-time when conducting this study, further widening the gap between the parallel and non-parallel sampling methods.

7. Discussion

Quantum sampling seems to perform similarly to classical sampling methods in the 64-pixel bars and stripes problem. While classical sampling methods find the well-performing model parameters faster, quantum sampling seems to catch up with the classical methods after some additional training. The prediction rates of quantum sampled RBMs seem to be sometimes more unstable during training, probably due to the noisiness of the gradient estimation. These results indicate that quantum sampling can at least be considered to be a good alternative for estimating the model distribution of a Hamiltonian energy function, to the contemporary classical method of CD-1.

The largest issue with using the quantum sampling approach comes from the larger parameter space, which needs to be controlled for the duration of the training (Benedetti et al., 2016). Device parameters need to be chosen well enough for the training to be effective and there are no known heuristics for choosing them correctly, other than applying some commonly used default values for them. The optimal values for these parameters can be dependent on the embedded problem, which makes constantly evaluating new values for them during the training an intractable task. Finding a heuristic for estimating these parameters could be vital for the commercial viability of the quantum sampling approach.

The passive labeling strategy seemed to perform well in this learning task when comparing the prediction rates for conventional classical training methods and using the passive labeling scheme classically, though its performance can suffer when using it on more complicated machine learning tasks. This means that this method of attaching labels without any increase to the effective size of the embedded problem can be used to evaluate the performance of quantum sampling methods. As most often RBMs are used only for pretraining more complicated deep neural networks like deep belief networks (Hinton et al., 2006), attaching labels this way is probably not needed in industry-level machine learning tasks. It provides a more concrete way of looking into the effectiveness of quantum sampling, compared to reconstruction rate or evaluating generated sampled images out of the network.

The dropout method, when used in tandem with quantum sampling, seems to produce more volatile results as shown in Figure 5B. As both techniques introduce some noise in the gradient estimation process, the resulting quantum sampled models ended up performing worse than their classical counterparts. This could be because of poor parameter choices for many of the quantum annealed RBMs, as the only model that behaved similarly to the classical equivalent was the $S_{max} = 64$ model, which was also incidentally the model size which was used for determining the hyperparameters for the training. It is also possible that the use of the dropout technique is not compatible with these quantum sampling techniques. Further research on the topic of unit dropout and quantum annealing should be done, but this was not possible to do here due to a lack of access to quantum hardware. The key takeaway is that better heuristics for device parameters could allow introducing dropout into quantum sampling in actual use cases.

Parallelizing quantum sampling tasks into the quantum annealing device showed a slight decrease in performance but lowered the time-to-solution of the problem by a good margin. This is especially true when using these parallelization techniques in tandem with the dropout technique, allowing us to sample from all the sub-RBMs at the same time. Likely, the upper limit of the number of different RBMs that could be embedded into the quantum annealing device is two, as the optimal value for the dropout rate dictates that going beyond 0.5 will only hinder the training process. This can still give us results about two times faster than normal, and these two distinct sub-RBMs can

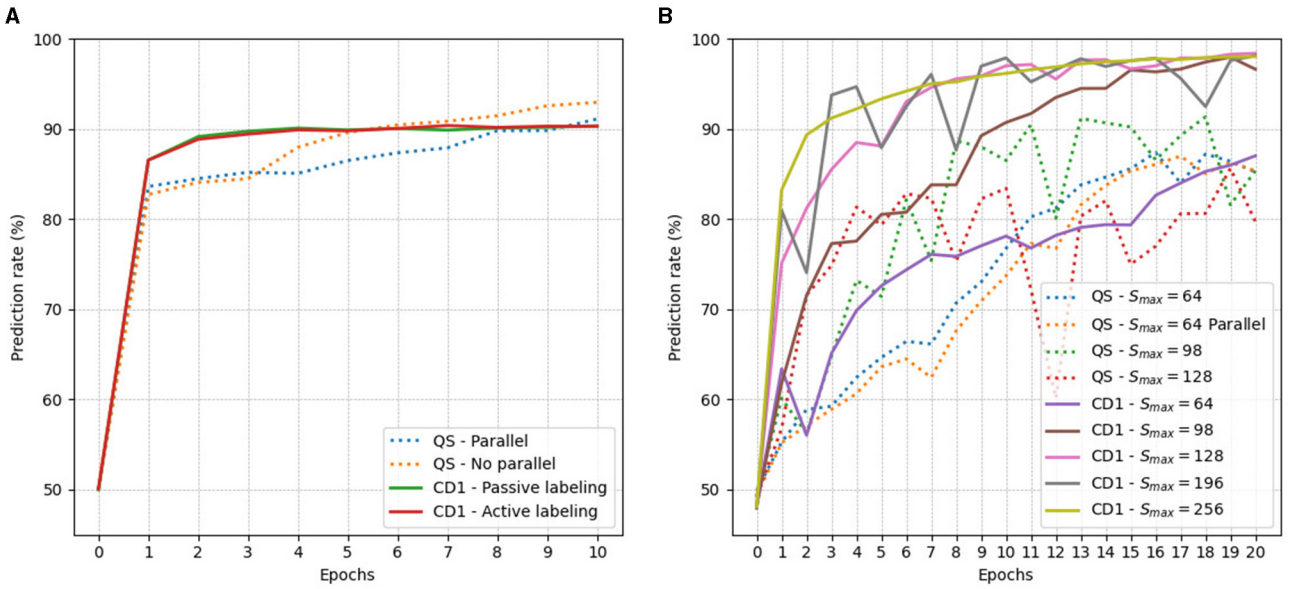


FIGURE 5
Prediction rates of RBMs that have been trained for 20 epochs. QS indicates quantum sampling and CD-1 contrastive divergence with one sampling step **(A)** results of 8 × 8 Bars and Stripes problem trained on RBM with 64 visible and hidden units. **(B)** Results of 16 × 16 Bars and Stripes problem trained on RBM with 256 visible and hidden units with various rates of dropout.

be further parallelized, assuming that the device size itself is large enough.

Quantum annealing devices have already grown quite large from the point of view of qubit counts, and further advances in hardware will bring us new ways quantum computing can be used to benefit existing computational methods. The importance of this study can be seen at two points in time in relation to hardware advancements: (1) in the near term these techniques can be used to train up to two or three times wider networks than normally would be possible due to hardware limitations and (2) in long term these techniques allow for parallelizing the training process of pruned networks, reducing the number of API calls or samples needed for completing quantum sampling tasks. Both of these possibilities are dependent on whether exploring the rather large hyperparameter space of quantum sampled RBMs becomes convenient in the future.

8. Conclusion

While the current generation quantum annealing devices are still quite small in the context of using them for quantum sampling, the industry leader of quantum annealing devices DWave has already envisioned creating larger devices with more advanced connectivity schemes (DWave, 2021). But despite the rapid development of hardware, it is still important to try to bridge the gap between it and the software side, as reaching applicability as early as possible can be vital for adoption on larger scales. It also has to be noted that whether quantum annealing can provide a proper quantum advantage in computational problems is still a highly debated topic (Hauke et al., 2020).

The unit dropout method can be seen as a convenient way of pruning RBM layers into more palatable chunks for next-generation quantum annealing devices, while the parallelization techniques can be used to compute these chunks in parallel on the same annealing device, saving precious computational time, especially on the classical side of things. The passive labeling scheme instead should be thought of as a convenient way of adding labels to RBMs without having to think about their effect on the embedding of the RBM into the hardware itself.

Some possible pitfalls of adopting quantum sampling as a method of evaluating the model distribution function of a Hamiltonian is the increased parameter space caused by the device parameters related to the annealing process. Quite a lot of work shows that choosing the effective temperature of the model can be an intractable task, which is why a lot of research ends up choosing a fiat default value for the duration of the training. Also moving away from the API model of quantum computing to a more integrated model, where the classical computer and the quantum computer work closely together will be vital for any of these speed-ups to matter.

Data availability statement

The original contributions presented in the study are publicly available. This data can be found at: GitHub, https://github.com/Ilmosal/QDBN.

Author contributions

IS: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing—original draft, review, and editing. JN: Funding acquisition, Supervision, Writing—review and editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was partly funded by Business Finland Quantum Computing Campaign project FrameQ (8578/31/2022) and ITEA3 programme project IVVES (ITEA-2019-18022-377).

Acknowledgments

The authors would like to express gratitude to Valter Uotila from the University of Helsinki for providing access to computational resources for the AWS platforms' quantum devices.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Adachi, S., and Henderson, M. (2015). Application of quantum annealing to training of deep neural networks. *arXiv preprint arXiv:1510.06356*. doi: 10.48550/arXiv.1510.06356

Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B., and Melko, R. (2018). Quantum Boltzmann machine. *Phys. Rev. X* 8, 021050. doi: 10.1103/PhysRevX.8.021050

Benedetti, M., Realpe-Gomez, J., Biswas, R., and Perdomo-Ortiz, A. (2016). Estimation of effective temperatures in quantum annealers for sampling applications: a case study with possible applications in deep learning. *Phys. Rev. A* 94. doi: 10.1103/PhysRevA.94.022308

Cai, J., Macready, W., and Roy, A. (2014). A practical heuristic for finding graph minors. *arXiv preprint arXiv:1406.2741*. doi: 10.48550/arXiv.1406.2741

Carreira-Perpiñán, M. Á., and Hinton, G. E. (2005). "On contrastive divergence learning," in *International Conference on Artificial Intelligence and Statistics*.

Coyle, B., Mills, D., Danos, V., and Kashefi, E. (2020). The born supremacy: quantum advantage and training of an ising born machine. *NPJ Quant. Inform.* 6, 60. doi: 10.1038/s41534-020-00288-9

Dixit, V., Selvarajan, R., Aldwairi, T., Koshka, Y., Novotny, M., Humble, T., et al. (2021). Training a quantum annealing based restricted Boltzmann machine on cybersecurity data. *IEEE Trans. Emerg. Top. Comput. Intell.* 6, 417–428.

Dumoulin, V., Goodfellow, I. J., Courville, A., and Bengio, Y. (2013). On the challenges of physical implementations of RBMs. *arXiv preprint arXiv:1312.5258*. doi: 10.1609/aaai.v28i1.8924

DWave (2021). *Zephyr Topology of D-Wave Quantum Processors*.

Hauke, P., Katzgraber, H. G., Lechner, W., Nishimori, H., and Oliver, W. D. (2020). Perspectives of quantum annealing: methods and implementations. *Rep. Prog. Phys.* 83, 054401. doi: 10.1088/1361-6633/ab85b8

Hinton, G., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Comput.* 18, 1527–1554. doi: 10.1162/neco.2006.18.7.1527

Hinton, G., and Sejnowski, T. (1983). "Optimal perceptual inference," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (Washington, DC), 448–453.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Comput.* 14, 1771–1800. doi: 10.1162/089976602760128018

Hinton, G. E. (2012). "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*, eds G. Montavon, K.-R. Müller, and G. B. Orr (Berlin; Heidelberg: Springer), 599–619.

Joyce, J. (2011). *Kullback-Leibler Divergence*. Berlin; Heidelberg: Springer.

Marshall, J., Venturelli, D., Hen, I., and Rieffel, E. G. (2019). Power of pausing: advancing understanding of thermalization in experimental quantum annealers. *Phys. Rev. Appl.* 11, 044083. doi: 10.1103/PhysRevApplied.11.044083

Pelofske, E., Hahn, G., and Djidjev, H. N. (2022). Parallel quantum annealing. *Sci. Rep.* 12, 4499. doi: 10.1038/s41598-022-08394-8

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1929–1958.

frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 19 October 2023
DOI 10.3389/fcomp.2023.1281100

OPEN ACCESS

EDITED BY
Nicholas Chancellor,
Durham University, United Kingdom

REVIEWED BY
Catherine Potts,
D-Wave Systems, Canada
Teng Bian,
Facebook, United States

*CORRESPONDENCE
Phillip Kerger
✉ pkerger@jhu.edu

RECEIVED 21 August 2023
ACCEPTED 02 October 2023
PUBLISHED 19 October 2023

CITATION
Kerger P and Miyazaki R (2023) Quantum image denoising: a framework via Boltzmann machines, QUBO, and quantum annealing. *Front. Comput. Sci.* 5:1281100. doi: 10.3389/fcomp.2023.1281100

Quantum image denoising: a framework via Boltzmann machines, QUBO, and quantum annealing

Phillip Kerger[1,2,3]* and Ryoji Miyazaki[4,5]

[1]Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD, United States, [2]Quantum Artificial Intelligence Laboratory, NASA Ames Research Center, Moffett Field, CA, United States, [3]Research Institute of Advanced Computer Science, Universities Space Research Association (USRA), Moffett Field, CA, United States, [4]Secure System Platform Research Laboratories, NEC Corporation, Kawasaki, Japan, [5]NEC-AIST Quantum Technology Cooperative Research Laboratory, National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan

We investigate a framework for binary image denoising via restricted Boltzmann machines (RBMs) that introduces a denoising objective in quadratic unconstrained binary optimization (QUBO) form well-suited for quantum annealing. The denoising objective is attained by balancing the distribution learned by a trained RBM with a penalty term for derivations from the noisy image. We derive the statistically optimal choice of the penalty parameter assuming the target distribution has been well-approximated, and further suggest an empirically supported modification to make the method robust to that idealistic assumption. We also show under additional assumptions that the denoised images attained by our method are, in expectation, strictly closer to the noise-free images than the noisy images are. While we frame the model as an image denoising model, it can be applied to any binary data. As the QUBO formulation is well-suited for implementation on quantum annealers, we test the model on a D-Wave Advantage machine, and also test on data too large for current quantum annealers by approximating QUBO solutions through classical heuristics.

KEYWORDS

denoising, quantum annealing, machine learning, image processing, quadratic unconstrained binary optimization

1. Introduction

Quantum annealing (QA) (Kadowaki and Nishimori, 1998; Das and Chakrabarti, 2008; Albash and Lidar, 2018) is a promising technology for obtaining good solutions to difficult optimization problems, by making use of quantum interactions to aim to solve Ising or quadratic unconstrained binary optimization (QUBO) instances. Since Ising and QUBO instances are NP-hard, and many other combinatorial optimization problems can be reformulated as Ising or QUBO instances (see e.g., Glover et al., 2018), QA has the potential to become an extremely useful tool for optimization. As the capacities of commercially available quantum annealers continue to improve rapidly, it is of great interest to build models that are well-suited for this emerging technology. Furthermore, QA has promising machine learning applications surrounding Boltzmann Machines (BMs), as both QA and BMs are closely connected to the Boltzmann distribution. Boltzmann Machines are a type of generative artificial neural network that aim to learn the distribution of some training data set by fitting a Boltzmann distribution to the data, as described thoroughly in (Goodfellow et al., 2016, §20). On the other hand, QA aims to produce approximate minimum energy

(maximum likelihood) solutions to a Boltzmann distribution via finding the ground state of the associated Hamiltonian that determines the distribution. Hence, maximum likelihood type problems on BMs are a natural candidate for applying QA in a machine learning framework. We contribute to the goal of furthering useful applications of QA in machine learning in this paper by building an image denoising model particularly well-suited for implementation via QA.

The task of image denoising is a fundamental problem in image processing and machine learning. In any means of collecting images, there is always a chance of some pixels being afflicted by noise that we wish to remove; see e.g., Boyat and Joshi (2015) for a good overview. Accordingly, many classical and data-driven approaches to the image denoising problem have been studied in the literature (Greig et al., 1989; Rudin et al., 1992; Buades et al., 2005; Tang et al., 2012; Cho, 2013). This paper studies a quantum binary image denoising model using Restricted Boltzmann Machines (RBMs henceforth) (Goodfellow et al., 2016, §20.2) that can take advantage of QA by formulating the denoising problem as a QUBO instance. Specifically, given a trained RBM, we introduce a penalty-based denoising scheme that admits a simple QUBO form, for which we derive the statistically optimal penalty parameter as well as a practically-motivated robustness modification. The denoising step only needs to solve a QUBO admitting a bipartite graph representation, and so is well-suited for QA. As QA has also shown promise for training BMs (Adachi and Henderson, 2015; Dixit et al., 2021), our full model lends itself well for denoising images using quantum annealers, and could thus play a role in the their future applications since QA can then be leveraged for *both* the training and denoising steps. The model also shows promise in absence of QA, and our insights presented are not limited to the QA framework, as the QUBO formulation of the denoising problem and its statistical properties we prove may be of independent interest.

The paper is organized as follows. Section 2 gives a summary of background on quantum annealing and Boltzmann Machines. Section 3 describes our main contribution of the image denoising model for QAs, and Section 4 shows some practical results obtained.

Remark 1.1. We frame our work as a binary image denoising method, although the framework does not depend on the data being images, and can be applied to the denoising of any binary data. This is because the framework does not use any spatial relationships between the pixels, and instead treats the image as a flattened vector whose distribution is to be learned. Hence, the denoising scheme can be applied as-is to any other binary data setting.

1.1. Contributions and organization

We provide QUBO-based denoising method for binary images (applicable to general binary data) using restricted Boltzmann machines in Section 3. This is done by formulating the denoising objective in equation 6 by combining the energy function of the distribution learned by the RBM with a (parameterized) penalty term for deviations from a given noisy image. This objective turns out to have an equivalent QUBO formulation, which is shown in claim 1. In Theorem 3.4, we derive the optimal choice for the penalty parameter under the assumption that the true images follow the distribution learned by the RBM, which also recovers the maximum a posteriori estimate per Corollary 3.5, though our model is more flexible, and this flexibility allows for useful practical modifications. Theorem 3.6 shows that the denoising method yields a result that is *strictly* closer (in expectation) to the true image than the noisy image is, under some additional assumptions. Given that these idealistic assumptions won't be met in reality, we propose a robustness modification in Section 3.3 that *improves* performance empirically. In Section 4, as the method lends itself well to quantum annealing, we then implement the method on a D-Wave Advantage 5000-qubit quantum annealer, demonstrating strong empirical performance. Since only small datasets can be tested on the D-Wave machine due to the relatively low number of qubits, we also test the method on a larger dataset, for which we use simulated annealing on a conventional computer in place of quantum annealing to find good solutions the QUBO denoising objective. Though we highlight the method being well-suited for quantum annealers, we emphasize that it may be of independent interest to the machine learning and image processing communities at large.

1.2. Related work

Closely related work of Koshka and Novotny (2021) uses a similar model as ours for the image reconstruction task, also solving QUBO formulations via quantum annelaing. In the reconstruction task, some subset of pixels is unknown (or obscured or missing), and needs to be restored, whereas our work considers denoising, where which pixels are noise-afflicted is unknown. Greig et al. (1989) derives a maximum a posteriori (MAP) estimator for the noise free image as a denoising method in a particular model of binary images that is less general than ours, though we would recover their estimator under a particular choice of our penalty parameter if we were to apply our framework to their model (since we recover MAP in a more general setting). Further, RBMS and quantum annealing have been studied for the classification problem, for instance in Adachi and Henderson (2015) and Krzysztof et al. (2021). Other research in the machine learning communities has also studied handling *label noise*, such as related work in Vahdat (2017), which studies the problem of training models in the presence of noisy labels, whereas our approach is entirely unsupervised (the data need not have any labels to begin with).

2. Background

Quantum Annealers make use of quantum interactions with the primary goal of finding the ground state of Hamiltonian by initializing and then evolving a system of coupled qubits over time (Johnson et al., 2011). In particular, we may view QA as implementing the Ising spin-glass model (Nishimori, 2001) evolving over time. As the QUBO model is equivalent to the Ising model (Glover et al., 2018), and QUBO instances can be efficiently transformed to Ising instances, a QA is well suited to provide good

solutions to QUBO problems. A QUBO cost function, or energy function, takes the form

$$f_Q(x) := \sum_{i,j} Q_{ij} x_i x_j \tag{1}$$

where $x_i \in \{0,1\}$, and Q is a symmetric, real-valued matrix. We will occasionally refer to Q_{ij} as the *weight* between x_i and x_j. QUBO is well-known to be NP hard (Barahona, 1982), and many combinatorial problems can be reformulated as QUBO instances. See Lucas (2014) and Glover et al. (2018) for thorough presentation of QUBO formulations of various problems. A Boltzmann Distribution using the above QUBO as its energy function takes the form

$$P_Q^{model}(x) = \frac{1}{z}\exp\big(-f_Q(x)\big), \tag{2}$$

where z is a normalizing constant. Note that a parameter called inverse temperature has been fixed to unity and is not explicitly shown in the above expression. In this paper, we will focus on making use of Boltzmann Machines, a type of generative neural network that fits a Boltzmann Distribution to the training data via making use of latent variables. Specifically, we consider Restricted Boltzmann Machines (RBMs), which have seen significant success and frequent use in deep probabilistic models (Goodfellow et al., 2016). RBMs consist of an input layer of *visible* nodes, and a layer of latent, or *hidden* nodes, which each have zero intra-group weights. Let $\mathrm{v} \in \{0,1\}^v$ and $\mathrm{h} \in \{0,1\}^h$ denote the visible and hidden nodes, respectively. It will be convenient for us to write $x = (\mathrm{v},\mathrm{h}) \in \{0,1\}^{v+h}$ as their concatenation. The probability distribution represented by a RBM is then

$$P_Q^{model}((\mathrm{v},\mathrm{h})) = \frac{1}{z}\exp\big(-f_Q(\mathrm{v},\mathrm{h})\big) \tag{3}$$

with the restriction that $Q_{ij} = Q_{ji} = 0$ if $i,j \in \{1,\dots,v\}$ or $i,j \in \{v+1,\dots,v+h\}$. Hence, we have the simplified energy function

$$\begin{aligned} f((\mathrm{v},\mathrm{h}),Q) &= \sum_{i=1}^{v+h}\sum_{j=1}^{v+h} 2Q_{ij}(\mathrm{v},\mathrm{h})_i(\mathrm{v},\mathrm{h})_j = \sum_{i=1}^{v}\sum_{j=v+1}^{v+h} Q_{ij}\mathrm{v}_i\mathrm{h}_j \\ &+ \sum_{i=1}^{v} Q_{ii}\mathrm{v}_i^2 + \sum_{i=v+1}^{v+h} Q_{ii}\mathrm{h}_i^2 \\ &= \mathrm{h}^T W \mathrm{v} + b_v^T\mathrm{v} + b_h^T\mathrm{h} =: f_{W,b_v,b_h}(\mathrm{v},\mathrm{h}) \end{aligned} \tag{4}$$

where W is the $v \times h$ matrix consisting of the Q_{ij} weights between the visible and hidden nodes, and b_v and b_h are vectors of the diagonal entries Q_{ii}, $i \in \{1,\dots,v\}$ corresponding to visible nodes, and $Q_{ii}, i \in \{n+1,...,v+h\}$ corresponding to hidden nodes, respectively. We will write the Boltzmann distribution with this energy function as P_{W,b_v,b_h}, noting that this is also P_Q^{model} for the appropriate Q.

It is well-known that RBMs can universally approximate discrete distributions (Goodfellow et al., 2016), making them a powerful model. They are also more easily trained than general Boltzmann Machines, usually through the contrastive divergence algorithm as described in Hinton (2002), or variants thereof.

2.1. Training Boltzmann Machines

We first devote some discussion to the training of RBMs. Subsection 3.1 then describes how to denoise images via QUBO given a well-trained RBM.

Continuing with the notation as in Equation (4), the probability distribution represented by a RBM is

$$P_\theta(\mathrm{v},\mathrm{h}) = \frac{1}{z_\theta}\exp\big(-f_\theta\big).$$

For simplicity, denote $\theta = (W, b_v, b_h)$ as the model parameters henceforth. The normalizing constant z_θ above is

$$z_\theta = \sum_{\mathrm{v}\in\{0,1\}^v}\sum_{\mathrm{h}\in\{0,1\}^h}\exp\big(-f_\theta(\mathrm{v},\mathrm{h})\big)$$

which is becomes intractable quickly even for relatively small values of v and h. The common training approach aims to maximize the log-likelihood of the data. At a high-level, this will be done by approximating gradients and following a stochastic gradient scheme. However, since our data consists only of the visible nodes, we need to work with the marginal distribution of the visible nodes. This is given by

$$P_\theta(\mathrm{v}) = \sum_{\mathrm{h}} P_\theta(\mathrm{v},\mathrm{h}) = \sum_{\mathrm{h}} \frac{\exp\big[-f_\theta(\mathrm{v},\mathrm{h})\big]}{z_\theta}$$

Denote our set training data samples by $V := \{\mathrm{v}^1,...,\mathrm{v}^N\}$. We will use superscripts to indicate training data samples, and reserve subscripts to denote entries of vectors. Then the log-likelihood is given by

$$\begin{aligned} l_\theta(V) &= \sum_{k=1}^{N}\log P_\theta(\mathrm{v}^k) = \sum_{k=1}^{N}\log\sum_{\mathrm{h}} P_\theta(\mathrm{v}^k,\mathrm{h}) \\ &= \left(\sum_k \log\sum_{\mathrm{h}}\exp\Big(-f_\theta(\mathrm{v}^k,\mathrm{h})\Big)\right) - N\cdot\log z_\theta \\ &= \left(\sum_k \log\sum_{\mathrm{h}}\exp\Big(-f_\theta(\mathrm{v}^k,\mathrm{h})\Big)\right) \\ &\quad - N\cdot\log\sum_{\mathrm{v}}\sum_{\mathrm{h}}\exp\big(-f_\theta(\mathrm{v},\mathrm{h})\big) \end{aligned} \tag{5}$$

Now we can calculate the gradient with respect to θ as

$$\begin{aligned} \nabla l_\theta(V) &= \sum_{k=1}^{N}\frac{\sum_{\mathrm{h}}\exp\Big(-f_\theta(\mathrm{v}^k,\mathrm{h})\Big)\nabla(-f_\theta(\mathrm{v}^k,\mathrm{h}))}{\sum_{\mathrm{h}}\exp\big(-f_\theta(\mathrm{v}^k,\mathrm{h})\big)} \\ &\quad - N\cdot\frac{\sum_{\mathrm{v},\mathrm{h}}\exp\big(-f_\theta(\mathrm{v},\mathrm{h})\big)\nabla(-f_\theta(\mathrm{v},\mathrm{h}))}{\sum_{\mathrm{v},\mathrm{h}}\exp\big(-f_\theta(\mathrm{v},\mathrm{h})\big)} \\ &= \sum_{k=1}^{N}\mathbb{E}_{P_\theta(\mathrm{h}|\mathrm{v}^k)}\Big[-\nabla f_\theta(\mathrm{v}^k,\mathrm{h})\Big] - N\cdot\mathbb{E}_{P_\theta(\mathrm{v},\mathrm{h})}\Big[-\nabla f_\theta(\mathrm{v},\mathrm{h})\Big] \\ &= \frac{1}{N}\sum_{k=1}^{N}\mathbb{E}_{P_\theta(\mathrm{h}|\mathrm{v}^k)}\Big[(\mathrm{v}^k)^T\mathrm{h} + \mathrm{v}^k + \mathrm{h}\Big] \\ &\quad - \mathbb{E}_{P_\theta(\mathrm{v},\mathrm{h})}\Big[\mathrm{v}^T\mathrm{h} + \mathrm{v} + \mathrm{h}\Big] \end{aligned}$$

The first term can be computed exactly and efficiently from the data, since the conditional $P_\theta(\mathrm{h}|\mathrm{v})$ admits the simple form $P(\mathrm{h}_j = 1|\mathrm{v}) = logistic(b_h + (\mathrm{v}^T W)_j)$; we refer the interested reader to Goodfellow et al. (2016) or Dixit et al. (2021) and will focus on the second term. Due to its intractability to compute (one would have to sum over all possibilities of v and h), the most promising approach is to approximate it by sampling from $P_\theta(\mathrm{v}, \mathrm{h})$. Classically, this is done via Gibbs sampling as described in Hinton (2002). However, recent research has also investigated using quantum annealers to sample from the relevant Boltzmann distribution, as suggested in Benedetti et al. (2015) and Dixit et al. (2021), which would make QAs useful in the training process since obtaining good Gibbs samples can be expensive. We note that together with our framework, QAs show promise to become useful for both the RBM training and the denoising process in the implementation of our method.

3. Image denoising as quadratic unconstrained binary optimization

This section is devoted to showing how one can naturally frame the image denoising problem as a QUBO instance over a learned Boltzmann Distribution fit to the data.

3.1. Denoising via QUBO

Let us assume we are given a trained Restricted Boltzmann Machine described in Section 2. The model prescribes to each vector $x \in \{0,1\}^{v+h}$ the cost $f_Q(x)$ and corresponding likelihood $P_Q^{model}(x)$ defined in Equations (1) and (3), respectively. We will here make the assumption that P_Q^{model} describes the distribution of our data. Hence, high likelihood vectors in P_Q^{model} correspond to low cost vectors of f_Q. In particular, note that finding the maximum likelihood argument in Equation (2) corresponds to finding a solution to the QUBO instance in Equation (1).

Now, supposing this model, our goal is to reconstruct an image that has been affected by noise. The visible portion of our vector will be considered to be a flattened image with v pixels, black or white corresponding to 0 or 1, respectively, in the binary entries of the vector.

3.1.1. Noise model

We now describe the noise assumptions we will conduct our analysis under.

Definition 3.1. For $x \in \{0,1\}^v$, we define *x afflicted by salt-and-pepper noise of level σ* as the random variable $\tilde{X}_{x,\sigma} := (x+\epsilon) mod 2$, where $\epsilon_i = B_i(p) \sim Bern(\sigma)$, independently.

In other words, a binary image afflicted by salt-and-pepper noise has each pixel independently flipped with probability σ. In particular, we are interested in $\tilde{X}_{X,\sigma}$, where $X \sim P_Q^{model}$, which is the compound random variable obtained by sampling X from the learned distribution of the data and then afflicting it with salt-and-pepper noise. For notational simplicity, will simply write $\tilde{X}$ when the intended subscripts are clear from context.

We remark here that this salt-and-pepper noise model, also sometimes called impulse valued noise, is a natural choice for binary data and can occur in image processing through faulty sensors or pixel elements in cameras; see e.g., Boyat and Joshi (2015) for discussion of noise models in digital image processing. Since the pixels (or binary data entries for non-image binary data) only take the values 0 or 1, individual entries can only be corrupted by the value being flipped. Hence, continuous noise models such as Gaussian noise are not appropriate. Further, since the data we can work with on currently available quantum machines are very small, imposing additional structure on the noise does not seem fitting. However, the related problem of image reconstruction, in which some *known* set of pixels is damaged, is another model appropriate for such data, as studied in Koshka and Novotny (2021). We emphasize that in our noise model, which pixels are affected by noise is random and unknown, leading to the *denoising* problem.

Suppose we are given a realization $\tilde{x} \in \{0,1\}^v$ of $\tilde{X}_{X,\sigma}$. The reconstruction process aims to retrieve this original X using $\tilde{x}$ and the trained model through Q. The approach we will take begins from the intuition that X is likely to be a high-likelihood image that is close to $\tilde{x}$. To enforce this "closeness" to $\tilde{x}$ while searching for higher likelihood images in our model to remove noise, we add to the cost in Equation (1) a penalty for deviations from $\tilde{x}$ to formulate the following natural denoising cost function:

$$f_{Q,\tilde{x},\rho}(x) = f_Q(x) + \rho \sum_{i,j} (x_i - \tilde{x}_i)^2 \tag{6}$$

for some $\rho > 0$ that determines the penalty level. The intuition is that the minimizer of this function for a well-chosen ρ will change a restricted number of pixels to find an image that is similar to the noisy image, but has a lower cost, i.e., higher likelihood, under the model, in hopes of removing the noise.

We show next that this minimizing Equation (6) corresponds to solving a QUBO instance.

Claim 1. Defining $\tilde{Q}^{\rho,\tilde{x}} \in \mathbb{R}^{(v+h)\times(v+h)}$ by setting $\tilde{Q}_{ij}^{\rho,\tilde{x}} = Q_{ij}$ if $i \neq j$ and $\tilde{Q}_{ij}^{\rho,\tilde{x}} = Q_{ii} + \rho(1-2\tilde{x})$ if $i = j$, we have

$$argmin_x f_{Q,\tilde{x},\rho}(x) = argmin_x f_{\tilde{Q}^{\rho,\tilde{x}}}(x). \tag{7}$$

Proof.

$$\begin{aligned} f_{Q,\tilde{x},\rho}(x) &= f_Q(x) + \rho \sum_i (x_i - \tilde{x}_i)^2 = \sum_{i,j} Q_{ij} x_i x_j \\ &\quad + \rho \sum_i x_i^2 - 2x_i\tilde{x}_i + \tilde{x}_i^2 \\ &= \sum_{i\neq j} Q_{ij} x_i x_j + \sum_i Q_{ii} x_i^2 + \rho(x_i^2 - 2x_i^2\tilde{x}_i + \tilde{x}^2) \\ &= \sum_{i\neq j} Q_{ij} x_i x_j + \sum_i (Q_{ii} + \rho(1-2\tilde{x}_i)) x_i^2 + \rho\tilde{x}_i^2 \\ &= f_{\tilde{Q}^{\rho,\tilde{x}}}(x) + \sum_i \rho\tilde{x}_i \end{aligned}$$

Noting that $x_i = x_i^2$ for the above derivation since they are in $\{0,1\}$ here. Since the $\tilde{x}_i$ terms do not depend on x, the claim follows.

Hence, solving the QUBO in on the right hand side of Equation (7) gives us the solution to Equation (6). Claim Equation 1 thus

tells us that we simply need to modify the diagonal of the original matrix Q of our model by adding $diag(1-2\tilde{x}_1, ..., 1-2\tilde{x}_n)$ and then solve the resulting QUBO to get the denoised image. We can then make use of quantum annealing to solve the resulting QUBO of 7, or use classical methods and heuristics like simulated annealing instead. We formally spell out the denoising procedure in algorithm `QUBO_Denoise`.

`QUBO_Denoise`
Input: A matrix Q, a noisy image $\tilde{x}$ sampled from the distribution of $\tilde{X}_{X,\sigma}$ with $X \sim P_Q^{model}$, and a penalty parameter $\rho > 0$.
Output: A denoised image $X^*_{\rho,\tilde{x},Q}$.

1. Set $\tilde{Q}_{ij}^{\rho,\tilde{x}} = Q_{ij}$ if $i \neq j$ and $\tilde{Q}_{ij}^{\rho,\tilde{x}} = Q_{ii} + \rho(1-2\tilde{x})$ if $i = j$.
2. Set $X^*_{\rho,\tilde{x},Q} := argmin_x f_{\tilde{Q}^{\rho,\tilde{x}}}(x)$.

For the remainder of the paper, $X^*_{\rho,\tilde{x},Q}$ will denote the denoised image obtained by applying `QUBO_Denoise` with noisy image $\tilde{x}$, penalty parameter ρ, and the distribution-defining matrix Q.

Remark 3.2. Considering the entire process of sampling a noisy image and then denoising it, the measurability of $X^*_{\rho,\tilde{X}_{X,\sigma},Q}$ is inherited from the measurability of $\tilde{X}_{X,\sigma}$, which in turn inherits its measurability as compound random variable of the measurable noise and original image $X \sim P_Q^{model}$.

3.2. Optimal choice of penalty parameter ρ

The choice of the parameter ρ for the proposed image denoising model is clearly crucial to its success, since different choices will result in different solutions. If ρ is chosen to be too small, there is very little cost to flipping a pixel, and then many pixels may be flipped and the solution may not resemble the noisy image at all anymore. If ρ is too large, we may be too heavily penalizing flipping pixels, and thus may not be able to get rid of noise effectively. Hence, we now turn toward finding the optimal choice for ρ. We will evaluate the choice of ρ via *expected overlap*:

Definition 3.3. The *expected overlap* between two distributions P and a P', is defined by

$$d(P,P') := \mathbb{E}_P\mathbb{E}_{P'}\left[n - \left\|X - X'\right\|_1\right],$$

where $X \sim P, X' \sim P'$.

We will consider $X \sim P_Q^{model}$, and X' as $X^*_{\rho,\tilde{X}_{X,\sigma},Q}$ the corresponding denoised image, and will also call $d(P,P')$ the *expected overlap between X and X'*. To keep notation simple, for the remainder of this section allow us to write $\tilde{X}$ in place of $\tilde{X}_{X,\sigma}$, with X and σ being clear from context.

Our main positive result concerning the choice of ρ is summarized in the following theorem:

Theorem 3.4. Let $X \sim P_Q^{model}$ as in Equation (2) and $\tilde{X}$ be the noisy image. Then choosing $\rho = log\frac{1-\sigma}{\sigma}$ to obtain $X^*_{\rho,\tilde{X},Q}$ is optimal with respect to maximizing the expected overlap between X and $X^*_{\rho,\tilde{X},Q}$.

Proof. Let X dist P_Q^{model}, and $\tilde{X}$ be X afflicted by salt-and-pepper noise of level σ. Then since $\tilde{X}_{X,\sigma}$ is obtained by flipping pixels with probability σ, we have the conditional probability

$$\begin{aligned} P_\sigma(\tilde{X} = \tilde{x}|X = x) &= \prod_{i=1}^{v}\left\{\sigma(\tilde{x}_i - x_i)^2 + (1-\sigma)[1-(\tilde{x}_i - x_i)^2]\right\} \\ &= \frac{\exp\left[-\beta_\sigma \sum_{i=1}^{v}(\tilde{x}_i - x_i)^2\right]}{(1+e^{-\beta_\sigma})^v}, \end{aligned} \tag{8}$$

where $\beta_\sigma := \log\frac{1-\sigma}{\sigma}$. In order to infer the original image X from the noisy one $\tilde{X}$, we utilize the Bayes formula and calculate the conditional probability $P^{\text{post}}_{\beta_\sigma,Q}(X = x|\tilde{X} = \tilde{x})$.

$$\begin{aligned} P^{\text{post}}_{\beta_\sigma,Q}(x|\tilde{x}) &= \frac{P_\sigma(\tilde{X} = \tilde{x}|X = x)P_Q^{model}(x)}{\sum_{\{x\}} P_\sigma(\tilde{x}|x)P_Q^{model}(x)} \\ &= \frac{\exp\left[-\beta_\sigma \sum_{i=1}^{v}(\tilde{x}_i - x_i)^2 - \sum_{i,j=1}^{v+h} Q_{ij}x_ix_j\right]}{\sum_{\{x\}}\exp\left[-\beta_\sigma \sum_{i=1}^{v}(\tilde{x}_i - x_i)^2 - \sum_{i,j=1}^{v+h} Q_{ij}x_ix_j\right]}. \end{aligned} \tag{9}$$

Note that x includes pixels for hidden nodes, which is fine here. Our approach finds the state which is most likely under this distribution, which is realized by annealing for the above QUBO with the β_σ term.

The overlap of two vectors x^* and x is given by

$$m(x,x^*) := \frac{1}{v+h}\sum_{i=1}^{v+h}(2x_i - 1)(2x_i^* - 1), \tag{10}$$

the proportion of shared entries. We consider the average (over the noise) of solutions, $\bar{X}_{\rho,\tilde{x},Q}$ with

$$(\bar{X}_{\rho,\tilde{x},Q})_i = \theta\left(\sum_{\{x\}} P_{\tilde{Q}}^{model}(x)x_i - \frac{1}{2}\right), \tag{11}$$

where $\theta(x) = 1$ if $x > 0$, otherwise 0, noting that the right hand side represents the inferred pixel value based on the expectation from $P_{\tilde{Q}}^{model}$. We have formally distinguished $P_{\tilde{Q}}^{model}(x)$ from $P^{\text{post}}_{\rho,Q}(x|\tilde{x})$, but in fact they are the same. Note that

$$2(\bar{X}_{\rho,\tilde{x},Q})_i - 1 = \text{sign}\left(\sum_{\{x\}} P_{\tilde{Q}}^{model}(x)(2x_i - 1)\right), \tag{12}$$

where sign(x) is the sign of x. Let $\alpha_{\sigma,Q} := -\beta_\sigma \sum_i(\tilde{x}_i - x_i)^2 - \sum_{i,j} Q_{ij}x_ix_j$ for conciseness. In order to evaluate the statistical performance of our method with coefficient ρ of penalty term, we calculate the average of overlap as

$$\begin{aligned} M_{\beta_\sigma,Q}(\rho) &:= \sum_{\{\tilde{x}\},\{x\}} P_\sigma(\tilde{x}|x)P_Q^{model}(x)m(\bar{X}_{\rho,\tilde{x},Q}), x) \\ &= \frac{1}{(1+e^{\beta_\sigma})^v}\frac{1}{z}\frac{1}{v+h}\sum_i\sum_{\{\tilde{x}\},\{x\}} e^{\alpha_{\sigma,Q}}[2(\bar{X}_{\rho,\tilde{x},Q})_i - 1] \\ &\quad (2x_i - 1). \end{aligned} \tag{13}$$

A sum in the right hand side of the above equation holds

$$\begin{aligned}
&\sum_{\{x\}} e^{\alpha_{\sigma,Q}}[2(\mathbb{E}(X^*_{\rho,\tilde{x},Q})_i - 1](2x_i - 1) \\
&\quad \leq \left| \sum_{\{x\}} e^{\alpha_{\sigma,Q}}[2(\mathbb{E}(X^*_{\rho,\tilde{x},Q})_i - 1](2x_i - 1) \right| \\
&\leq \left| \sum_{\{x\}} e^{\alpha_{\sigma,Q}}(2x_i - 1) \right| = \sum_{\{x\}} e^{\alpha_{\sigma,Q}}(2x_i - 1) \\
&\quad \frac{\sum_{\{x'\}} e^{-\beta_\sigma \sum_i (\tilde{x}_i - x'_i)^2 - \sum_{i,j} Q_{ij} x'_i x'_j}(2x'_i - 1)}{\left| \sum_{\{x'\}} e^{-\beta_\sigma \sum_i (\tilde{x}_i - x'_i)^2 - \sum_{i,j} Q_{ij} x'_i x'_j}(2x'_i - 1) \right|} \\
&= \sum_{\{x\}} e^{\alpha_{\sigma,Q}}(2x_i - 1)\text{sign}\left(\sum_{\{x'\}} P^{model}_{\tilde{Q}}(x')(2x'_i - 1) \right) \\
&\quad = \sum_{\{x\}} e^{\alpha_{\sigma,Q}}[2(\bar{X}_{\rho,\tilde{x},Q})_i - 1](2x_i - 1).
\end{aligned} \tag{14}$$

Hence, the averaged overlap holds

$$\begin{aligned}
M_{\beta_\sigma,Q}(\rho) \leq & \frac{1}{(1+e^{\beta_\sigma})^v} \frac{1}{Z_{1,Q}} \frac{1}{v+h} \sum_i \sum_{\{\tilde{x}\},\{x\}} \\
& e^{-\beta_\sigma \sum_i (\tilde{x}_i - x_i)^2 - \sum_{i,j} Q_{ij} x_i x_j}[2(\bar{X}_{\rho,\tilde{x},Q})_i - 1](2x_i - 1) \\
= & M_{\beta_\sigma,Q}(\beta_\sigma).
\end{aligned} \tag{15}$$

This inequality means that the averaged overlap is maximized when $\rho = \beta_\sigma = \log \frac{1-\sigma}{\sigma}$.

This theorem is based on a known fact in statistical physics of information processing (Nishimori, 2001) and translates the fact into the setting of our problem. Notably, the optimal choice of ρ does *not* depend on the distribution of the data, but only on the noise level, for which in many real world cases one may have good estimates. The proof of the theorem also reveals the following corollary:

Corollary 3.5. Under the same assumptions of Theorem 3.4, setting $\rho := \log \frac{1-\sigma}{\sigma}$ makes $X^*_{\rho,\tilde{X},Q}$ the maximum a posteriori estimator for the original noise-free image X.

The corollary follows from observing that the energy function in the numerator of the posterior distribution Equation (9) is exactly Equation (6) with $\rho := \frac{1-\sigma}{\sigma}$, noting that minimizing Equation (6) is equivalent to maximizing Equation (9). However, this framework allows for additional flexibility in choosing the ρ parameter that is absent in standard MAP estimation. In fact, in Sections 3.3 and 4.1 we go on to demonstrate that in practice, choosing a larger ρ may be beneficial for robustness of the method.

Though Theorem 3.4 derives the optimal choice of ρ, it does not give any guarantees that the method will yield an improvement in expected overlap, even under its assumptions. Next, we prove a theorem to show that in the case of visible units being independent of one another, our image denoising method produces in expectation *strict* denoising improvements with respect to the expected overlap. For $c > 0$ and a model distribution P^{model}_Q as in Equation 2, let $\mathcal{I}_c$ be the set of indices i such that $|Q_{ii}| > c$. These indices correspond to components of X that are either 0 or 1 with probability at least $\frac{1}{1+e^{-c}}$, depending on whether Q_{ii} is positive or negative, respectively.

Theorem 3.6. Suppose that Q is diagonal, $X \sim P_Q$, and that $\tilde{X}$ is X afflicted by salt-and-pepper noise of level σ. With $\mathcal{I}_c$ as defined above for $c > 0$, setting $\rho \geq \log\left(\frac{1-\sigma}{\sigma}\right)$, and assuming that $\mathcal{I}_\rho \neq \emptyset$, the expected overlap of the denoised image and the true image is strictly larger than the expected overlap of the noisy image and the true image, i.e.,

$$\mathbb{E}\left[\sum \mathbb{I}((X^*_{\rho,\tilde{X},Q})_i = X_i)\right] > \mathbb{E}\left[\sum \mathbb{I}(\tilde{X}_i = X_i)\right]. \tag{16}$$

Proof. Let $\mathcal{I}^0_c := \{i \in \mathcal{I}_c : Q_{ii} > 0\}, \mathcal{I}^1_c := \{i \in \mathcal{I}_c : Q_{ii} < 0\}$. Intuitively, these are the indices which are likely to be zero or one, respectively. Further, letting $x^{\dagger i}$ denote the vector obtained by flipping entry i of x, we have that $|f_Q(x) - f_Q(x^{\dagger i})| = Q_{ii} > c$ if and only if $i \in \mathcal{I}_c$. Hence, this reveals that x^* solves Equation (6) by setting $x^*_i = 1 \ \forall i \in \mathcal{I}^1_\rho, x^*_i = 0 \ \forall i \in \mathcal{I}^0_\rho$, and $x^*_i = \tilde{x}_i$ otherwise, since the value of f_Q of Equation (1) is reduced by more than ρ, so that the overall penalized objective Equation (6) improves despite the ρ penalty accrued by the pixel flips.

Now, let $X \sim P^{model}_Q$. Let us compute $P((X^*_{\rho,\tilde{X},Q})_i = X_i)$. The cases where this happens are: $i \in I^0_\rho$ and $X_i = 0$, $i \in I^1_\rho$ and $X_i = 1$, or $i \notin I_\rho$ and pixel i was not flipped by the noise.

We know that if $i \in \mathcal{I}^b_\rho, P(X_i = b) \geq \frac{1}{1+e^{-\rho}}$, for $b \in \{0,1\}$, so $P((X^*_{\rho,\tilde{X},Q})_i = X_i) \geq \frac{1}{1+e^{-\rho}}$ for these. For $i \notin \mathcal{I}_\rho, P((X^*_{\rho,\tilde{X},Q})_i = X_i) = 1 - \sigma$, where σ is the probability that the pixel was flipped by the noise. On the other hand, $P(\tilde{X}_i = X_i) = 1 - \sigma \ \forall i$. We characterize

$$\mathbb{E}\left[\sum \mathbb{I}((X^*_{\rho,\tilde{X},Q})_i = X_i)\right] > \mathbb{E}\left[\sum \mathbb{I}(\tilde{X}_i = X_i)\right] \tag{17}$$

$$\sum P((X^*_{\rho,\tilde{X},Q})_i = X_i) > \sum P(\tilde{X}_i = X_i) = n \cdot (1 - \sigma) \tag{18}$$

For the left-hand side, assuming $\mathcal{I}_\rho \neq \emptyset$, we have

$$\begin{aligned}
\sum P((X^*_{\rho,\tilde{X},Q})_i = X_i) &> \sum_{i \in \mathcal{I}_\rho} \frac{1}{1+e^{-\rho}} + \sum_{i \notin \mathcal{I}_\rho} (1 - \sigma) \\
&= |\mathcal{I}_\rho| \cdot \frac{1}{1+e^{-\rho}} + (n - |\mathcal{I}_\rho|)(1 - \sigma)
\end{aligned}$$

so that Equation (17) holds when

$$\begin{aligned}
&|\mathcal{I}_\rho| \cdot \frac{1}{1+e^{-\rho}} + (n - |\mathcal{I}_\rho|)(1 - \sigma) \geq n(1 - \sigma) \\
&\iff |\mathcal{I}_\rho| \neq 0 \\
&\text{and } \frac{1}{1+e^{-\rho}} \geq 1 - \sigma \iff \rho \geq log(\frac{1-\sigma}{\sigma}) \text{ and } \mathcal{I}_\rho \neq \emptyset,
\end{aligned} \tag{19}$$

and the theorem is proven.

The assumption that matrix Q is diagonal is equivalent to the components of X being independent, which is not realistic with real data. However, since in the RBM model the visible units are independent conditioned on the hidden units, we still consider

this independent case to be informative to the denoising method. In fact, if the hidden states were fixed (or known, or recovered correctly), Theorem 3.6 would apply. We leave it as a tantalizing open question to generalize this result beyond the independent case. The assumption of nonemptiness of $\mathcal{I}_\rho$ is a natural one for the denoising task; indeed, when $\mathcal{I}_\rho$ is empty, no entries of Q are large in magnitude, which is equivalent to the entries of X being close to uniformly distributed. In that case, intuitively of course it should not be possible to guarantee that we can denoise an image well if it looks like noise to begin with.

3.3. Robust choice of ρ

The optimal choice of ρ as derived in Theorem 3.4 relies on the assumption that the observed data comes from the learned distribution, or equivalently that the distribution generating our data has been perfectly learned by the RBM. However, in practice we will always only approximately learn the data distribution. Hence, we do not want to rely too heavily on the exact distribution we have learned when we denoise the images. One may hope to have a more robust method by only changing the value of a pixel when there is some confidence in the model that the pixel should be flipped. We may thus want to penalize flipping pixels slightly more than we should under the idealistic setting of Theorem 3.4, which corresponds to choosing a larger ρ value than $\log \frac{1-\sigma}{\sigma}$, or equivalently using a smaller $\sigma' < \sigma$ value when setting $\rho := \log \frac{1-\sigma'}{\sigma'}$. We opt for the latter as a means of intentionally biasing ρ to make the approach more robust for application. Figures 2, 3 in Section 4 show the effect this proposed robustness modification has, demonstrating indeed that choosing a larger ρ via intentionally using a smaller σ yields positive results. If the true noise level is σ, our experiments demonstrate that setting to roughly $\rho := \frac{1-0.75\sigma}{0.75\sigma}$ has a positive effect on performance.

4. Empirical results

This section contains results from implementing the previously described method and comparing it against other denoising approaches. Datasets and code are available on the first author's GitHub[1] for the purpose of easy reproducibility.

4.1. Datasets and setup

In this subsection, we present empirical results obtained by implementing our model on a quantum annealer, D-Wave's Advantage_system4.1, which has 5,000 qubits and enables embedding of a complete bipartite graph of size 172 × 172. Hence, we use 12 × 12 pixel images here so that the visible layer is of size 144. We test the method on two different datasets with very differently structured data.

The first dataset is a 12 × 12 version of the well-known MNIST dataset (LeCun et al., 2010), created by downsizing the original dataset with nearest-neighbor image downscaling and binarizing pixels. The second dataset we use is a 12 × 12 pixel Bars-and-Stripes (BAS) dataset, as has been used in closely related work (Dixit et al., 2021; Koshka and Novotny, 2021), in which the authors used a smaller 8 × 8 version of BAS in order to accommodate a 2,000 qubit machine, so we implement a larger 12 × 12 version for the 5,000 qubit machine we use. Each image consists of binary pixels with either each row or each column sharing the same values, so that each image consists of either "bars" or "stripes". Some examples of noise-free, noisy, and denoised images across different noise levels are presented in Figure 1.

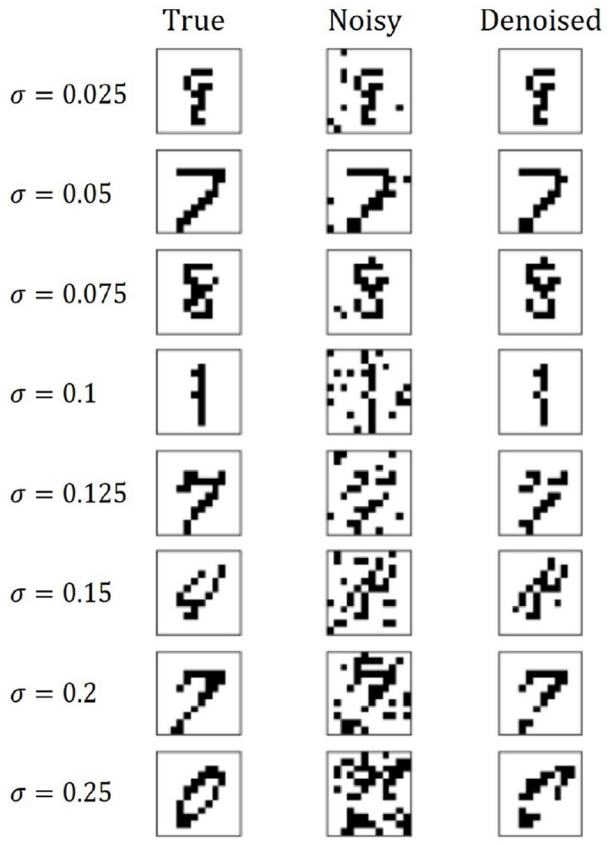


FIGURE 1
Examples of the denoising process using our method showing the true, noisy, and denoised images across different noise levels.

For both datasets we train the RBM by using the classical Contrastive Divergence algorithm first presented in Hinton (2002), and as described in Section 2.1. The number of hidden units was set to 50 and 64 for BAS and MNIST, respectively. For both datasets, we used learning rate of 0.01, batch size of 50, and 150 epochs as the training hyperparameters. For the BAS data, 4,000 images were generated as training data, and 1,000 as test data, while for MNIST, we simply used the full MNIST provided training set of 60,000 images and test set of 10,000 images. Noisy images were generated by adding salt-and-pepper noise of level σ to images from the test dataset. Given a noisy image, we are then able to embed and solve the resulting denoising QUBO of 7 onto a D-Wave quantum annealer, Advantage_system4.1. A function of D-Wave's Ocean software, find_embedding, is utilized to find appropriate

1 https://github.com/PhillipKerger

mappings from variables in a QUBO to physical qubits on D-Wave's Pegasus graph. A variable in QUBO is often mapped to multiple physical qubits, called chain, that are strongly connected to each other to behave like a single variable. A mapping can be used for every noisy images for each dataset, since their QUBO have the same graph structure. We have prepared in advance 50 sets of the different mappings for each dataset and choose a mapping from the pool at random to embed QUBO of each image. This random selection is done to avoid possible artificial effects on the denoising performance from using only a particular mapping. Parameters for embedding and annealing, i.e., chain_strength and annealing_time, are tuned to maximize the performance. In particular, we set chain_strength as the product of a coefficient c_0 and the maximum abstract value among the elements of each QUBO matrix, where we tune c_0. The adopted values of the parameters are different between MNIST and BAS but the same values for all the range of σ. We set (c_0, annealing_time) = (0.6, 50 μs), (0.5, 40 μs) for BAS and MNIST, respectively. The number num_reads of reads of annealing is 100 for each noisy image. We calculate the average of solution of each pixel over the reads to approximate Equation (11) and use it to evaluate the overlap that is proportion of pixels in denoised images that matched the original image. We denoise 200 noisy images for each σ, which are randomly selected from the pool of test images for each sigma. Note also that for each value of sigma, the different methods compared use the same set of (randomly selected) noisy test images.

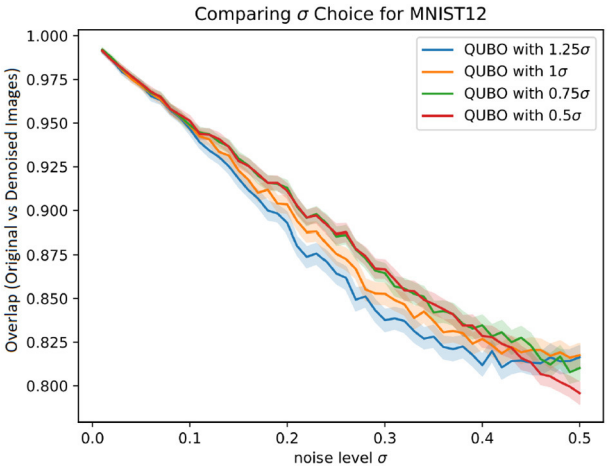


FIGURE 2
Proportion of pixels in denoised MNIST images that matched the original image, for different denoising methods with 95% CI error bars.

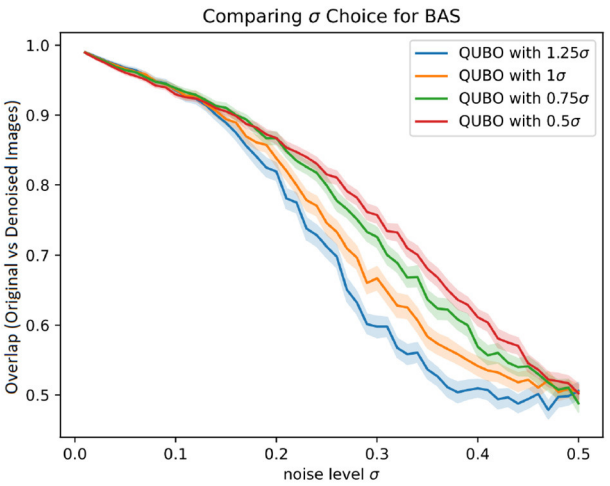


FIGURE 3
Proportion of pixels in denoised BAS images that matched the original image, for different denoising methods with 95% CI error bars.

4.2. Results with quantum annealing

Figures 2, 3 first investigate the robust choice of ρ as discussed in Section 3.3. This is done by using a biased value of σ when setting $\rho = \log \frac{1-\sigma}{\sigma}$, instead setting $\rho := \log \frac{1-b\sigma}{b\sigma}$ for some bias factor b. The denoising performance for $b \in \{1.25, 1, 0.75, 0.5\}$ are shown, with 95% confidence intervals obtained by bootstrapping. Note that using a bias factor $b = 1$ means using the true value of σ for determining ρ.

Based on the empirical performance, using a bias factor of around 0.75 seems to give an improved performance compared to using a bias factor of 1 in both data sets. A bias factor of 0.5 seems to perform quite well-across most noise regimes as well, with largely overlapping confidence regions to the 0.75 parameter setting, though in the low-noise setting for the BAS dataset we observe an adverse effect. The authors thus suggest a setting of 0.75 for the bias factor.

Next, in Figures 4, 5, we compare our method to popular other denoising methods for binary images on the 12×12 MNIST and bars-and-stripes datasets, respectively, across different noise levels. When comparing to other methods, a crucial factor is that we choose ρ based off of σ, but in practice σ may be unknown. In light of this, we include two versions of our method in these comparisons. First, we use our method with $\rho := \log \frac{1-\sigma}{\sigma}$, using the true value of σ without introducing the recommended bias factor. Secondly, we simulate the situation in which the true σ is unknown, and instead we only have a guess for σ. To simulate having an approximate guess for σ, for each image afflicted by noise of level σ, we sample σ' uniformly from an interval of size $\sigma/2$ centered at sigma. We then set $\rho := \log \frac{1-0.75\sigma'}{0.75\sigma'}$, using a bias factor of 0.75 on with this "guessed" value of σ. This is a significantly more realistic way of testing our method, since it gives an idea of how well the method may perform when the true noise level present in the noisy images is unknown and must be guessed. Our implementation here only assumes that the practitioner roughly knows the magnitude of the noise. For example, if the true noise is $\sigma = 0.2$, here we sample σ' uniformly from $[0.15, 0.25]$ to simulate the guess.

We compare our method to Gibbs denoising with an RBM (Tang et al., 2012, Section 3.2), median filtering (Huang et al., 1979), Gaussian filtering (Stockman and Shapiro, 2001, Chapter 5), and a graph-cut method (Greig et al., 1989) for denoising. For the Gibbs denoising, we use the same well-trained RBM as

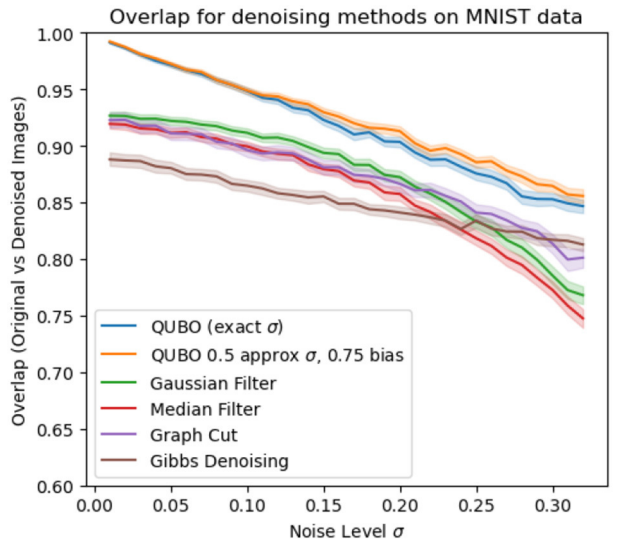


FIGURE 4
Proportion of pixels in denoised MNIST images that matched the original image, for different denoising methods with 95% CI error bars.

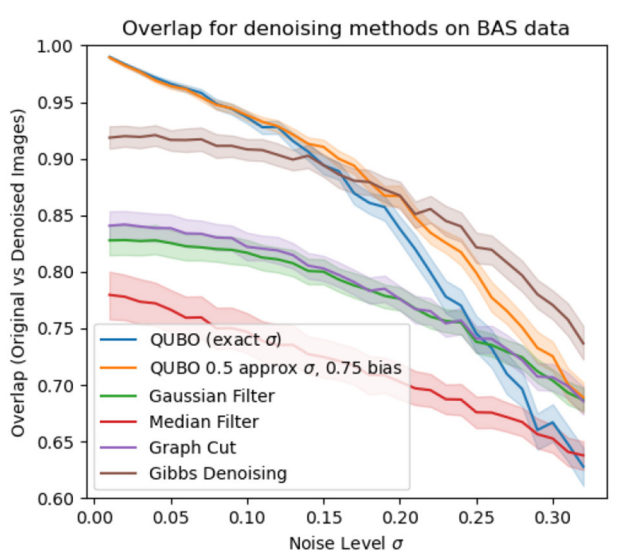


FIGURE 5
Proportion of pixels in denoised BAS images that matched the original image, for different denoising methods with 95% CI error bars.

for our QUBO-based method, and parameters of the method were carefully tuned for best performance to use 20 Gibbs iterations to then construct the denoised image as the exponentially weighted average of the samples with decay factor 0.8. Notably, as Gibbs-based denoising also requires a well-trained RBM, this method incurs the same computational overhead of training an RBM as our method does. However, it has the disadvantage of requiring careful tuning of the hyperparameters of the number of Gibbs iterations and decay factor to use, whereas our method of picking ρ is much more straightforward and shows good results without tuning. For the graph-cut method, the recommended parameter setting in the reference of $\beta = 0.5$ is used. The median filter, Gaussian filter, and Gibbs denoising (excluding the overhead of training the RBM) each have complexity $O(n)$, where n is the number of pixels, whereas the graph-cut method has complexity $O(n^3)$ since a maximum-flow problem is solved on a graph whose nodes are the pixels of the image. Keeping the annealing time and number of reads as constant, the scaling of our method is also $O(n)$. We forego wall-time here, since the software implementations we compare against are specialized for large problems, so comparing walltime for the small problems that can be implemented on current quantum annealers may not be representative. However, we note that for the QUBO denoising as we use up to $50\mu s$ annealing time and 100 reads per image, denoising an image only takes a total of $5ms$ of annealing time in our case.

Results are summarized in Figures 4, 5. Overall, the QUBO-based method performs quite strongly. Across all noise regimes in the MNIST data, and in most noise regimes in the bars-and-stripes dataset, the method outperforms the others. In particular, for the MNIST data the 95% confidence region for the QUBO method entirely dominates the others. Indeed, we see the good performance that our analysis from Section 3 suggests, even when the true σ is unknown and instead guessed. Using a guessed σ and the robustness modification of Section 3.3 makes the method perform as well (if not slightly better) as knowing the true σ without the robustness modification. Only in the noise regime of $\sigma \geq 0.2$ in the BAS data does Gibbs denoising outperform our method.

4.3. Testing on larger images

Though we see the the straightforward implementability of our method on quantum annealers as a strong positive, a current drawback on using QAs is the limited data size that can be handled to accomodate their still small qubit capacities. Of course we can still instead test our method on larger datasets by obtaining solutions to the denoising QUBO 6 using other means. In Figure 6, we implement our method on a binarized version of the popular MNIST dataset (LeCun et al., 2010) by using simulated annealing (Kirkpatrick et al., 1983) to find solutions to (6). We particularly choose to test on the full-size MNIST dataset since we could only use a downscaled version on the QA due to size limitations on the input data, so this experiment serves to test our method without this downscaling. All methods are implemented as described in Section 4.1, and again for our method we use a guessed σ to simulate the unknown σ case and bias the guess for robustness.

5. Conclusion and future work

We investigated an image denoising framework via a penalty-based QUBO denoising objective that shows promise both theoretically through its statistical properties and practically through its empirical performance together with

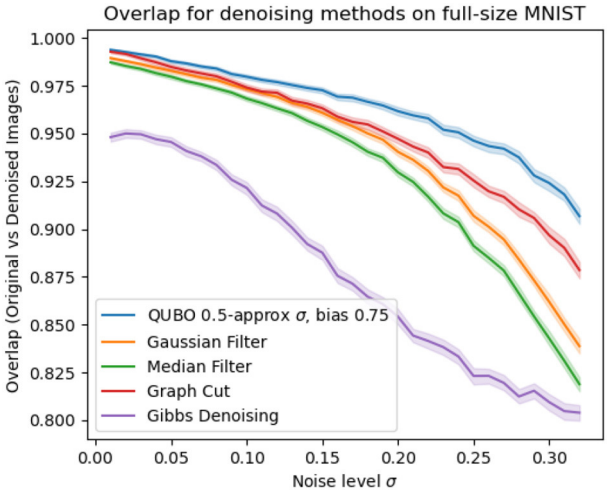


FIGURE 6
Proportion of pixels in denoised images that were correctly denoised, for different denoising methods on the MNIST dataset, with 95% confidence intervals shaded.

the proposed robustness modification. The method is well-suited for implementability on a quantum annealer, providing an important application of QAs within machine learning through the fundamental image denoising task. Good results are still obtained on larger datasets when the QUBO is only classically approximated by simulated annealing instead, revealing the approach to be promising even in the absence of QAs. As RBMs form a core building block of many deep generative models such as deep Boltzmann machines or deep belief networks (Goodfellow et al., 2016), a natural next step is to attempt to incorporate this approach into these more complex models, though current hardware limitations on existing quantum annealers are restrictive. Further, since our method takes advantage of QAs for the denoising step, further research into making use of QAs for the training process of RBMs would yield a full image denoising model where both the model training and image denoising make use of QA.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://github.com/PhillipKerger/Code_for_figures_of_QUBO_RBM_denoising_paper.

Author contributions

PK: Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Software, Validation, Visualization, Writing—original draft, Writing—review and editing. RM: Conceptualization, Formal analysis, Funding acquisition, Project administration, Resources, Software, Supervision, Writing—review and editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. PK was supported in part by g-RIPS Sendai, Cyberscience Center at Tohoku Univ., and NEC Japan, in early stages of the work. PK is grateful to the USRA Feynman Academy internship program, support from the NASA Academic Mission Services (contract NNA16BD14C), and funding from DARPA under DARPA-NASA agreement SAA2-403688.

Acknowledgments

The early stage of this work is based on the work in the g-RIPS Sendai 2021 program. The authors thank Y. Araki, E. Escobar, T. Mihara, V. Q. H. Huynh, H. Kodani, A. T. Lin, M. Shirane, Y. Susa, and H. Suito for collaboration in the program. The authors also acknowledge H. Kobayashi and M. Sato for the use of the computing environment in the program. PK thanks Y. Sukurdeep for helpful feedback and discussions.

Conflict of interest

RM was employed by NEC Corporation.

The remaining author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Adachi, S. H., and Henderson, M. P. (2015). Application of quantum annealing to training of deep neural networks. *ArXiv*. ArXiv:1510.06356. doi: 10.48550/arXiv.1510.06356

Albash, T., and Lidar, D. A. (2018). Adiabatic quantum computing. *Rev. Mod. Phys.* 90, 15002. doi: 10.1103/RevModPhys.90.015002

Barahona, F. (1982). On the computational complexity of Ising spin glass models. *J. Phys. A Math. Gen.* 15, 3241–3253. doi: 10.1088/0305-4470/15/10/028

Benedetti, M., Realpe-G'omez, J., Biswas, R., and Perdomo-Ortiz, A. (2015). Estimation of effective temperatures in quantum annealers for sampling applications:

a case study with possible applications in deep learning. *Phys. Rev. A* 94, 022308. doi: 10.1103/PhysRevA.94.022308

Boyat, A. K., and Joshi, B. K. (2015). A review paper: noise models in digital image processing. *ArXiv*. doi: 10.5121/sipij.2015.6206

Buades, A., Coll, B., and Morel, J.-M. (2005). A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.* 4, 490–530. doi: 10.1137/040616024

Cho, K. (2013). Boltzmann machines and denoising autoencoders for image denoising. *ArXiv*. ArXiv:1301.3468. doi: 10.48550/arXiv.1301.3468

Das, A., and Chakrabarti, B. K. (2008). Colloquium: quantum annealing and analog quantum computation. *Rev. Mod. Phys.* 80, 1061–1081. doi: 10.1103/RevModPhys.80.1061

Dixit, V., Selvarajan, R., Alam, M. A., Humble, T. S., and Kais, S. (2021). Training restricted boltzmann machines with a d-wave quantum annealer. *Front. Phys.* 9, 589626. doi: 10.3389/fphy.2021.589626

Glover, F. W., Kochenberger, G. A., and Du, Y. (2018). Quantum bridge analytics i: a tutorial on formulating and using qubo models. *4OR* 17, 335–371. doi: 10.1007/s10288-019-00424-y

Goodfellow, I. J., Bengio, Y., and Courville, A. (2016). *Deep Learning*. Cambridge, MA: MIT Press. Available online at: http://www.deeplearningbook.org (accessed August 20, 2023).

Greig, D., Porteous, B., and Seheult, A. H. (1989). Exact maximum a posteriori estimation for binary images. *J. R. Stat. Soc. Ser. B Methodol.* 51, 271–279. doi: 10.1111/j.2517-6161.1989.tb01764.x

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Comput.* 14, 1771–1800. doi: 10.1162/089976602760128018

Huang, T. S., Yang, G., and Tang, G. (1979). A fast two-dimensional median filtering algorithm. *IEEE Transact. Acoust. Speech Signal Process.* 27, 13–18. doi: 10.1109/TASSP.1979.1163188

Johnson, M. W., Amin, M. H. S., Gildert, S., Lanting, T., Hamze, F., Dickson, N., et al. (2011). Quantum annealing with manufactured spins. *Nature* 473, 194–198. doi: 10.1038/nature10012

Kadowaki, T., and Nishimori, H. (1998). Quantum annealing in the transverse ising model. *Phys. Rev. E* 58, 5355–5363. doi: 10.1103/PhysRevE.58.5355

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science* 220, 671–680. doi: 10.1126/science.220.4598.671

Koshka, Y., and Novotny, M. A. (2021). Comparison of use of a 2000 qubit d-wave quantum annealer and mcmc for sampling, image reconstruction, and classification. *IEEE Transact. Emerg. Top. Comp. Intell.* 5, 119–129. doi: 10.1109/TETCI.2018.2871466

Krzysztof, K., Mateusz, S., Marek, S., and Rafał, R. (2021). Applying a quantum annealing based restricted boltzmann machine for mnist handwritten digit classification. *Comp. Methods Sci. Technol.* 27, 99–107. doi: 10.12921/cmst.2021.0000011

LeCun, Y., Cortes, C., and Burges, C. (2010). *Mnist Handwritten Digit Database*. ATT Labs. Available online at: http://yann.lecun.com/exdb/mnist (accessed August 20, 2023).

Lucas, A. (2014). Ising formulations of many NP problems. *Front. Phys.* 2, 5. doi: 10.3389/fphy.2014.00005

Nishimori, H. (2001). *Statistical Physics of Spin Glasses and Information Processing: An Introduction*. New York, NY: Oxford University Press.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Phys. D* 60, 259–268. doi: 10.1016/0167-2789(92)90242-F

Stockman, G., and Shapiro, L. G. (2001). *Computer Vision*. Upper Saddle River, NJ: Prentice Hall PTR.

Tang, Y., Salakhutdinov, R., and Hinton, G. E. (2012). "Robust boltzmann machines for recognition and denoising," in *2012 IEEE Conference on Computer Vision and Pattern Recognition* (Providence, RI), 2264–2271.

Vahdat, A. (2017). "Toward robustness against label noise in training deep discriminative neural networks," in *NeurIPS Proceedings 2017* (Long Beach, CA).

frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 15 November 2023
DOI 10.3389/fcomp.2023.1294564

OPEN ACCESS

EDITED BY
Susan Mniszewski,
Los Alamos National Laboratory (DOE),
United States

REVIEWED BY
Ma Hongyang,
Qingdao University of Technology, China
Richard Warren,
The Pennsylvania State University (PSU),
United States

*CORRESPONDENCE
Juan Francisco Ariño Sales
✉ juanarinosales@gmail.com
Raúl Andrés Palacios Araos
✉ raul.andres.palacios@gmail.com

†These authors have contributed equally to this work

RECEIVED 14 September 2023
ACCEPTED 30 October 2023
PUBLISHED 15 November 2023

CITATION
Ariño Sales JF and Palacios Araos RA (2023) Adiabatic quantum computing impact on transport optimization in the last-mile scenario. *Front. Comput. Sci.* 5:1294564. doi: 10.3389/fcomp.2023.1294564

Adiabatic quantum computing impact on transport optimization in the last-mile scenario

Juan Francisco Ariño Sales*† and Raúl Andrés Palacios Araos*†

Higher Technical School of Computer Systems Engineering (ETSISI), Polytechnic University of Madrid (UPM), Madrid, Spain

In the ever-evolving landscape of global trade and supply chain management, logistics optimization stands as a critical challenge. This study takes on the Vehicle Routing Problem (VRP), a variant of the Traveling Salesman Problem (TSP), by proposing a novel hybrid solution that seamlessly combines classical and quantum computing methodologies. Through a comprehensive analysis of our approach, including algorithm selection, data collection, and computational processes, we provide in-depth insights into the efficiency, and effectiveness of our hybrid solution compared to traditional methods. The results after analysis of 14 datasets highlight the advantages and limitations of this approach, demonstrating its potential to address NP-hard problems and contribute significantly to the field of optimization algorithms in logistics. This research offers promising contributions to the advancement of logistics optimization techniques and their potential implications for enhancing supply chain efficiency.

KEYWORDS

quantum computing, quantum annealing, quadratic unconstrained binary optimization (QUBO), vehicle routing problem (VRP), traveling salesman problem (TSP), supply chain, last mile

1 Introduction

The quantum computing industry is in a bustling emerging phase, and many around the world are determining its applicability in real-life business scenarios. Enthusiasts, startups, academia, and governments are rushing to find "quantum advantage" and funding mid- to long-term research and development in this area. Many companies are working hard to develop and mature current quantum hardware, plus there is an increasing growth in areas related to software and services aiming to reap the benefits of quantum computing.

We have been investigating how to bridge the gap between scientific developments and current industry trends and needs. The process we followed is shown in Figure 1.

Current quantum computing technology is focused on problems such as simulation, optimization, factorization, linear algebra, and machine learning. Through these, it promises to deliver value in many different areas: life sciences, transport and logistics, financial services, and telecommunications, just to name a few.

2 Market analysis

The case for quantum computing in transport optimization is quite compelling. Current world trade is based on a strong and healthy supply chain, where logistics plays a key role in producing and providing key assets and goods to keep societies and economies going. One facet of the transport optimization problem is the vehicle routing problem (VRP). This

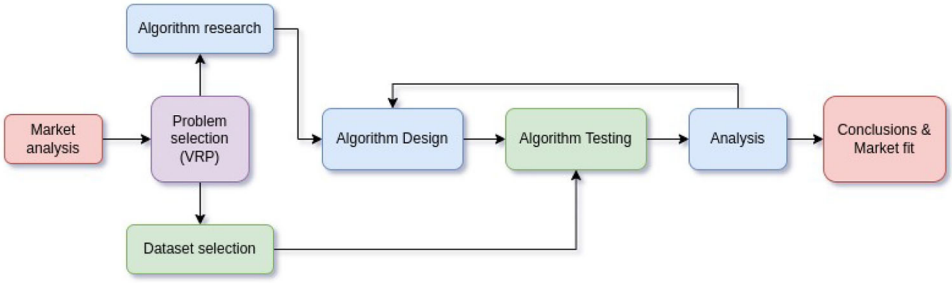


FIGURE 1
Research flowchart.

problem attempts to find an optimal set of routes for a fleet of vehicles to service a given set of customers; the business impact of the VRP is well measured.

The goal of last-mile delivery is to transport an item to its recipient in the quickest way possible. This has been driven by the continuously evolving market and demand for a convenient customer experience across industries such as e-commerce, food, retail, and many more. The last-mile delivery market has been steadily growing in the last decade, and the forecast opportunity follows the same path. The last-mile delivery market in Europe is expected to grow from USD 677.0 Mn in 2018 to USD 2,491.8 Mn by the year 2027, with a compound annual growth rate of 16.1% from 2019 to 2027,[1] while in Latin America, the level of investment for the last five years is close to USD 300 Mn, leaving countries like Mexico, Colombia, Chile, and Argentina without a leading independent last-mile logistics company, where 60% of the last-mile delivery market is dominated by small, informal companies. This results in inefficiencies due to a lack of technologies such as route optimization as well as a lack of operating scale. These issues are quickly becoming more pronounced as e-commerce in Latin America has taken off at a compound annual industry growth rate of 16% over the past five years. In the case of Latam, the biggest e-commerce companies and retailers have made last-mile logistics the key value differentiator for growth, leveraging the technology tools and analytics processes to make investments and plans in advance.[2] The situation for Europe is quite similar, given the importance of optimization in last-mile transportation. Key factors driving the region's market growth include rapid industrialization, the growth of the e-commerce sector, and the presence of large and established logistics players. While Germany is a predominant player in the European market, the main segment responsible for its growth is the business-to-consumer (B2C) sector. In Spain, the last-mile market is mainly indexed to the B2C sector, which is accountable for over USD 40 Bn e-commerce market size, where last-mile represents around 40% of total costs of logistics operations in a market dominated up to 80% by small or micro-enterprises (Deloitte, 2020).

The fact that there are common components in the last-mile market makes the proposal in this paper appealing for a close-term application of the technology and solution. In a rough estimate, for a market of USD 27 billion in Spain, with an average of 10% margin, where the Last Mile may represent something between 30 and 40% of the total cost, we aim for a USD 15 billion market, split in a granular small to micro enterprise sector, with close to 2,000 companies (de los Mercados y la Competencia, 2021). Any 1% savings in optimization can prove to be worth a very competitive return on investment; this is shown in Table 1.

Our work is focused on the applicability of transport optimization for the last-mile scenario. Transport optimization is the process of finding the best way to move assets from one place (the source location) to another (the destination). It is impacted by many distinct factors, like shipment analysis, transport cost structures, rates, and schedules, cargo, routes, delivery requirements and needs, etc. Combining all these different factors makes this problem extraordinarily complex and demands high computing power to find viable solutions. The problem is categorized as an NP-hard problem. Transport optimization, may be rephrased as finding the optimal value for a transport function; this is where it becomes a high-prospect match for current quantum technologies, specifically quantum annealing (Farhi et al., 2000).

3 Implementation

In order to find the best approach in terms of technology and time-to-market applicability, we solve the VRP using a hybrid approach (Feld et al., 2019), which exploits both classical and quantum techniques to find an optimized solution. The hybrid algorithm models the VRP problem using a 2-phase approach: first clustering or grouping the customers, and then finding the optimal routes inside each cluster. This approach is known as a cluster-first, route-second algorithm. For each of the two phases, we developed both a quantum and a classical algorithm to compare them and determine the most effective combination. The algorithms used are shown in Table 2.

1 https://techcrunch.com/2021/07/22/last-mile-delivery-in-latin-america-is-ready-to-take-off/

2 https://www.mundomaritimo.cl/noticias/mercado-libre-amplia-brecha-con-falabella-y-se-prepara-para-enfrentar-la-irrupcion-de-amazon

TABLE 1 Preliminary return on investment estimations.

Annual market size	$ 27,000,000,000
Estimated costs	$ 810,000,000
Gross yearly inv. estimate	$ 5,000,000
Return on investment	≈ 62%

TABLE 2 Algorithms developed for solving the VRP problem with a cluster-first, route-second approach.

	Clustering	Routing
Classical	K-Medoids	Combinatorial optimization
Quantum	QUBO clustering	QUBO routing

3.1 Clustering phase

During the clustering phase, the objective is to find clusters of customers such that the intra-cluster distances are minimized. The clustering problem has additional constraints imposed so the sum of the demand of each customer inside each cluster does not exceed the available transport capacity of the vehicles; thus, the problem is a constrained clustering problem with cluster-level constraints. To solve it, we developed a modified version of the K-Medoids algorithm that takes into account the capacity constraints as the classical approach and a quadratic unconstrained binary optimization (QUBO) formulation of the problem as the quantum approach.

The QUBO formulation (Bauckhage et al., 2019; Date et al., 2021; Matsumoto et al., 2022) for the clustering phase shown below (Equation 4) is composed of the main objective function M (Equation 1) subject to two additional constraints. The main formula M tries to find an assignment of customers in clusters such that the total distance between customers inside each cluster is minimized. The first constraint C_1 (Equation 2) adds a penalty for each customer not included in a cluster; the second constraint C_2 (Equation 3) adds a penalty for each cluster in which the total customer demand is greater than the available vehicle capacity.

$$M = \sum_{k \in K} \sum_{i,j \in I j > i} dist_{i,j} * x_{i,k} * x_{j,k} \tag{1}$$

$$C_1 = \sum_{k \in K} x_{i,k} = 1 \qquad \forall i \in I \tag{2}$$

$$C_2 = \sum_{i \in I} d_i * x_{i,k} \leq C \qquad \forall k \in K \tag{3}$$

$$H = M + C_1 * M_1 + C2 * M_2 \tag{4}$$

K is the total number of clusters, while I indicates the customer nodes. $dist_{ij}$ represents the distance matrix between all the possible customer nodes; this matrix is pre-computed beforehand. x_{ik} is a binary decision variable that indicates if the customer i is assigned to cluster k. C represents the available vehicle capacity, and d_i is the demand of customer i. The multipliers M_1 and M_2 are used to assign the weight of the corresponding penalty for each of the two constraints.

The developed K-Medoids algorithm is based on the Partitioning Around Medoids algorithm with an added capacity constraint. The steps of the algorithm are the following:

1. Select K data points with the highest demand as the medoids.
2. Determine the clusters by associating each data point to its closest medoid.
3. Compute the initial cluster costs by adding the distances from every point in each cluster to their medoid, add a penalty cost if the total demand of the cluster exceeds the vehicle capacity.
4. While the cluster costs decrease and the maximum number of iterations has not been reached:
   (a) For each medoid m and for each non-medoid data point: n
      i. Swap m and o and recompute the cluster costs.
      ii. If the new cluster cost is higher than the previous one, undo the swap.
   (b) Increase number of iterations.
5. Return the clusters.

3.2 Routing phase

Once the clusters have been established, the routing phase attempts to find the shortest routes starting from the depot, which travel through all the nodes and finally return to the depot. This problem is very similar to the Traveling Salesman Problem.

To solve the routing phase, we developed a combinatorial optimization algorithm as the classical approach and a QUBO formulation of the problem as the quantum approach. The QUBO formulation (Lucas, 2014), shown below (Equation 10), is composed of two different QUBO equations. The first equation (8) attempts to solve the Hamiltonian cycle problem, while the second equation (9) minimizes the route distances, thus solving the Traveling Salesman problem.

$$C_1 = \sum_{j \in N+1} (1 - \sum_{i \in N+1} x_{i,j}) \tag{5}$$

$$C_2 = \sum_{i \in N+1} (1 - \sum_{j \in N+1} x_{i,j}) \tag{6}$$

$$C_3 = (1 - x_{0,0}) \tag{7}$$

$$H_A = C_1 + C_2 + C_3 \tag{8}$$

$$H_B = \sum_{h \in N+1} \sum_{i \in N+1, h \neq i} \sum_{j \in N} d_{h,i} x_{j,h} x_{j+1,i} \tag{9}$$

$$H = H_A * m_A + H_B * m_B \tag{10}$$

$x_{i,j}$ is a binary variable where i represents the order and j represents the customer. $x_{i,j}$ is equal to 1 if the customer with index j is visited in position i in the cycle, $i, j \in 0, \ldots, N$ where N is

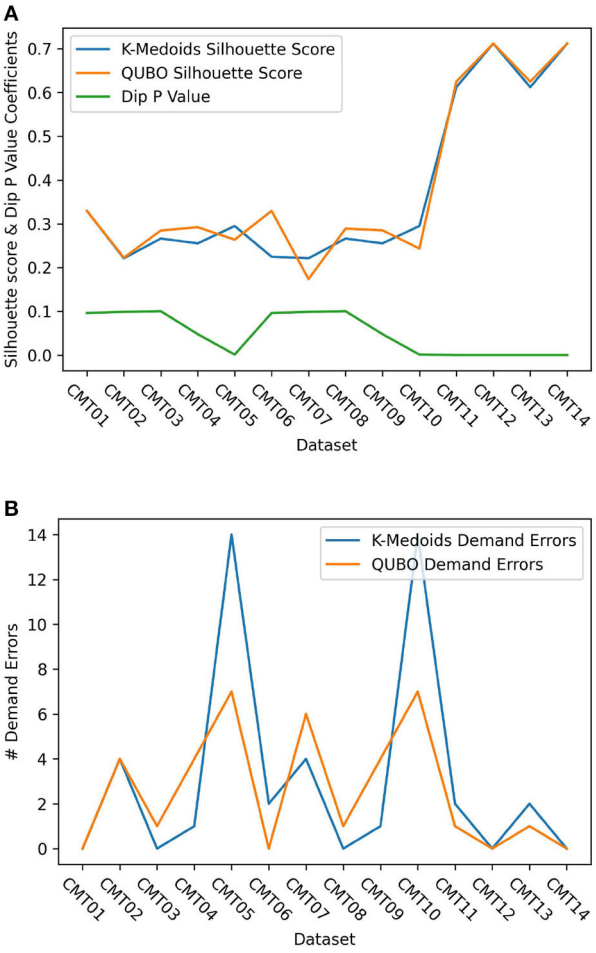


FIGURE 2
Silhouette score and number of demand errors obtained with K-Medoids and QUBO clustering algorithms on the *CMT 01-14* data-sets. **(A)** Silhouette score and dip *P* value. **(B)** Number of demand errors.

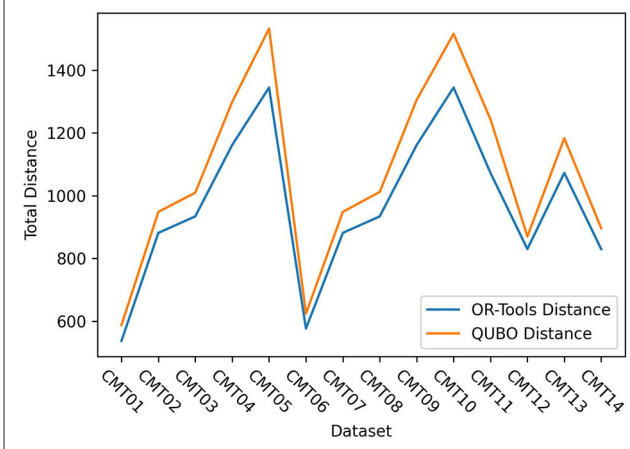


FIGURE 3
Total distance obtained with OR-Tools and QUBO routing algorithms on the *CMT 01-14* data-sets.

equal to the total number of customers. d is the distance matrix, which contains the distance between every customer; the depot is included as customer 0. The multipliers m_A and m_B are used to set the penalties for the distinct parts of the equation.

The first constraint C_1 (Equation 5) ensures that every customer can only appear once in the cycle. The second constraint C_2 (Equation 6) ensures that each position in the cycle must be assigned to only one customer. The third constraint C_3 (Equation 7) is added so that every cycle starts at the depot.

The combinatorial optimization algorithm models the Traveling Salesman Problem using Google's OR-Tools framework.[3] The search strategy used to find the solution is a meta-heuristic strategy called Guided Local Search (GLS). It is built on top of a local search algorithm while gradually adding penalties to certain features of the solutions to help the local search escape from local minima and plateaus.

3 https://acrogenesis.com/or-tools/documentation/user_manual/manual/tsp/routing_library.html

4 Analysis

The experiments conducted in this study involved the utilization of a diverse range of datasets (Mendoza et al., 2014), featuring varying numbers of customers and vehicles. It is noteworthy that these datasets exhibit a wide spectrum of clusterability (Ackerman et al., 2016) rates regarding customer positions.

To evaluate the performance of our clustering algorithms, we employed two different metrics: the silhouette score (Rousseeuw, 1987) and the error count. The silhouette score measures the quality of the generated clusters by comparing the similarity of objects with their own cluster and with the other clusters. Its value ranges from −1 to +1; a higher value indicates the elements are well clustered. The number of errors generated by each clustering algorithm is the number of clusters where the total demand of its members exceeds the available vehicle capacity. The results, presented in Figure 2, focus on the application of these metrics to the datasets presented by Christofides, Mingozzi, and Toth (CMT) (Christofides et al., 1979).

In Figures 2A, B, we observe that both algorithms perform comparably when the data exhibits a high level of clusterability (as indicated by a low dip P value Hartigan and Hartigan, 1985). However, in scenarios where the data exhibits a lower rate of clusterability, the QUBO formulation generally excels in producing more robust clusters when contrasted with the traditional K-Medoids algorithm. This success can be attributed to the enhanced flexibility inherent in the QUBO formulation compared to the classical K-Medoids algorithm. Furthermore, the quantum approach typically demonstrates a lower error rate, underscoring its adaptability and efficiency.

To assess the quality of the routing algorithms, we focused on measuring the total route distance generated by each algorithm. Figure 3 presents a comparison of the results obtained by both algorithms when applied to the CMT datasets used in the clustering phase.

Figure 3 highlights a notable trend, specifically that the QUBO formulation for the routing problem typically yields longer route distances when compared to those produced by the combinatorial optimization algorithm. This observation underscores the need for further refinement of the quantum approach to match the optimization efficiency demonstrated by the classical algorithm.

The experiments with the quantum algorithms were performed using D-Wave's Advantage System 6.1 quantum annealer, offered by Amazon Braket. The size of the QUBO formula generated by the clustering algorithm is too large to embed on the available quantum annealers, so QBSolv (Booth et al., 2017) is used to split it into smaller sub-problems.

All the code necessary to run the experiments is available at https://github.com/punkyfer/vrpc.

5 Conclusions

During our research, we found that classical algorithms typically perform better than their quantum counterparts. This is not a totally fair comparison since, on the one hand, we have fine-tuned algorithms running on classical computing hardware, and on the other hand, we have QUBO formulations running on quantum annealers in noisy intermediate-scale quantum era hardware. Both technologies are on wildly different edges of the technology maturity ladder. Despite this disadvantageous situation, we have found that under certain circumstances, the quantum clustering algorithm presents an advantage over its classical counterpart, mainly in scenarios where the clusterability rate of the data is lower. When the data presents a lower rate of clusterability or a higher degree of randomness, the quantum clustering approach delivers better results than the K-Medoids algorithm. This is an outstanding finding, as it proves the potential for quantum computing in real business scenarios and sets the basis for future research into developing quantum algorithms for the constrained clustering problem. A bigger advantage may be achievable in future versions of quantum hardware, where more qubits and a more interconnected topology may provide better results at larger scales.

For the business analysis, we identified the potential for a cost-effective relationship between the cost of running a quantum algorithm and the quality of the results obtained. This cost-effectiveness is especially true when the data shows a higher degree of randomness, as is usually the case with real customer location data. This demonstrates a theoretical advantage for the quantum computing approach when applied to the constrained clustering problem.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

Author contributions

JA: Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Software, Supervision, Validation, Visualization, Writing—original draft, Writing—review & editing. RP: Conceptualization, Formal analysis, Funding acquisition, Investigation, Project administration, Resources, Supervision, Validation, Writing—original draft, Writing—review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. Grupo GTD financed us with the funds necessary to run the experiments on real quantum hardware.

Acknowledgments

Our thanks to Alberto García and the rest of the teachers at UPM for providing us with the knowledge and guidance required to accomplish this work.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Ackerman, M., Adolfsson, A., and Brownstein, N. (2016). An effective and efficient approach for clusterability evaluation. *arXiv preprint arXiv:1602.06687*.

Bauckhage, C., Piatkowski, N., Sifa, R., Hecker, D., and Wrobel, S. (2019). "A qubo formulation of the k-medoids problem," in *Lernen, Wissen, Daten, Analysen*.

Booth, M., Reinhardt, S. P., and Roy, A. (2017). *Partitioning optimization problems for hybrid classical/quantum execution technical report*. Technical Report.

Christofides, N., Mingozzi, A., and Toth, P. (1979). *Combinatorial Optimization*. Chichester, UK: Wiley.

Comisión Nacional de los Mercados y la Competencia. (2021). Informe anual del sector postal 2020. Available online at: https://www.cnmc.es/sites/default/files/4276304.pdf (accessed October 27, 2023).

Date, P., Arthur, D., and Pusey-Nazzaro, L. (2021). Qubo formulations for training machine learning models. *Scient. Rep*. 11, 10029. doi: 10.1038/s41598-021-89461-4

Deloitte (2020). *Last mile logistics, challenges and solutions in Spain*. Technical Report.

Farhi, E., Goldstone, J., Gutmann, S., and Sipser, M. (2000). Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*.

Feld, S., Roch, C., Gabor, T., Seidel, C., Neukart, F., Galter, I., et al. (2019). A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer. *Front. ICT* 6, 13. doi: 10.3389/fict.2019.00013

Hartigan, J. A., and Hartigan, P. M. (1985). The dip test of unimodality. *Ann. Statist*. 13, 70–84. doi: 10.1214/aos/1176346577

Lucas, A. (2014). Ising formulations of many NP problems. *Front. Phys*. 2, 5. doi: 10.3389/fphy.2014.00005

Matsumoto, N., Hamakawa, Y., Tatsumura, K., and Kudo, K. (2022). Distance-based clustering using Qubo formulations. *Scient. Rep*. 12, 2669. doi: 10.1038/s41598-022-06559-z

Mendoza, J., Guéret, C., Hoskins, M., Lobit, H., Pillac, V., Vidal, T., et al. (2014). "VRP-REP: the vehicle routing community repository," in *Third meeting of the EURO Working Group on Vehicle Routing and Logistics Optimization (VeRoLog)* (Oslo, Norway).

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Mathem*. 20, 53–65. doi: 10.1016/0377-0427(87)90125-7

frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 21 November 2023
DOI 10.3389/fcomp.2023.1275948

OPEN ACCESS

EDITED BY
Nicholas Chancellor,
Durham University, United Kingdom

REVIEWED BY
Itay Hen,
University of Southern California, United States
Jie-Hong Jiang,
National Taiwan University, Taiwan

*CORRESPONDENCE
Georg Hahn
ghahn@hsph.harvard.edu

RECEIVED 10 August 2023
ACCEPTED 16 October 2023
PUBLISHED 21 November 2023

CITATION
Hahn G, Pelofske E and Djidjev HN (2023) Posiform planting: generating QUBO instances for benchmarking.
Front. Comput. Sci. 5:1275948.
doi: 10.3389/fcomp.2023.1275948

Posiform planting: generating QUBO instances for benchmarking

Georg Hahn[1]*, Elijah Pelofske[2] and Hristo N. Djidjev[2,3]

[1]T. H. Chan School of Public Health, Harvard University, Boston, MA, United States, [2]Los Alamos National Laboratory, CCS-3 Information Sciences, Los Alamos, NM, United States, [3]Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Sofia, Bulgaria

We are interested in benchmarking both quantum annealing and classical algorithms for minimizing quadratic unconstrained binary optimization (QUBO) problems. Such problems are NP-hard in general, implying that the exact minima of randomly generated instances are hard to find and thus typically unknown. While brute forcing smaller instances is possible, such instances are typically not interesting due to being too easy for both quantum and classical algorithms. In this contribution, we propose a novel method, called *posiform planting*, for generating random QUBO instances of arbitrary size with known optimal solutions, and use those instances to benchmark the sampling quality of four D-Wave quantum annealers utilizing different interconnection structures (Chimera, Pegasus, and Zephyr hardware graphs) and the simulated annealing algorithm. Posiform planting differs from many existing methods in two key ways. It ensures the uniqueness of the planted optimal solution, thus avoiding groundstate degeneracy, and it enables the generation of QUBOs that are tailored to a given hardware connectivity structure, provided that the connectivity is not too sparse. Posiform planted QUBOs are a type of 2-SAT boolean satisfiability combinatorial optimization problems. Our experiments demonstrate the capability of the D-Wave quantum annealers to sample the optimal planted solution of combinatorial optimization problems with up to 5,627 qubits.

KEYWORDS

quantum annealing (QA), QUBO problem, MAX-2-SAT, planted solution, time-to-solution, combinatorial optimization problem, boolean satisfiability (SAT), quadratic unconstrained binary optimization

1 Introduction

Many important NP-hard optimization problems can be easily expressed in a QUBO (quadratic unconstrained binary optimization) or an Ising form (Lucas, 2014), given by the quadratic function

$$Q(x_1, \ldots, x_n) = \sum_{i=1}^{n} a_i x_i + \sum_{i<j} a_{ij} x_i x_j \tag{1}$$

in $n \in \mathbb{N}$ binary variables. In Equation (1), the linear weights $a_i \in \mathbb{R}$ and the quadratic couplers $a_{ij} \in \mathbb{R}$ define the problem under investigation and are chosen by the user. The assignments of the variables x_i for $i \in \{1, \ldots, n\}$ are unknown, and we seek a configuration of $(x_1, \ldots, x_n)$ minimizing Equation (1). If $x_i \in \{0, 1\}$, then Equation (1) is called a QUBO problem, and if $x_i \in \{-1, +1\}$, then it is called an Ising model.

Since many NP hard problems can be formulated as QUBO models, it is of interest to efficiently compute the optimal solution(s) of general QUBO problems. To this end, researchers have developed a variety of classical approaches (Kirkpatrick et al., 1983; Boros et al., 2006, 2007) to compute solutions of high quality that minimize Equation (1). Quantum annealing offers an experimental route to sample combinatorial problems. Quantum annealing is a type of analog quantum computation that uses quantum fluctuations to attempt to arrive at an optimal (or a very good) minimum of Equation (1) (Kadowaki and Nishimori, 1998; Das and Chakrabarti, 2008; Morita and Nishimori, 2008; Hauke et al., 2020). The quantum annealing algorithm has been physically instantiated in a number of ways, including superconducting flux qubit hardware that is manufactured by D-Wave Systems, Inc. The D-Wave quantum annealers have been evaluated for sampling a large number of different types of problems, typically focusing on combinatorial optimization problems or Hamiltonian dynamics (Boixo et al., 2013, 2014, 2016; Lanting et al., 2014; Venturelli et al., 2015; Harris et al., 2018; King et al., 2021, 2022, 2023; Tasseff et al., 2022). D-Wave quantum annealing devices offer on the scale of hundreds to thousands of qubits, but are still subject to connectivity constraints, control errors, and noise from the environment (Pearson et al., 2019; Lanting et al., 2020; Nelson et al., 2021; Zaborniak and de Sousa, 2021; Grant and Humble, 2022; Pelofske et al., 2023). To map a QUBO Q of Equation (1) directly on the hardware chip of a quantum annealer, its connectivity structure should be consistent with the connectivity structure of the quantum device. Specifically, each variable x_i is mapped to a distinct qubit q_i. For each non-zero coefficient a_{ij}, there should be a coupler (direct link) between qubits q_i and q_j. If a direct embedding is not possible, then a *minor embedding* of the graph representing the sparsity structure of the QUBO Q onto the graph defined by the hardware structure can be used (Choi, 2008, 2011; Könz et al., 2021; Marshall et al., 2022). However, the number of qubits required in that case may grow quadratically with the size of Q.

To better assess the capabilities of both classical and quantum approaches for sampling (approximate) solutions of combinatorial optimization problems, methods are needed that generate benchmark problems with (ideally) known solutions. Two strategies exist to achieve this goal. First, one can generate problems of the type of Equation (1) with randomly sampled linear and quadratic weights, and then brute force them. However, brute forcing is only feasible for problems with a relatively small number of variables (roughly 30 variables for full brute force computations). Second, methods have been developed that allow one to generate QUBO problems with planted solutions, that is, problems generated to have a solution that is specified a priori. A detailed overview of such methods is given in Section 1.1. Importantly, existing methods often have two shortcomings. Many approaches only ensure that the generated problem has a minimum at the planted solution, but do not guarantee its uniqueness. Moreover, for many methods, the sparsity structure of the generated QUBO cannot be chosen, which means the QUBO cannot directly be solved on certain hardware devices. Naturally, since the minimization of Equation (1) is NP-hard, all methods exploit some form of shortcut or mathematical device to generate large problems with non-trivial structures and known solutions.

In this contribution, we introduce a new method to generate QUBO problems of the type of Equation (1) with a single planted solution. The method is called *posiform planting*, in reference to the mechanism we exploit that generates a QUBO in posiform representation. The posiforms are converted to QUBOs only at a later stage when the solution has been planted. Two features of our algorithm are noteworthy. First, it guarantees the uniqueness of the planted solution. Moreover, the connectivity structure of the QUBO can, in principle, be chosen freely. Naturally, the generated QUBOs need to have at least a certain number of quadratic terms to guarantee the uniqueness of the planted solution and thus cannot be too sparse, although this also depends on the solution being planted. In contrast to some existing solution-planting methods, such as the *tile planting* or *deceptive cluster loops* methods of the *Chook* toolbox (Perera et al., 2021), posiform planting generates QUBO problems which include linear terms.

The adaptation to an arbitrary connectivity structure is of importance when generating problems that are tailored to, for instance, the qubit connectivity structure of the D-Wave quantum annealers. In particular, the physical qubits across currently existing D-Wave generations use connections determined by Chimera, Pegasus, or Zephyr graphs (Dattani et al., 2019; Boothby et al., 2020). Being able to tailor the generated problems to any arbitrary architecture allows one to generate much larger benchmark problems compared to the case where the problems cannot be directly embedded, thus necessitating the computation of a minor embedding onto the D-Wave QPU chip structure.

One of the properties of the transverse field driver in quantum annealing and other approximate quantum optimization algorithms is that degenerate ground states are not in general sampled uniformly (Matsuda et al., 2009; Mandrà et al., 2017; Zhang et al., 2017; Könz, 2019; Könz et al., 2019; Kumar et al., 2020; Pelofske et al., 2021; Nelson et al., 2022). Posiform planting guarantees the uniqueness of the planted optimal solution. Thus, any use cases in which biased sampling of degenerate solutions needs to be avoided could benefit from posiform planting. Some use cases in which biased sampling of degenerate solutions should be avoided include the estimation the ground-state entropy of a degenerate physical systems, estimating the count of the total number of solutions in combinatorics or the estimation of ground state probabilities in industrial applications where the problem has several solutions by design (Mandrà et al., 2017).

This article is structured as follows. After a literature review in Section 1.1, we introduce the idea of posiform planting in Section 2. We evaluate the QUBO problems generated by posiform planting on D-Wave devices using both native connectivity (using the Chimera, Pegasus, and Zephyr hardware graphs), and arbitrarily connected minor embedded problem instances (Section 3). The hardware native QUBOs are also sampled using the classical heuristics simulated annealing and steepest gradient descent. The article concludes with a discussion in Section 4. Data and extra figures generated from this research are publicly available as a Zenodo dataset (Hahn et al., 2023).

1.1 Literature review

A variety of contributions in the literature focus on the generation of QUBO or Ising models of the type of Equation (1) that can serve as benchmark problems. These methods can be grouped according to the underlying mechanism they use to generate problems and the properties they guarantee. Originally, this property of known planted solutions was introduced from satisfiability problems (Barthel et al., 2002; Krzakala and Zdeborová, 2009).

One popular way to generate problems is with the help of frustrated loops, meaning Ising models of the form $Q = \sum_{j=1}^{M} Q_j$, where each Q_j only contains a subset of the variables. For instance, Hen et al. (2015) and King A. D. et al. (2015) generated frustrated Ising models with tunable hardness, though the authors explicitly pointed out that they cannot guarantee uniqueness. Similar methods are the so-called tile-planting (Perera et al., 2020) and patch-planting for Ising models (Wang et al., 2017). In Pei et al. (2020), the authors generate weighted MAX-2-SAT instances with the help of frustrated loops that have known solutions. Notably, the hardness of their problems can be tuned through a parameter called the frustration index.

One major drawback of many published planted solution methods is the fact that they do not guarantee the uniqueness of the planted solution, meaning that the input configuration is only guaranteed to be one of a possibly unknown number of minima. A notable exception is Kowalsky et al. (2022), who ensure the uniqueness of solution with an approach based on equation planting. However, the resulting QUBOs have a very special form as each linear equation is required to contain exactly three binary variables.

Another route is called equation planting, that is, the generation of QUBO problems from a set of (linear) equations. In Hen (2019), the author considers a set of linear equations modulo 2 to pin down the bitstring to be planted, and then recasts it as an Ising model. Their method is based on the experimental observation that although linear equations are easy to solve, they disguise the solution well for machines when being recast as an optimization problem. According to the author, equation planting guarantees the uniqueness of the planted solution. However, tailoring the instances to a given connectivity structure is not mentioned.

A popular tool for generating binary optimization problems with planted solutions is the *Chook* toolbox of Perera et al. (2021). Chook implements several approaches, such as "tile planting," "Wishart planting," "equation planting," and "k-local planting." However, none of those approaches guarantees uniqueness, and some of them (such as Wishart planting) are not designed to tailor to arbitrary connectivity structures. Notably, the method "deceptive cluster loops" is tailored to the D-Wave Chimera topology.

The software package *dwig* contains Python implementations of several existing planted solution methods, specifically, *RAN-pr* (Zdeborova and Krzakala, 2016), *RAN-k* (King J. et al., 2015), *FL-k* (King A. D. et al., 2015), *FCL-k* (King et al., 2017), *weak–strong cluster network* (Denchev et al., 2016), *frustrated cluster loops* (Albash and Lidar, 2018), and *corrupted biased ferromagnet* (Pang et al., 2021).

There are several studies which have examined the sampling of MAX 2-SAT combinatorial optimization problems using quantum annealing, some with an emphasis on generating MAX 2-SAT, which are challenging for quantum annealing to sample (Crosson et al., 2014; Santra et al., 2014; Hsu et al., 2018; Mehta et al., 2021, 2022; Mirkarimi et al., 2023).

The above methods have been used in a number of studies on sampling characteristics of quantum annealers (King J. et al., 2015; Zhang et al., 2017; Barash et al., 2019; Marshall et al., 2019).

2 Methods

This section introduces a novel method to generate QUBO models of the type of Equation (1) for a customized connectivity structure and with a guarantee of uniqueness for the planted solution. The method is based on the generation of a posiform representation of Equation (1), which is introduced in Section 2.1. The construction of the posiform and the guarantee of uniqueness are based on the fact that testing if a posiform attains the value zero is equivalent to a 2-SAT problem, which can be solved in polynomial time (Section 2.2). The complete algorithm is summarized in Section 2.3. A note on how the generation can naturally be adapted to a given connectivity structure is discussed in Section 2.4.

2.1 Conversion from QUBO to posiform

A *posiform* is a quadratic function with positive coefficients on an extended set of variables $\mathcal{Z} = \{x_1, \ldots, x_n\} \cup \{\bar{x}_1, \ldots, \bar{x}_n\}$, meaning that a posiform can contain either a variable $x_i \in \{0, 1\}$ or its complement $\bar{x}_i = 1 - x_i$, where $i \in \{1, \ldots, n\}$. A posiform can be expressed as

$$P(x_1, \ldots, x_n) = P(x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n) = \sum_{z \in \mathcal{Z}} b_z z + \sum_{z, z' \in \mathcal{Z}} b_{zz'} zz', \tag{2}$$

where each $z \in \mathcal{Z}$ and $z' \in \mathcal{Z}$ stand for one of the variables x_i or its complement $\bar{x}_i$, $i \in \{1, \ldots, n\}$ and the coefficients b_z and $b_{zz'}$ are non-negative.

Any QUBO of the form of Equation (1) can be written as a posiform. To this end, consider first the linear terms. If $a_i > 0$ for some $i \in \{1, \ldots, n\}$ in Equation (1), it remains unchanged in the posiform. If $a_i < 0$, we rewrite $a_i x_i = a_i(1 - \bar{x}_i) = a_i + (-a_i)\bar{x}_i$. The single summand a_i is constant and can be omitted as it does not impact the location of the minimum of Equation (1). The term $(-a_i)\bar{x}_i$ complies with the posiform requirement as $-a_i > 0$ given $a_i < 0$.

Similarly, any quadratic term $a_{ij}x_i x_j$ with $a_{ij} > 0$ in Equation (1) remains unchanged in the posiform. If $a_{ij}x_i x_j$ with $a_{ij} < 0$ in Equation (1), we rewrite it as either $a_{ij}(1 - \bar{x}_i)x_j = a_{ij} + (-a_{ij})\bar{x}_j + (-a_{ij})\bar{x}_i x_j$ or $a_{ij}x_i(1 - \bar{x}_j) = a_{ij} + (-a_{ij})\bar{x}_i + (-a_{ij})x_i\bar{x}_j$. Both options are valid choices and none is preferable over the other. As can be seen, apart from the constant term a_{ij}, which can be omitted, the remaining summands have positive coefficients $-a_{ij} > 0$ given $a_{ij} < 0$.

As a simple example, consider the following QUBO in three variables, $Q(x_1, x_2, x_3) = 2x_1 - x_2 + x_1x_2 - 2x_2x_3$. In posiform

representation, it can be written as $P(x_1, x_2, x_3) = 2x_1 + \bar{x}_2 + 2\bar{x}_3 + x_1x_2 + 2\bar{x}_2x_3$, where we omitted the offset -3 that results from the conversion.

2.2 Connection to 2-SAT problems

The idea of posiform planting is to generate posiforms that attain a value zero at a unique known (planted) combination of values of the variables. Assume a posiform of the type of Equation (2) is given. Clearly the minimum of Equation (2) is bounded below by zero as all coefficients and variables are non-negative. Moreover, we can test if there is a configuration $x = (x_1, \ldots, x_n)$ that achieves $P(x_1, \ldots, x_n) = 0$ in linear time.

This can be seen as follows. If $P(x_1, \ldots, x_n) = 0$, then all summands in Equation (2) must be zero. Therefore, we aim to find $x = (x_1, \ldots, x_n)$ such that $z = 0$ for all linear terms, and $zz' = 0$ for all quadratic terms in Equation (2), where $z, z' \in \mathcal{Z}$. For the quadratic terms, $zz' = 0$ is equivalent to $\bar{z} \vee \bar{z}' = True$. We thus rewrite all linear and quadratic terms in Equation (2) without their coefficients into a 2-SAT problem, which can be solved in linear time (Krom, 1967; Even et al., 1976; Aspvall et al., 1979). Any solution to the constructed 2-SAT problem will satisfy $P(x_1, \ldots, x_n) = 0$ and vice versa. Importantly, if the 2-SAT problem has a unique solution, so does the corresponding posiform.

2.3 QUBO generation with given connectivity and planted unique solution

We are given a bitstring $x^* = (x_1^*, \ldots, x_n^*)$, denoting the solution to be planted. The first step is to generate a 2-SAT problem having x^* as its unique solution. We aim to construct a 2-SAT problem having x^* as its unique solution with the help of an exclusion argument, meaning that we add clauses to the 2-SAT problem that exclude any bitstring other than x^*. This is achieved as follows.

We select two random indices $i, j \in \{1, \ldots, n\}$ with $i \neq j$ and consider the two bits x_i^* and x_j^* in the solution to be planted. We then randomly select one of the three possible binary tuples $(\hat{x}_i, \hat{x}_j)$ satisfying $(\hat{x}_i, \hat{x}_j) \neq (x_i^*, x_j^*)$. Depending on the choice of $(\hat{x}_i, \hat{x}_j)$, we add a clause to the current 2-SAT problem that excludes the possibility of $(x_i, x_j) = (\hat{x}_i, \hat{x}_j)$ in an optimal solution, precisely, the clause

$$\begin{aligned}
\neg(\bar{x}_i \wedge \bar{x}_j) &= (x_i \vee x_j) \qquad \text{if } (\hat{x}_i, \hat{x}_j) = (0,0), \\
\neg(\bar{x}_i \wedge x_j) &= (x_i \vee \bar{x}_j) \qquad \text{if } (\hat{x}_i, \hat{x}_j) = (0,1), \\
\neg(x_i \wedge \bar{x}_j) &= (\bar{x}_i \vee x_j) \qquad \text{if } (\hat{x}_i, \hat{x}_j) = (1,0), \\
\neg(x_i \wedge x_j) &= (\bar{x}_i \vee \bar{x}_j) \qquad \text{if } (\hat{x}_i, \hat{x}_j) = (1,1).
\end{aligned} \tag{3}$$

After each added clause, we attempt to solve the generated 2-SAT problem at its current stage. By construction, the choice of the clauses added to the 2-SAT problem will never exclude the planted bitstring x^* from the solution set of the generated 2-SAT problem.

We continue in this fashion until we arrive at a 2-SAT problem which has x^* as its unique solution. Our procedure only requires polynomial effort. Indeed, it is known that the phase transition in 2-SAT problems occurs for n variables at $O(n)$ clauses (Gent and Walsh, 1994; Coja-Oghlan and Panagiotou, 2016), thus we expect to only add a linear number of clauses until x^* remains as the unique solution of the 2-SAT problem. Moreover, solving a 2-SAT problem can be done in linear time (Krom, 1967; Even et al., 1976; Aspvall et al., 1979). Note that, to save computational effort, it is not necessary to solve the 2-SAT problem being generated each time a new clause is added. Instead, it suffices to solve it after adding a certain batch size $B \in \mathbb{N}$ of new clauses. In the experiments of Section 3, we use the *MiniSat* solver of Eén and Sörensson (2023).

Once a 2-SAT problem is constructed with x^* as its unique solution, we construct a posiform from it. Thus, in the second step, we convert each clause $(z \vee z') = \neg(\bar{z} \wedge \bar{z}')$ into the quadratic term $b_{zz'}\bar{z}\,\bar{z}'$, where $z, z' \in \mathcal{Z}$. The negation is necessary here as each clause $(z \vee z')$, that is, *True* (value 1) in the 2-SAT problem needs to be zero in the posiform (see Section 2.2) as it is a function to be minimized. Importantly, the coefficient $b_{zz'} > 0$ of the posiform is actually freely choosable (as long as it is positive). Substituting any complement $\bar{x}_i$ as $1 - x_i$ and multiplying out the expression yields a QUBO with (typically) both positive and negative QUBO coefficients.

As an example, suppose we aim to plant the solution $x^* = (1, 0, 1)$ in $n = 3$ variables. For the random indices $(i, j) = (2, 3)$, we choose $(\hat{x}_2, \hat{x}_3) = (1, 1)$, thus satisfying $(\hat{x}_2, \hat{x}_3) \neq (x_2^*, x_3^*)$. According to Equation (6), we add the clause $(\bar{x}_2 \vee \bar{x}_3)$ to the 2-SAT problem being generated. By continuing in this fashion for other randomly chosen variable pairs in x^*, we might obtain the 2-SAT instance

$$\begin{aligned}
(\bar{x}_2 \vee \bar{x}_3) \wedge (x_1 \vee \bar{x}_2) \wedge (x_1 \vee \bar{x}_3) \wedge (x_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3) \\
\wedge (\bar{x}_1 \vee x_3),
\end{aligned} \tag{4}$$

which can easily be checked to have the unique solution x^*. Rewriting Equation (4) into a posiform results in $P = x_2x_3 + \bar{x}_1x_2 + \bar{x}_1x_3 + \bar{x}_1\bar{x}_2 + x_2\bar{x}_3 + x_1\bar{x}_3$. Note that the coefficients of P (set here to 1) can be freely chosen as long as they are positive. Multiplying out the posiform leads to the QUBO $Q(x_1, x_2, x_3) = x_2 + x_3 - 2x_1x_3$, which can easily be verified to have a unique minimum at x^*.

2.4 Adaptation to connectivity structures

Apart from the guarantee of uniqueness, the algorithm of Section 2.3 allows one to adapt the generated QUBOs to a given connectivity structure. This is possible since there are no restrictions on the choice of tuples (x_i^*, x_j^*) with $i, j \in \{1, \ldots, n\}$ that are being used to narrow down the solution space to x^* in the 2-SAT problem.

To be precise, instead of sampling $i, j \in \{1, \ldots, n\}$, it is valid to sample $(i, j) \in \mathcal{E}$ for some edge set $\mathcal{E} \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$. When converting the generated 2-SAT problem to a posiform, the clauses become the quadratic terms, and when multiplying out the posiform into a QUBO, no further couplers are being introduced. Therefore, the edges in $\mathcal{E}$ will translate 1-to-1 to the quadratic couplers in the posiform and in the QUBO. For instance, $\mathcal{E}$ can be chosen as the fixed connectivity graph of one of the D-Wave annealer generations. Naturally, if $\mathcal{E}$ is too sparse, it might not be guaranteed any more that enough clauses can be sampled to narrow

TABLE 1 D-Wave quantum annealing processor summary.

D-wave QPU chip ID	Topology name	Available qubits	Available couplers	Annealing time (min, max) microseconds
`DW_2000Q_6`	Chimera C_{16}	2,041	5,974	(1, 2,000)
`Advantage_system4.1`	Pegasus P_{16}	5,627	40,279	(0.5, 2,000)
`Advantage_system6.1`	Pegasus P_{16}	5,616	40,135	(0.5, 2,000)
`Advantage2_prototype1.1`	Zephyr Z_4	563	4,790	(1, 2,000)

down x^* as the unique solution; however, this problem was not encountered for any of the D-Wave hardware graphs.

3 Results

In this section, we investigate the performance of the posiform planting methodology introduced in Section 2. The section starts with an overview of the D-Wave devices and their parameters in Section 3.1. In Section 3.2, we use posiform planting to generate and solve QUBO instances on four D-Wave machines that fit their hardware natively, thus allowing for very large instance sizes. The hardness of the generated instances is assessed by computing the ground state probability (GSP) and the time-to-solution (TTS) metrics. In Section 3.3, we investigate instances with arbitrary qubit connectivity, thus requiring a minor embedding of the problem QUBO onto the D-Wave hardware.

3.1 Parameter settings

Table 1 shows the four generations of the D-Wave quantum annealer used in the experiments of this section. Apart from the Chip ID and the name of the D-Wave topology, Table 1 displays the number of available qubits and couplers and the annealing times supported by the device.

The posiform planting method requires solving a 2-SAT problem repeatedly during the planting process to verify the uniqueness of the planted solution (see Section 2.3). For efficiency reasons, we add an initial batch of B clauses to the 2-SAT problem before starting to check for uniqueness. In Section 3.2, we employ the choice $B = 2,000$ for the Chimera hardware graph of `DW_2000Q_6`, $B = 30,000$ for the Pegasus hardware graph of `Advantage_system4.1` and `Advantage_system6.1`, and $B = 1,000$ for the Zephyr hardware graph of `Advantage2_prototype1.1`. In Section 3.3, we employ $B = 1$ to generate the 52 variable all-to-all graphs. These choices of B are arbitrary, they do not influence the uniqueness of the solution but the runtime of the generation process, and they were selected to correspond to the number of variables in the hardware graph. Similarly, the posiform coefficients can be chosen arbitrarily in posiform planting. We select the posiform coefficients from the set $\{1, 2\}$ for both the hardware native QUBOs and the minor embedded QUBOs, which depending on the hardware graph can result in highly variable QUBO coefficients after converting the posiform to a QUBO. However, the QUBO models can still be mapped onto the D-Wave hardware due to the auto coefficient scaling and the maximum energy scale that is programmable onto the chip. Choosing the posiform coefficients as integers also ensures that the QUBO coefficients will be integers. Visualizations of the hardware native QUBO coefficients can be found in Appendix 1.

The hardware native QUBOs in Section 3.2 are sampled using annealing times of 0.5 μs for the `Advantage_system6.1` and `Advantage_system4.1`, and in the range $\{1, 2, \ldots, 10\}$ and $\{20, 30, \ldots, 1,990, 2,000\}$ μs for all four D-Wave quantum annealers. Each hardware native QUBO is sampled using two D-Wave device calls, each having 400 anneal-readout cycles, resulting in a total of 800 measurements made per annealing time and per hardware native QUBO.

3.2 Results for hardware native QUBOs

We generate 100 unique QUBO problems tailored to the four D-Wave quantum annealers outlined in Table 1. Those are being solved as a function of the anneal time, using the D-Wave settings described in Section 3.1. Since the unique solution and thus the ground state of each QUBO is known, computing the ground state success probability (GSP) is straightforward.

Figure 1 shows the GSP for the hardware native QUBOs measured on the four D-Wave devices. Each subplot shows the results of the 100 randomly generated QUBOs on each device, with one line per QUBO visualizing probability of reaching the ground state (among the 800 anneals) as a function of the annealing time.

Several observations are noteworthy. Since the GSP is mostly non-zero, the D-Wave quantum annealers are able to sample the optimal solution during some anneal. This even holds true for QUBO instances with up to 5,627 variables in the case of `Advantage_system4.1`. Although it is difficult to see in the plots, at small annealing times, in particular, 500 ns and 1 μs, the two Pegasus chip devices fail to sample the optimal solution across all 100 problem instances.

We observe an increasing trend in the measured GSP as a function of the annealing time, but with diminishing returns as annealing time increases. The results show a difference in behavior between the four D-Wave devices. In particular, the 563 qubit system `Advantage2_prototype1.1` samples the optimal solution at a much higher rate than the other devices. This finding can be attributed to the fact that the number of variables on this device is less than on the other devices, while also being the newest generation of the D-Wave annealer with reported lower error rates than the previous generations.

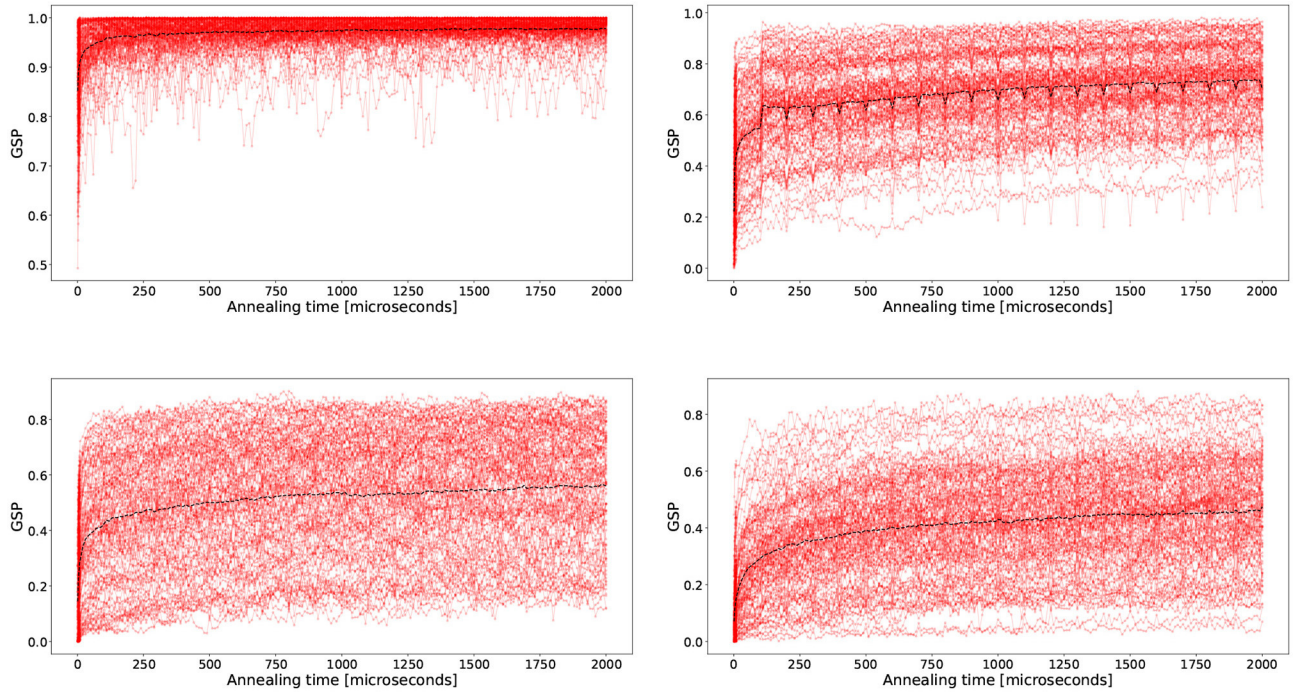


FIGURE 1
Ground state success probability (GSP) for hardware native QUBOs computed on the devices `Advantage2_prototype1.1` **(top left)**, `DW_2000Q_6` **(top right)**, `Advantage_system4.1` **(bottom left)**, and `Advantage_system6.1` **(bottom right)**. Each subplot corresponds to one D-Wave annealer and contains 100 separate lines which are showing GSP results for the 100 unique random hardware native posiform planted QUBOs. Each line shows the probability of reaching the ground state (among the 800 anneals) as a function of the annealing time. The dashed black line denotes the mean GSP computed at each evaluated annealing time.

We observe that the results for `DW_2000Q_6` in Figure 1 show periodic variations of the measured GSP. This is because the annealing time measurements in increments of 100 μs were made several weeks apart from the measurements made for all other annealing times in increments of 10 μs, and previous studies (Pelofske et al., 2023) have shown that there are long-term variations (in solution quality) of the computations carried out on current D-Wave quantum annealing devices. Therefore, the variations that have a periodicity of 100 μs are due to variance of the noise profile of the device, rather than variations that are a function of the annealing time.

Next, we examine the *time-to-solution* (*TTS*) metric for the 100 QUBO instances that were generated for each of the four D-Wave annealers. TTS is an estimate of the time it takes to reach an optimum solution with a 99% confidence. It is defined as

$$\mathrm{TTS}_{0.99} = \frac{\text{QPU-access-time}}{A} \cdot \frac{\log(1-0.99)}{\log(1-p)}, \tag{5}$$

where QPU-access-time (in seconds) is the real compute time used on the D-Wave backend (including the hardware programming time, anneal-readout cycle, and anneal times), A is the number of anneals and $p \in (0,1)$ is the success probability observed among the A anneals, that is, the proportion of anneals that found the ground state. The QPU-access-time also includes all communication time with the device on top of the annealing time used in the computation. When $p = 1$, we set $\mathrm{TTS}_{0.99} = \text{QPU-access-time}/A$. When $p = 0$, $\mathrm{TTS}_{0.99}$ is undefined, and therefore is not computed.

Figure 2 plots the TTS (computed with Equation 5) to reach the optimal planted solution for the set of 100 randomly generated hardware native QUBOs for each of the four D-Wave annealers. We observe that for `Advantage2_prototype1.1`, the lowest TTS is achieved for short annealing times, whereas for the other three generations of the D-Wave annealer, both low and high annealing times incur higher TTS values, with the lowest TTS being achieved in-between.

In addition to solving the 100 native hardware QUBOs sampled on each of the four D-Wave devices, we also investigate how successfully classical heuristics can solve them. Figure 3 shows histograms for the achieved GSP when sampling the same set of hardware native QUBO problems using the classical heuristics Simulated Annealing (SA), implemented in the function *neal* in the D-Wave SDK (D-Wave Systems, 2023b), and greedy Steepest Descent (analogous to steepest gradient descent), implemented in the function *greedy* in the D-Wave SDK (D-Wave Systems, 2023a). Figure 3 demonstrates that the SA algorithm, in particular, is able to find the optimal solution of the QUBOs generated with posiform planting with very high success probability.

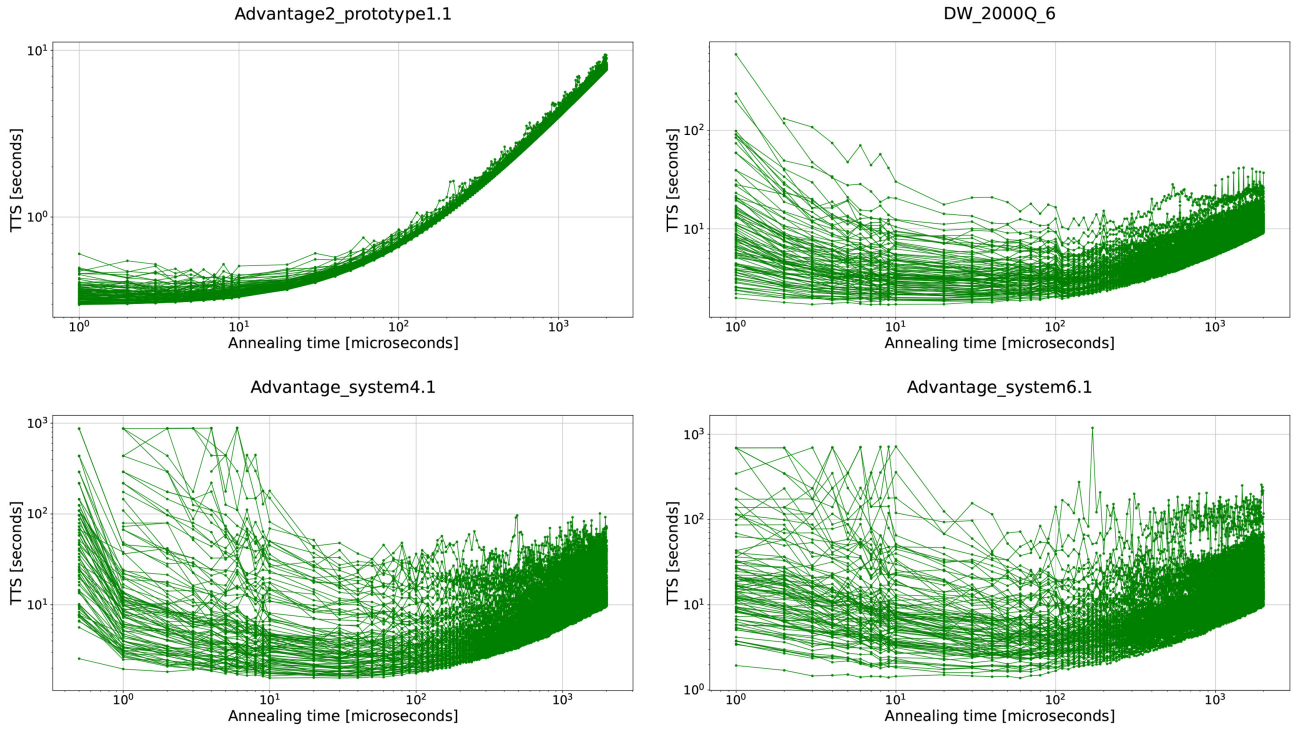


FIGURE 2
TTS as a function of the annealing time on the 100 hardware QUBO problems for each of the four D-Wave quantum annealers, in particular, `Advantage2_prototype1.1` **(top left)**, `DW_2000Q_6` **(top right)**, `Advantage_system4.1` **(bottom left)**, and `Advantage_system6.1` **(bottom right)**. One line per QUBO instance. Log scale on both axes.

3.3 Results for minor embedded QUBOs

Posiform planting as introduced in Section 2.3 generates new clauses to be added to the 2-SAT instance without any constraints on the indices. Although clauses are arbitrary, the generated QUBOs are usually not fully connected. Using generated QUBOs with all-to-all connectivity require a minor embedding onto the D-Wave quantum hardware before being solved, since the D-Wave hardware graphs are (relatively) sparse. Despite the challenges associated with minor-embedded QUBO instances, which require chained qubits and pose issues such as selecting appropriate chain strengths, utilizing such QUBOs enables a direct comparison of D-Wave devices on the same set of input problems. A diagram showing these complete minor embeddings on the four hardware graphs is shown in Figure A4 in the Appendix.

We generate five QUBOs with varying density with the aim to allow for a range of GSP rates among those planted QUBOs. Each QUBO instance has 52 logical variables, which is the largest problem size with an all-to-all connectivity that can be minor embedded on the `Advantage2_prototype1.1` device. Since `Advantage2_prototype1.1` has the smallest such embedding, the same QUBO instances are guaranteed to be executed on all four D-Wave quantum annealers, thereby allowing for a fair comparison. Note that these 52 variable QUBOs are not fully connected, but they are arbitrarily connected in that the generator can select arbitrary edges to include.

Figure 4 shows ground state success probability (GSP) measurements as a function of the chain strength, computed for the five fixed QUBO instances on the four D-Wave annealers of Table 1. Each subplot additionally showcases the behavior for different annealing times. The figure highlights several observations. First, the `DW_2000Q_6` device seems to achieve a considerably lower GSP than the other devices, followed by `Advantage_system4.1` and `Advantage_system6.1`, while `Advantage2_prototype1.1` achieves highest GSP across the instances. Second, the anneal times do influence the solution quality throughout all instances, with longer annealing times usually resulting in an increased solution quality. Third, although the 5 QUBO instances were generated with the same parameters, there seems to be a considerable range in difficulty, with the instances in the left columns being harder to solve than the ones in the rightmost columns.

4 Discussion

This study proposes a new method, called posiform planting, to generate QUBOs with a unique planted solution. Apart from guaranteeing the uniqueness, posiform planting can be adapted to any arbitrary connectivity structure, meaning that it allows one to generate tailored QUBO instances whose quadratic couplers fit, for instance, the hardware connectivity of modern quantum annealers. Therefore, posiform planting allows one to efficiently generate QUBO instances with thousands of variables and a unique planted solution. Posiform planting also generates QUBOs that have linear terms, a property that not all of existing planted solution methods have.

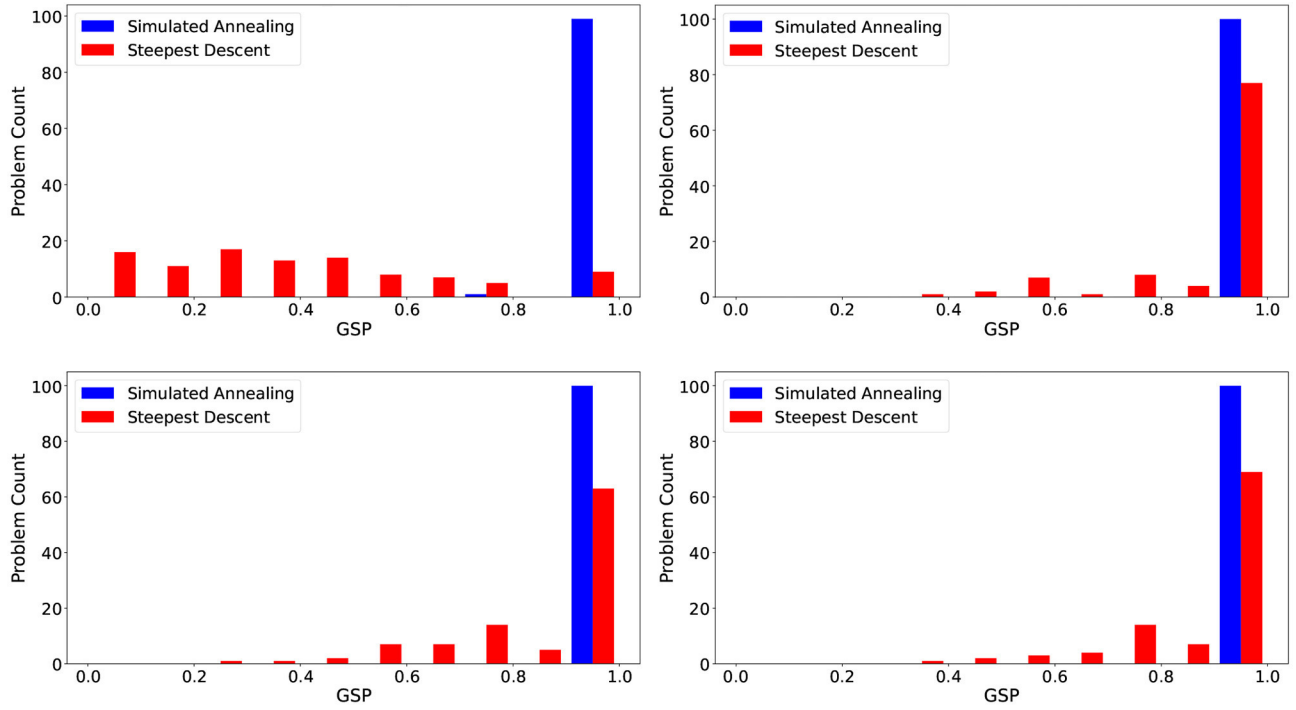


FIGURE 3
Histograms of the GSP for the same hardware native posiform planted QUBOs used in Figure 1 sampled using classical heuristics. Simulated annealing and steepest descent heuristics applied to the QUBOs generated for the hardware graphs of `DW_2000Q_6` **(top left)**, `Advantage2_prototype1.1` **(top right)**, `Advantage_system4.1` **(bottom left)**, and `Advantage_system6.1` **(bottom right)**. The side-by-side histogram bars correspond to each bin, so the sampling rates for simulated annealing are extremely high (usually at a proportion of 1).

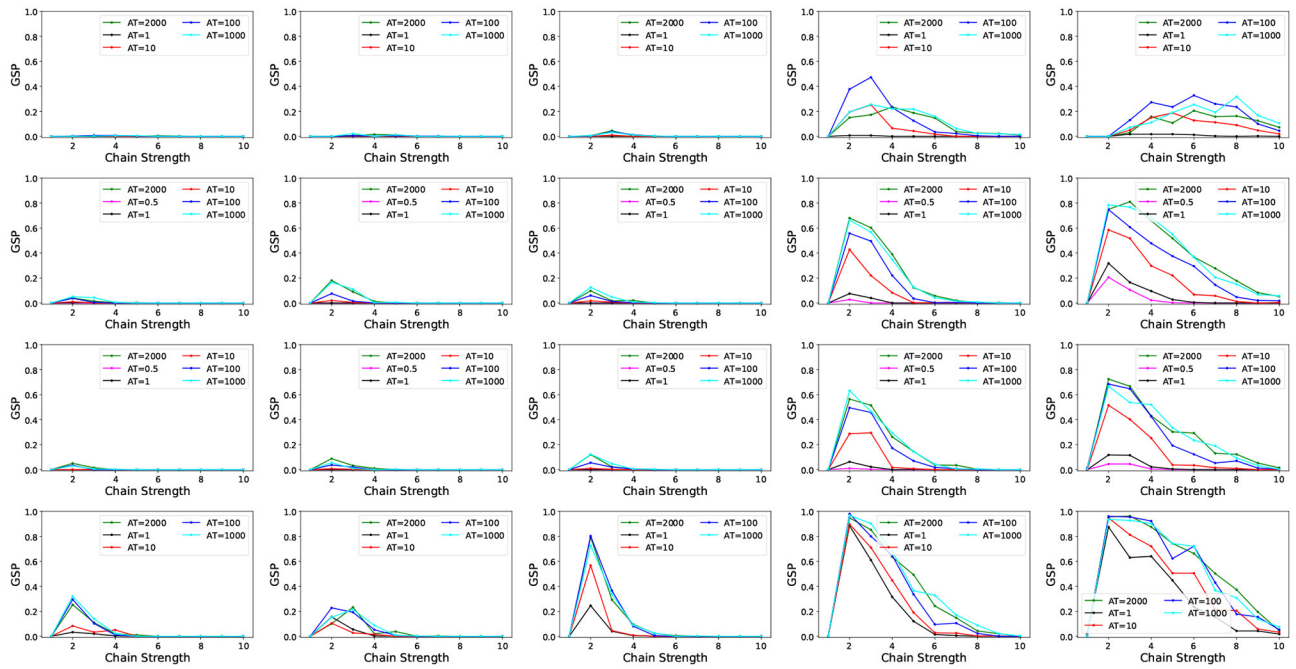


FIGURE 4
GSP measurements as a function of the chain strength for the five minor-embedded QUBO instances. The five columns correspond to the five QUBO instances being solved, and the four rows correspond to the four D-Wave quantum annealers (`DW_2000Q_6`, `Advantage_system4.1`, `Advantage_system6.1`, and `Advantage2_prototype1.1` from top to bottom). The annealing times are varied (see legends).

Interestingly, our construction shows that, for the generation of a QUBO with a planted solution, the coefficients in a posiform representation do not matter and can be freely chosen (as long as they are positive as required by definition of a posiform). Since the choice of the posiform coefficients does not impact the planted solution or its uniqueness, posiform planted allows for an efficient generation of a set of QUBO instances having the same planted solution. Posiform planting also allows for an arbitrary bitstring to be chosen as the planted solution.

Posiform planting can be used to verify whether good classical heuristic algorithms, such as simulated annealing, are able to find the single optimal solution for extremely large QUBOs. This not only applies to classical algorithms but also other emerging computing technologies such as spiking neuromorphic computing (Alom et al., 2017; Mniszewski, 2019) or the hybrid quantum-classical gate model algorithm QAOA (Farhi et al., 2014; Hadfield et al., 2019). We experimentally demonstrated that four D-Wave quantum annealers, with a total of three different classes of hardware graphs, can sample the unique planted solution for hardware native QUBO problems that use the entire hardware chip. Since we scaled our instances to the maximal size that can be embedded on D-Wave, the current hardware limitations (of maximally 5,627 qubits on D-Wave Advantage) somewhat limit us from scaling our instances to sizes where D-Wave starts to struggle.

Posiform planting generates QUBOs of a special form to guarantee the uniqueness of the planted solution. To be precise, all QUBOs generated by posiform planting have the property that when converted to a posiform representation, they are solvable (meaning they attain a value of zero). However, not all QUBOs have this property. Nevertheless, posiform planting is complete in the sense that it can generate any QUBO whose posiform representation is solvable.

The study leaves scope for further avenues of research. Most importantly, it remains to investigate if posiform planting allows one to tune the difficulty of the QUBO problems, for instance, via the choice of the posiform coefficients (which can be tuned without constraints other than being positive). Another topic for future research is to be able to vary the ground state degeneracy of posiform planted QUBOs, if there are specific use cases where obtaining a QUBO with a specific number of ground states would be advantageous.

Finally, posiform planting can enhance an existing planted solution method, denoted as M, to guarantee the uniqueness of the planted solution. For instance, given the desired solution x^* to be planted, we first use M to produce a QUBO Q_1 with the planted solution x^*, which may be non-unique. Subsequently, leveraging posiform planting, we generate a QUBO Q_2 that ensures x^* is a unique optimal solution. By forming the linear combination $Q_{\text{new}} = \alpha_1 Q_1 + \alpha_2 Q_2$ with $\alpha_1, \alpha_2 > 0$, we obtain a problem with a unique solution x^*, while potentially preserving any desired properties of M.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found at: https://doi.org/10.5281/zenodo.8336707.

Author contributions

GH: Conceptualization, Investigation, Methodology, Software, Writing—original draft, Writing—review & editing, Supervision, Validation, Formal analysis. EP: Data curation, Investigation, Methodology, Software, Visualization, Writing—original draft, Writing—review & editing, Validation, Resources. HD: Methodology, Project administration, Supervision, Writing—review & editing, Investigation.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This study was supported by the U.S. Department of Energy through the Los Alamos National Laboratory. Los Alamos National Laboratory was operated by Triad National Security, LLC, for the National Nuclear Security Administration of U.S. Department of Energy (with Contract No. 89233218CNA000001). Research presented in this article was supported by the NNSA's Advanced Simulation and Computing Beyond Moore's Law Program at Los Alamos National Laboratory, and was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project 20210114ER. This research used resources provided by the Los Alamos National Laboratory Institutional Computing Program, which was supported by the U.S. Department of Energy National Nuclear Security Administration under Contract No. 89233218CNA000001. This research used resources provided by the Darwin testbed at Los Alamos National Laboratory (LANL), which was funded by the Computational Systems and Software Environments subprogram of LANL's Advanced Simulation and Computing program (NNSA/DOE). This study has been assigned LANL technical report number LA-UR-23-27170. The study of HD was supported by grant number KP-06-DB/1 of the Bulgarian National Science Fund. Funding for GH was provided through Cure Alzheimer's Fund, the National Institutes of Health [1R01 AI 154470-01; 2U01 HG 008685; R21 HD 095228 008976; U01 HL 089856; U01 HL 089897; P01 HL 120839; P01 HL 132825; 2U01 HG 008685; R21 HD 095228, and P01HL132825], the National Science Foundation [NSF PHY 2033046 and NSF GRFP 1745302], and a NIH Center grant [P30-ES002109].

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of

their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2023.1275948/full#supplementary-material

References

Albash, T., and Lidar, D. A. (2018). Demonstration of a scaling advantage for a quantum annealer over simulated annealing. *Phys. Rev. X* 8:031016. doi: 10.1103/PhysRevX.8.031016

Alom, M., Van Essen, B., Moody, A., Widemann, D., and Taha, T. M. (2017). "Quadratic unconstrained binary optimization (QUBO) on neuromorphic computing system," in *2017 International Joint Conference on Neural Networks (IJCNN)* (IEEE), 3922–3929. doi: 10.1109/IJCNN.2017.7966350

Aspvall, B., Plass, M., and Tarjan, R. (1979). A linear-time algorithm for testing the truth of certain quantified Boolean formulas. *Inform. Process. Lett.* 8, 121–123. doi: 10.1016/0020-0190(79)90002-4

Barash, L., Marshall, J., Weigel, M., and Hen, I. (2019). Estimating the density of states of frustrated spin systems. *N. J. Phys.* 21:073065. doi: 10.1088/1367-2630/ab2e39

Barthel, W., Hartmann, A. K., Leone, M., Ricci-Tersenghi, F., Weigt, M., and Zecchina, R. (2002). Hiding solutions in random satisfiability problems: a statistical mechanics approach. *Phys. Rev. Lett.* 88:188701. doi: 10.1103/PhysRevLett.88.188701

Boixo, S., Albash, T., Spedalieri, F. M., Chancellor, N., and Lidar, D. A. (2013). Experimental signature of programmable quantum annealing. *Nat. Commun.* 4:2067. doi: 10.1038/ncomms3067

Boixo, S., Rønnow, T. F., Isakov, S. V., Wang, Z., Wecker, D., Lidar, D. A., et al. (2014). Evidence for quantum annealing with more than one hundred qubits. *Nat. Phys.* 10, 218–224. doi: 10.1038/nphys2900

Boixo, S., Smelyanskiy, V. N., Shabani, A., Isakov, S. V., Dykman, M., Denchev, V. S., et al. (2016). Computational multiqubit tunnelling in programmable quantum annealers. *Nat. Commun.* 7:10327. doi: 10.1038/ncomms10327

Boothby, K., Bunyk, P., Raymond, J., and Roy, A. (2020). Next-generation topology of d-wave quantum processors. *arXiv [Preprint]. arXiv: 2003.00133.*

Boros, E., Hammer, P., and Tavares, G. (2006). Preprocessing of unconstrained quadratic binary optimization. *Rutcor Res. Rep.* 10, 1–58.

Boros, E., Hammer, P., and Tavares, G. (2007). Local search heuristics for Quadratic Unconstrained Binary Optimization (QUBO). *J. Heurist.* 13, 99–132. doi: 10.1007/s10732-007-9009-3

Choi, V. (2008). Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. *Quant. Inform. Process.* 7, 193–209. doi: 10.1007/s11128-008-0082-9

Choi, V. (2011). Minor-embedding in adiabatic quantum computation: II. Minor-universal graph design. *Quant. Inform. Process.* 10, 343–353. doi: 10.1007/s11128-010-0200-3

Coja-Oghlan, A., and Panagiotou, K. (2016). The asymptotic k-SAT threshold. *Adv. Math.* 288, 985–1068. doi: 10.1016/j.aim.2015.11.007

Crosson, E., Farhi, E., Lin, C. Y.-Y., Lin, H.-H., and Shor, P. (2014). Different strategies for optimization using the quantum adiabatic algorithm. *arXiv [Preprint]. arXiv: 1401.7320.*

Das, A., and Chakrabarti, B. K. (2008). Colloquium: Quantum annealing and analog quantum computation. *Rev. Modern Phys.* 80:1061. doi: 10.1103/RevModPhys.80.1061

Dattani, N., Szalay, S., and Chancellor, N. (2019). Pegasus: the second connectivity graph for large-scale quantum annealing hardware. *arXiv [Preprint]. arXiv: 1901.07636.*

Denchev, V. S., Boixo, S., Isakov, S. V., Ding, N., Babbush, R., Smelyanskiy, V., et al. (2016). What is the computational value of finite-range tunneling? *Phys. Rev. X* 6:031015. doi: 10.1103/PhysRevX.6.031015

D-Wave Systems (2023a). *Dwave-Greedy: An Implementation of a Steepest Descent Solver for Binary Quadratic Models.* Available online at: https://github.com/dwavesystems/dwave-greedy

D-Wave Systems (2023b). *Dwave-Neal: An Implementation of a Simulated Annealing Sampler*. Available online at: https://github.com/dwavesystems/dwave-neal

Eén, N., and Sörensson, N. (2023). *MiniSat Solver*. Available online at: http://minisat.se

Even, S., Itai, A., and Shamir, A. (1976). On the complexity of time table and multi-commodity flow problems. *SIAM J. Comput.* 5, 691–703. doi: 10.1137/0205048

Farhi, E., Goldstone, J., and Gutmann, S. (2014). A quantum approximate optimization algorithm. *arXiv [Preprint]. arXiv: 1411.4028.*

Gent, I. P., and Walsh, T. (1994). "The SAT phase transition," in *ECAI'94: Proceedings of the 11th European Conference on Artificial Intelligence* (PITMAN), 105–109.

Grant, E., and Humble, T. S. (2022). Benchmarking embedded chain breaking in quantum annealing. *Quant. Sci. Technol.* 7:025029. doi: 10.1088/2058-9565/ac26d2

Hadfield, S., Wang, Z., O'Gorman, B., Rieffel, E., Venturelli, D., and Biswas, R. (2019). From the quantum approximate optimization algorithm to a quantum alternating operator Ansatz. *Algorithms* 12:34. doi: 10.3390/a12020034

Hahn, G., Pelofske, E., and Djidjev, H. (2023). *Dataset for Posiform Planting: Generating QUBO Instances for Benchmarking [Data set]*. Zenodo. doi: 10.5281/zenodo.8336707

Harris, R., Sato, Y., Berkley, A., Reis, M., Altomare, F., Amin, M., et al. (2018). Phase transitions in a programmable quantum spin glass simulator. *Science* 361, 162–165. doi: 10.1126/science.aat2025

Hauke, P., Katzgraber, H. G., Lechner, W., Nishimori, H., and Oliver, W. D. (2020). Perspectives of quantum annealing: methods and implementations. *Rep. Prog. Phys.* 83:054401. doi: 10.1088/1361-6633/ab85b8

Hen, I. (2019). Equation planting: a tool for benchmarking Ising machines. *Phys. Rev. Appl.* 12:011003. doi: 10.1103/PhysRevApplied.12.011003

Hen, I., Job, J., Albash, T., Rønnow, T., Troyer, M., and Lidar, D. (2015). Probing for quantum speedup in spin-glass problems with planted solutions. *Phys. Rev. A* 92:042325. doi: 10.1103/PhysRevA.92.042325

Hsu, T.-J., Jin, F., Seidel, C., Neukart, F., Raedt, H. D., and Michielsen, K. (2018). Quantum annealing with anneal path control: application to 2-sat problems with known energy landscapes. *Commun. Comput. Phys.* 26, 928–946. doi: 10.4208/cicp.OA-2018-0257

Kadowaki, T., and Nishimori, H. (1998). Quantum annealing in the transverse Ising model. *Phys. Rev. E* 58, 5355–5363. doi: 10.1103/PhysRevE.58.5355

King, A. D., Lanting, T., and Harris, R. (2015). Performance of a quantum Annealer on range-limited constraint satisfaction problems. *arXiv [Preprint]. arXiv: 1502.02098.*

King, A. D., Raymond, J., Lanting, T., Harris, R., Zucca, A., Altomare, F., et al. (2023). Quantum critical dynamics in a 5,000-qubit programmable spin glass. *Nature* 617, 61–66. doi: 10.1038/s41586-023-05867-2

King, A. D., Raymond, J., Lanting, T., Isakov, S. V., Mohseni, M., Poulin-Lamarre, G., et al. (2021). Scaling advantage over path-integral Monte Carlo in quantum simulation of geometrically frustrated magnets. *Nat. Commun.* 12, 1–6. doi: 10.1038/s41467-021-20901-5

King, A. D., Suzuki, S., Raymond, J., Zucca, A., Lanting, T., Altomare, F., et al. (2022). Coherent quantum annealing in a programmable 2,000 qubit Ising chain. *Nat. Phys.* 18, 1324–1328. doi: 10.1038/s41567-022-01741-6

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

King, J., Yarkoni, S., Raymond, J., Ozfidan, I., King, A. D., Nevisi, M. M., et al. (2017). Quantum annealing amid local ruggedness and global frustration. *J. Phys. Soc. Jpn.* 88:061007. doi: 10.7566/JPSJ.88.061007

Kirkpatrick, S., Gelatt, C. D. Jr, and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science* 220, 671–680. doi: 10.1126/science.220.4598.671

Könz, M. (2019). *Embedding penalties for quantum hardware architectures and performance of simulated quantum annealing* (Ph.D. thesis). ETH Zurich, Zurich, Switzerland.

Könz, M. S., Lechner, W., Katzgraber, H. G., and Troyer, M. (2021). Embedding overhead scaling of optimization problems in quantum annealing. *PRX Quant.* 2:040322. doi: 10.1103/PRXQuantum.2.040322

Könz, M. S., Mazzola, G., Ochoa, A. J., Katzgraber, H. G., and Troyer, M. (2019). Uncertain fate of fair sampling in quantum annealing. *Phys. Rev. A* 100:030303. doi: 10.1103/PhysRevA.100.030303

Kowalsky, M., Albash, T., Hen, I., and Lidar, D. (2022). 3-regular three-XORSAT planted solutions benchmark of classical and quantum heuristic optimizers. *Quant. Sci. Technol.* 7:025008. doi: 10.1088/2058-9565/ac4d1b

Krom, M. (1967). The decision problem for a class of first-order formulas in which all disjunctions are binary. *Zeitschr. Math. Log. Grund. Math.* 13, 15–20. doi: 10.1002/malq.19670130104

Krzakala, F., and Zdeborová, L. (2009). Hiding quiet solutions in random constraint satisfaction problems. *Phys. Rev. Lett.* 102:238701. doi: 10.1103/PhysRevLett.102.238701

Kumar, V., Tomlin, C., Nehrkorn, C., O'Malley, D., and Dulny, J. (2020). Achieving fair sampling in quantum annealing. *arXiv [Preprint]. arXiv: 2007.08487.*

Lanting, T., Amin, M. H., Baron, C., Babcock, M., Boschee, J., Boixo, S., et al. (2020). Probing environmental spin polarization with superconducting flux qubits. *arXiv [Preprint]. arXiv: 2003.14244.*

Lanting, T., Przybysz, A., Smirnov, A., Spedalieri, F., Amin, M., Berkley, A., et al. (2014). Entanglement in a quantum annealing processor. *Phys. Rev. X* 4:21041. doi: 10.1103/PhysRevX.4.021041

Lucas, A. (2014). Ising formulations of many NP problems. *Front. Phys.* 2:5. doi: 10.3389/fphy.2014.00005

Mandrà, S., Zhu, Z., and Katzgraber, H. G. (2017). Exponentially biased ground-state sampling of quantum annealing machines with transverse-field driving Hamiltonians. *Phys. Rev. Lett.* 118:070502. doi: 10.1103/PhysRevLett.118.070502

Marshall, J., Mossi, G., and Rieffel, E. G. (2022). Perils of embedding for quantum sampling. *Phys. Rev. A* 105:022615. doi: 10.1103/PhysRevA.105.022615

Marshall, J., Venturelli, D., Hen, I., and Rieffel, E. G. (2019). Power of pausing: advancing understanding of thermalization in experimental quantum Annealers. *Phys. Rev. Appl.* 11:044083. doi: 10.1103/PhysRevApplied.11.044083

Matsuda, Y., Nishimori, H., and Katzgraber, H. G. (2009). Quantum annealing for problems with ground-state degeneracy. *J. Phys.* 143:012003. doi: 10.1088/1742-6596/143/1/012003

Mehta, V., Jin, F., Raedt, H. D., and Michielsen, K. (2021). Quantum annealing with trigger hamiltonians: application to 2-satisfiability and nonstoquastic problems. *Phys. Rev. A* 104:032421. doi: 10.1103/PhysRevA.104.032421

Mehta, V., Jin, F., Raedt, H. D., and Michielsen, K. (2022). Quantum annealing for hard 2-satisfiability problems: distribution and scaling of minimum energy gap and success probability. *Phys. Rev. A* 105:062406. doi: 10.1103/PhysRevA.105.062406

Mirkarimi, P., Callison, A., Light, L., Chancellor, N., and Kendon, V. (2023). Comparing the hardness of max 2-sat problem instances for quantum and classical algorithms. *Phys. Rev. Res.* 5:023151. doi: 10.1103/PhysRevResearch.5.023151

Mniszewski, S. (2019). "Graph partitioning as quadratic unconstrained binary optimization (QUBO) on spiking neuromorphic hardware," in *Proceedings of the International Conference on Neuromorphic Systems, ICONS '19* (New York, NY: Association for Computing Machinery), 1–5. doi: 10.1145/3354265.3354269

Morita, S., and Nishimori, H. (2008). Mathematical foundation of quantum annealing. *J. Math. Phys.* 49:125210. doi: 10.1063/1.2995837

Nelson, J., Vuffray, M., Lokhov, A. Y., Albash, T., and Coffrin, C. (2022). High-quality thermal Gibbs sampling with quantum annealing hardware. *Phys. Rev. Appl.* 17:044046. doi: 10.1103/PhysRevApplied.17.044046

Nelson, J., Vuffray, M., Lokhov, A. Y., and Coffrin, C. (2021). Single-qubit fidelity assessment of quantum annealing hardware. doi: 10.1109/TQE.2021.3092710

Pang, Y., Coffrin, C., Lokhov, A. Y., and Vuffray, M. (2021). The potential of quantum annealing for rapid solution structure identification. *Constraints* 26, 1–25. doi: 10.1007/s10601-020-09315-0

Pearson, A., Mishra, A., Hen, I., and Lidar, D. A. (2019). Analog errors in quantum annealing: doom and hope. *NPJ Quant. Inform.* 5:107. doi: 10.1038/s41534-019-0210-7

Pei, Y., Manukian, H., and Di Ventra, M. (2020). Generating weighted MAX-2-SAT instances with frustrated loops: an RBM case study. *J. Mach. Learn. Res.* 21, 1532–4435.

Pelofske, E., Golden, J., Bärtschi, A., O'Malley, D., and Eidenbenz, S. (2021). "Sampling on NISQ devices: 'Who's the fairest one of all?'," in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 207–217. doi: 10.1109/QCE52317.2021.00038

Pelofske, E., Hahn, G., and Djidjev, H. (2023). Noise dynamics of quantum annealers: estimating the effective noise using idle qubits. *Quant. Sci. Technol.* 8:035005. doi: 10.1088/2058-9565/accbe6

Perera, D., Akpabio, I., Hamze, F., Mandrà, S., Rose, N., Aramon, M., et al. (2021). Chook-A comprehensive suite for generating binary optimization problems with planted solutions. *arXiv[Preprint]. arXiv:2005.14344.*

Perera, D., Hamze, F., Raymond, J., Weigel, M., and Katzgraber, H. (2020). Computational hardness of spin-glass problems with tile-planted solutions. *Phys. Rev. E* 101:023316. doi: 10.1103/PhysRevE.101.023316

Santra, S., Quiroz, G., Steeg, G. V., and Lidar, D. A. (2014). Max 2-SAT with up to 108 qubits. *N. J. Phys.* 16:045006. doi: 10.1088/1367-2630/16/4/045006

Tasseff, B., Albash, T., Morrell, Z., Vuffray, M., Lokhov, A. Y., Misra, S., et al. (2022). On the emerging potential of quantum annealing hardware for combinatorial optimization. *arXiv [Preprint]. arXiv: 2210.04291.*

Venturelli, D., Mandrà, S., Knysh, S., O'Gorman, B., Biswas, R., and Smelyanskiy, V. (2015). Quantum optimization of fully connected spin glasses. *Phys. Rev. X* 5:31040. doi: 10.1103/PhysRevX.5.031040

Wang, W., Mandrà, S., and Katzgraber, H. (2017). Patch-planting spin-glass solution for benchmarking. *Phys. Rev. E* 96:023312. doi: 10.1103/PhysRevE.96.023312

Zaborniak, T., and de Sousa, R. (2021). Benchmarking Hamiltonian noise in the d-wave quantum annealer. *IEEE Trans. Quant. Eng.* 2, 1–6. doi: 10.1109/TQE.2021.3050449

Zdeborova, L., and Krzakala, F. (2016). Statistical physics of inference: thresholds and algorithms. *Adv. Phys.* 65, 453–552. doi: 10.1080/00018732.2016.1211393

Zhang, B. H., Wagenbreth, G., Martin-Mayor, V., and Hen, I. (2017). Advantages of unfair quantum ground-state sampling. *Sci. Rep.* 7:1044. doi: 10.1038/s41598-017-01096-6

frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 07 December 2023
DOI 10.3389/fcomp.2023.1285962

OPEN ACCESS

EDITED BY
Catherine McGeoch,
D-Wave Systems, Canada

REVIEWED BY
Michael Hanks,
Imperial College London, United Kingdom
Jemma Bennett,
University of Innsbruck, Austria

*CORRESPONDENCE
Yuto Ishikawa
ishikawa.yuto.f6@s.mail.nagoya-u.ac.jp
Takuma Yoshihara
21ef106@ms.dendai.ac.jp
Keita Okamura
6223024@ed.tus.ac.jp

RECEIVED 30 August 2023
ACCEPTED 09 November 2023
PUBLISHED 07 December 2023

CITATION
Ishikawa Y, Yoshihara T, Okamura K and Ohzeki M (2023) Individual subject evaluated difficulty of adjustable mazes generated using quantum annealing.
Front. Comput. Sci. 5:1285962.
doi: 10.3389/fcomp.2023.1285962

Individual subject evaluated difficulty of adjustable mazes generated using quantum annealing

Yuto Ishikawa[1]*, Takuma Yoshihara[2]*, Keita Okamura[3]* and Masayuki Ohzeki[4,5,6]

[1]Department of Computer Science, Nagoya University, Nagoya, Japan, [2]Department of Engineering, Tokyo Denki University, Adachi, Japan, [3]Department of Physics and Astronomy, Tokyo University of Science, Noda, Japan, [4]Graduate School of Information Sciences, Tohoku University, Sendai, Japan, [5]Department of Physics, Tokyo Institute of Technology, Meguro, Japan, [6]Sigma-i Co., Ltd., Shinagawa, Japan

In this study, the maze generation using quantum annealing is proposed. We reformulate a standard algorithm to generate a maze into a specific form of a quadratic unconstrained binary optimization problem suitable for the input of the quantum annealer. To generate more difficulty mazes, we introduce an additional cost function Q_{update} to increase the difficulty. The difficulty of the mazes was evaluated by the time to solve the maze of 12 human subjects. To check the efficiency of our scheme to create the maze, we investigated the time-to-solution of a quantum processing unit, classical computer, and hybrid solver. The results show that Q_{update} generates difficult mazes tailored to the individual. Furthermore, it show that the quantum processing unit is more efficient at generating mazes than other solvers. Finally, we also present applications how our results could be used in the future.

KEYWORDS

quantum annealing, combinatorial optimization, maze generation, bar-tipping algorithm, time-to-solution

1 Introduction

A combinatorial optimization problem is minimizing or maximizing their cost or objective function among many variables that take discrete values. In general, it takes time to solve the combinatorial optimization problem. To deal with many combinatorial optimization problems, we utilize generic solvers to solve them efficiently. Quantum annealing (QA) is one of the generic solvers for solving combinatorial optimization problems (Kadowaki and Nishimori, 1998) using the quantum tunneling effect. Quantum annealing is a computational technique to search for good solutions to combinatorial optimization problems by expressing the objective function and constraint time requirements of the combinatorial optimization problem by quantum annealing in terms of the energy function of the Ising model or its equivalent QUBO (Quadratic Unconstrained Binary Optimization) and manipulating the Ising model and QUBO to search for low energy states (Shu Tanaka and Seki, 2022). Various applications of QA are proposed in traffic flow optimization (Neukart et al., 2017; Hussain et al., 2020; Inoue et al., 2021), finance (Rosenberg et al., 2016; Orús et al., 2019; Venturelli and Kondratyev, 2019), logistics (Feld et al., 2019; Ding et al., 2021), manufacturing (Venturelli et al., 2016; Haba et al., 2022; Yonaga et al., 2022), preprocessing in material experiments (Tanaka et al., 2023), marketing (Nishimura et al., 2019), steel manufacturing (Yonaga et al., 2022), and decoding problems (Ide et al., 2020; Arai et al., 2021a). The model-based Bayesian optimization is also proposed in the literature (Koshikawa et al., 2021). A comparative study of quantum annealer was

performed for benchmark tests to solve optimization problems (Oshiyama and Ohzeki, 2022). The quantum effect on the case with multiple optimal solutions has also been discussed (Yamamoto et al., 2020; Maruyama et al., 2021). As the environmental effect cannot be avoided, the quantum annealer is sometimes regarded as a simulator for quantum many-body dynamics (Bando et al., 2020; Bando and Nishimori, 2021; King et al., 2022). Furthermore, applications of quantum annealing as an optimization algorithm in machine learning have also been reported (Neven et al., 2012; Amin et al., 2018; Khoshaman et al., 2018; Kumar et al., 2018; OMalley et al., 2018; Arai et al., 2021b; Sato et al., 2021; Urushibata et al., 2022; Goto and Ohzeki, 2023; Hasegawa et al., 2023). In this sense, developing the power of quantum annealing by considering hybrid use with various techniques is important, as mentioned in several previous studies (Hirama and Ohzeki, 2023; Takabayashi and Ohzeki, 2023).

In this study, we propose the generation of the maze by quantum annealing. In the application of quantum annealing to mazes, algorithms for finding the shortest path through a maze have been studied (Pakin, 2017). Automatic map generation is an indispensable technique for game production, including rogue-like games. Maze generation has been used to construct random dungeons in rogue-like games by assembling mazes (mok Bae et al., 2015). Therefore, considering maze generation as one of the rudiments of this technology, we studied maze generation using a quantum annealing machine. Several algorithms for the generation of the maze have been proposed. In this study, we focused on maze-generating algorithms. One can take the bar-tipping algorithm (Alg, 2023a), the wall-extending algorithm (Alg, 2023b), and the hunt-and-kill algorithm (Alg, 2023c).

The bar-tipping algorithm is an algorithm that generates a maze by extending evenly spaced bars one by one. For the sake of explanation, we will explain the terminology here. A path represents an empty traversable part of the maze and a bar a filled non-traversable part. Figure 1 shows where the outer wall, bars, and coordinate (i, j) are in a 3×3 maze. The maze is surrounded by an outer wall, as shown in Figure 1. It requires the following three constraints. First, each bar can be extended by one cell only in one direction. Second, the first column can be extended in four directions: up, down, left, and right, while the second and subsequent columns can be extended only in three directions: up, down, and right. Third, adjacent bars cannot overlap each other. We explain the detailed process of the bar-tipping algorithm using the 3×3 size maze. In this study, a maze generated by extending the $N \times N$ bars is called $N \times N$ size maze. First, standing bars are placed in every two cells in a field surrounded by an outer wall, as shown in Figure 1. Second, Figure 2 shows each step of bar-tipping algorithm. Figure 2A shows the first column of bars extended. The bars in the first column are randomly extended in only one direction with no overlaps, as shown in Figure 2A. The bars can be extended in four directions (up, down, right, and left) at this time. Figure 2B shows the second column of bars being extended. Third, the bars in the second column are randomly extended in one direction without overlap, as shown in Figure 2B. The bars can be extended in three directions (up, down, and right) at this time. Figure 2C shows the state in which the bars after the second column are extended. Fourth, the bars in subsequent columns are randomly

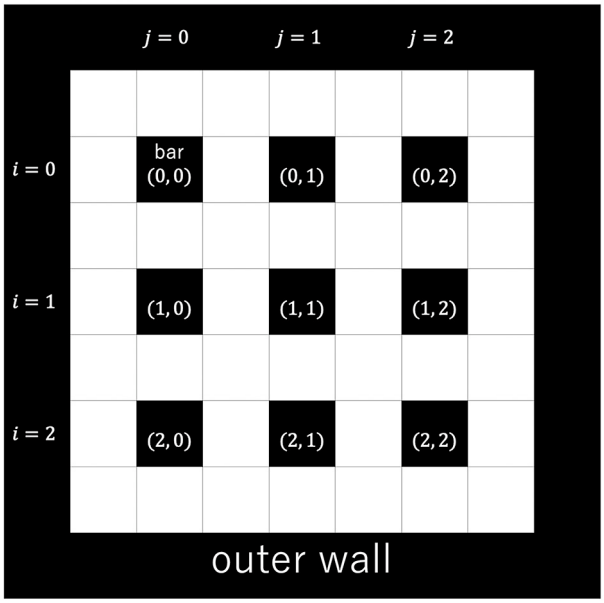


FIGURE 1
Positions where outer wall, bars, and coordinate (i, j) are in 3×3 maze.

extended in one direction, likewise the bars in the second column, as shown in Figure 2. Figure 2D shows the complete maze in its finished state. Following the process, we can generate a maze, as shown in Figure 2D.

If multiple maze solutions are possible, the maze solution is not unique, simplifying the time and difficulty of reaching the maze goal. These constraints must be followed for the reasons described below. The first constraint prevents a maze from generating a maze with multiple maze solutions and closed circuits. Figure 3A shows a maze state that violates the first constraint. The step violating the first constraint because one bar in the upper right corner is extended in two directions, as shown Figure 3A. The second constraint prevents generating a maze from a maze with closed circuits and multiple maze solutions. Figure 3B shows a state that violates the second constraint. The second constraint is violated, it has a closed circuit and multiple maze solutions, as shown in Figure 3B. The third constraint prevents maze generation from a maze with multiple maze solutions. Figure 3C shows a state that violates the third constraint. The bars overlap in the upper right corner, making it the third constraint as Figure 3C.

Next, we describe the wall-extending algorithm. It is an algorithm that generates a maze by extending walls. Figure 4 shows the extension starting coordinates of the wall-extending algorithm. Figure 5A shows the initial state of the wall-extending algorithm. First, as an initial condition, the outer perimeter of the maze is assumed to be the outer wall, and the rest of the maze is assumed to be the path, as shown in Figure 5A. Coordinate system is different from the bar-tipping algorithm, and all cells are labeled coordinates. As shown in Figure 4, the coordinates where both x and y are even and not walls are listed as starting coordinates for wall extending. The following process is repeated until all starting coordinates

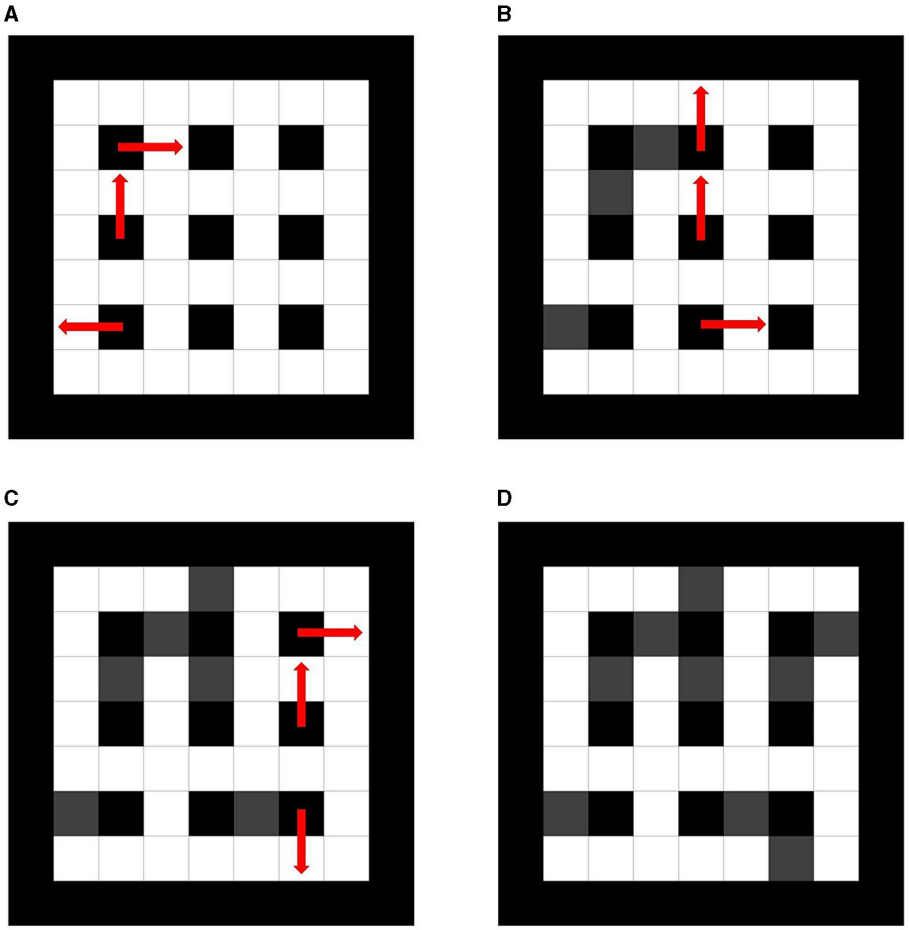


FIGURE 2
Step of bar-tipping algorithm. **(A)** Step1: bars in first column are extended. **(B)** Step2: bars in second column are extended. **(C)** Step3: bars in subsequent column are extended. **(D)** Step4: a complete maze through these steps.

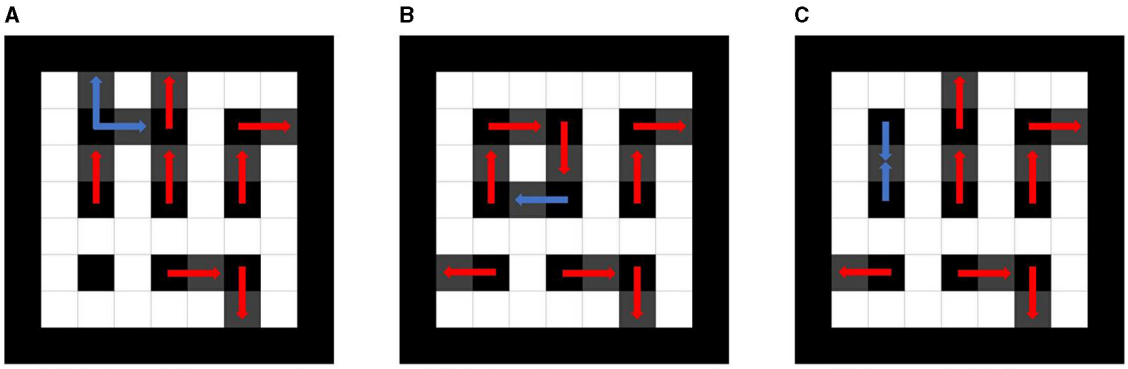


FIGURE 3
Mazes violated the constraints. **(A)** A maze violate the first constraint. **(B)** A maze violate the second constraint. **(C)** A maze violated the third constraint.

change to walls, as shown in Figure 5C. The coordinates were randomly chosen from the non-wall extension start coordinates. The next extending direction is randomly determined from which the adjacent cell is a path. Figure 5B shows how the path is extended. The extension will be repeated while two cells ahead of the extending direction to be extended are a path, as shown in Figure 5B. Figure 5C shows all starting coordinates changed to walls. These processes are repeated until all the starting coordinates

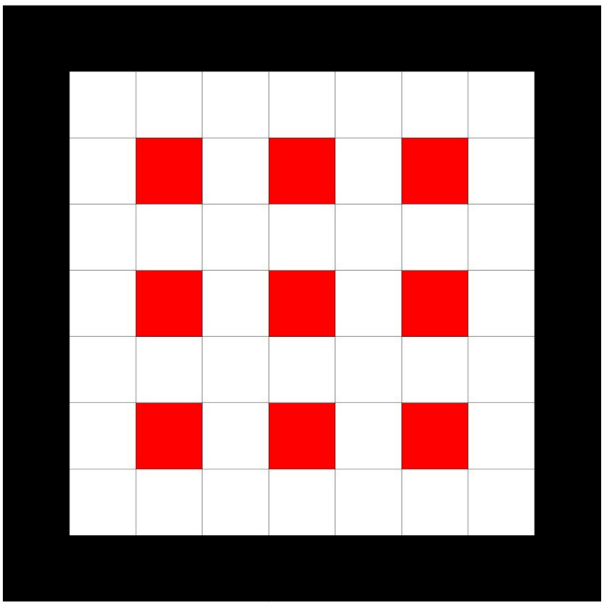


FIGURE 4
Red cells represent options of starting coordinates for the wall-extending algorithm.

change to walls, as shown in Figure 5C. Figure 5D shows a maze created by wall-extending. Following the process, we can generate a maze, as shown in Figure 5D.

As a third, the hunt-and-kill algorithm is explained below. It is an algorithm that generates a maze by extending paths. Figure 6 shows the extension starting coordinates of the hunt-and-kill algorithm. Figure 7A shows the initial state of the hunt-and-kill algorithm. The entire surface is initially walled off, as shown in Figure 7A. Coordinates, where both x and y are odd, are listed as starting coordinates for path extension, as shown in Figure 6. As with the wall-extending algorithm, all cells are set to coordinates. Figure 7B shows the state in which the path is extended. A coordinate is chosen randomly from the starting coordinates, and the path is extended from there, as shown in Figure 7B. Figure 7C shows the coordinate selection and re-extension after the path can no longer be extended. If the path can no longer be extended, a coordinate is randomly selected from the starting coordinates, which are already paths, and extension starts again from it, as shown in Figure 7C. This process is repeated until all the starting coordinates turn into paths to generate the maze. Figure 7D shows the complete maze with the hunt-and-kill algorithm. Following the process, we can generate a maze, as shown in Figure 7D.

Of the three maze generation algorithms mentioned above, the bar-tipping algorithm is relevant to the combinatorial optimization problem. In addition, unlike other maze generation algorithms, the bar-tipping algorithm is easy to apply because it only requires the consideration of adjacent elements. Thus, we have chosen to deal with this algorithm. Other maze generation algorithms could be generalized by reformulating them as combinatorial optimization problems. The wall-extending and hunt-and-kill algorithms will be implemented in future studies, considering the following factors. The former algorithm introduces the rule that adjacent walls are extended and so are their walls. The number of connected components will be computed for the latter, and the result will be included in the optimization.

Using the bar-tipping algorithm, we reformulated it to solve a combinatorial optimization problem that generates a maze with a longer solving time and optimizes it using quantum annealing. Quantum annealing (DW_2000Q_6 from D-Wave), classical computing (simulated annealing, simulated quantum annealing, and algorithmic solution of the bar-tipping algorithm), and hybrid computing were compared with each other according to the generation time of mazes, and their performance was evaluated. The solver used in this experiment is as follows: DW_2000Q_6 from D-Wave, simulated annealer called SASampler and simulated quantum annealer called SQASampler from OpenJij (Ope, 2023), D-Wave's quantum-classical hybrid solver called hybrid_binary_quadratic_model_version2 (BQM), and classical computer [MacBook Pro(14-inch, 2021), OS: macOS Monterey Version 12.5, Chip: Apple M1 Pro, Memory: 16 GB]. This comparison showed that quantum annealing was faster. This may be because the direction of the bars is determined at once using quantum annealing, which is several times faster than the classical algorithm. We do not use an exact solver to solve the combinatorial optimization problem. We expect some diversity in the optimal solution and not only focus on the optimal solution in maze generation. Thus, we compare three solvers, which generate various optimal solutions.

In addition, we generate mazes that reflect individual characteristics, whereas existing maze generation algorithms rely on randomness and fail to incorporate other factors. In this case, we incorporated the maze solution time as one of the other factors to solve the maze. The maze solving time was defined as the time (in seconds) from the start of solving the maze to the end of solving the maze.

The study is organized as follows. In the next Section, we explain the methods of our experiments. In Section 3, we describe the results of our experiments. In Section 4, we summarize this study.

2 Methods

2.1 Cost function

To generate the maze by quantum annealer, we need to set the cost function in the quantum annealer. One of the important features of the generation of the maze is diversity. In this sense, the optimal solution is not always unique. Since it is sufficient to obtain a structure consistent with a maze, the cost function is mainly derived from the necessary constraints of a maze, as explained below. Three constraints describe the basis of the algorithm of the bar-tipping algorithm. The cost function will be converted to a QUBO matrix to use the quantum annealer. To convert the cost function to a QUBO, the cost function must be written in a quadratic form. Using the penalty method, we can convert various constraints written

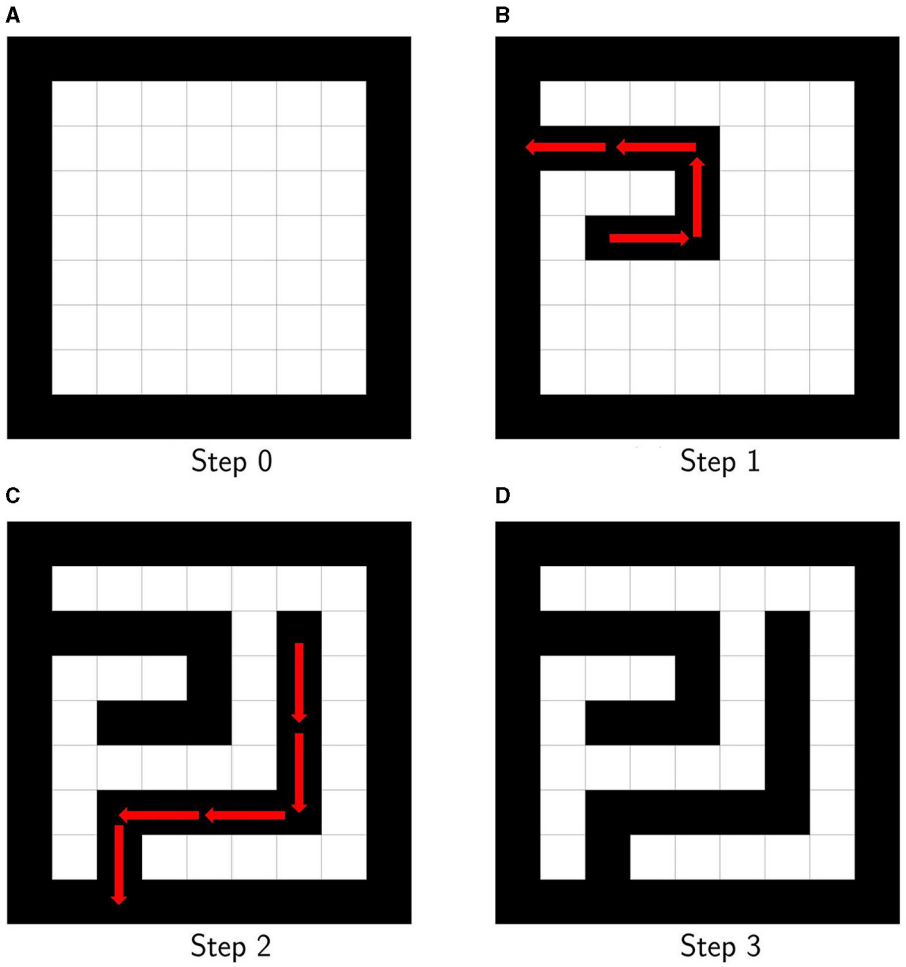


FIGURE 5
(A) Initial state for wall-extending algorithm. **(B)** Step 1 for wall-extending algorithm. **(C)** Step 2 for wall-extending algorithm. **(D)** Maze generated using wall-extending algorithm.

in a linear form into a quadratic function. The penalty method is a method to rewrite the equality constant as a quadratic function. For example, the penalty method can rewrite an equation constant $x = 1$ to $(x-1)^2$. Thus, we construct the cost function for generating the maze using the bar-tipping algorithm below.

The constraints of the bar-tipping algorithm correlate with each term in the cost function as described below. The first constraint of the bar-tipping algorithm is that the bars can be extended in only one direction. It prevents making closed circuits. The second constraint of the bar-tipping algorithm is that the bars of the first column be extended randomly in four directions (up, right, down, and left), and the second and subsequent columns can be extended randomly in three directions (up, right, and down). It also prevents the creation of closed circuits. The third constraint of the bar-tipping algorithm is that adjacent bars must not overlap. Following the constraint in the bar-tipping algorithm, we can generate a maze with only one path from the start to the goal.

The cost function consists of three terms to reproduce the bar-tipping algorithm, according to the three constraints, and determine the start and goal.

$$\begin{aligned} E(\{x_{i,j,d}, X_{m,n}\}) = &\sum_{i,i'}\sum_{j,j'}\sum_{d,d'} Q_{(i,j,d),(i',j',d')} x_{i,j,d} x_{i',j',d'} \\ &+\lambda_1 \sum_{i}\sum_{j}\left(\sum_{d} x_{i,j,d} - 1\right)^2 \\ &+\lambda_2 \left(\sum_{m}\sum_{n} X_{m,n} - 2\right)^2, \end{aligned} \tag{1}$$

where $x_{i,j,d}$ denotes whether the bar in i-th row, j-th column extended in direction d (up: 0, right: 1, down: 2, left: 3). When the bar in coordinate (i, j) is extended in direction, $x_{i,j,d}$ takes 1, otherwise takes 0. Due to the second constraint of the bar-tipping algorithm, the bars after the second column cannot be extended on the left side; only the first column has $(d = 3)$. Furthermore, $Q_{(i,j,d)(i',j',d')}$ in Equation (1) depends on i, j, d, i', j', and d' and is

10.3389/fcomp.2023.1285962

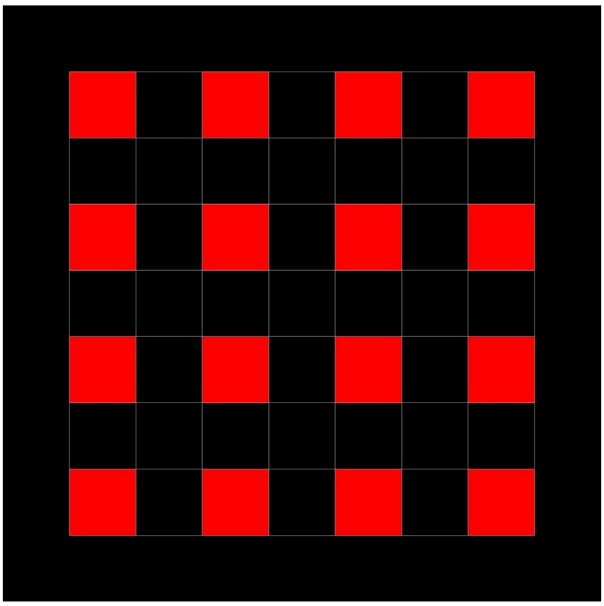


FIGURE 6
Red cells represent options of starting coordinates for the hunt-and-kill algorithm.

expressed as follows

$$Q_{(i,j,d),(i',j',d')} = \begin{cases} 1 \ (i = i' - 1, j = j', d = 2, d' = 0) \\ 1 \ (i = i' + 1, j = j', d = 0, d' = 2) \\ 0 \ \text{otherwise.} \end{cases} \quad (2)$$

The coefficients λ_1 and λ_2 are constants to adjust the effects of each penalty term. The first term prevents the bars from overlapping and extending each other face-to-face. It represents the third constraint of the bar-tipping algorithm. Here, due to the second constraint, bars in the second and subsequent columns cannot be extended to the left. Therefore, the adjacent bars in the same row cannot extend and overlap. This corresponds to the fact that d cannot take 3 when $j \geq 1$. Thus, there is no need to reflect, considering the left and right. In particular, the first term restricts the extending and overlapping between the up and down adjacent bars. For example, the situation in which one bar in (i, j) extended down ($d = 2$) and the lower bar in $(i + 1, j)$ extended up ($d = 0$) are represented by $x_{i,j,2}x_{i+1,j,0} = 1$ and $Q_{(i,j,2),(i+1,j,0)}$ take 1. In the same way, thinking of the relation between the bar in (i, j) and the upper bar in $(i - 1, j)$, $Q_{(i,j,0),(i-1,j,2)} = 1$. Thus, $Q_{(i,j,0),(i-1,j,2)}x_{i,j,0}x_{i-1,j,2}$ takes 1, and the value of the cost function taken will increase. By doing this, the third constraint is represented as a first term. The second term is a penalty term that limits the direction of extending to one per bar. It represents the first constraint of the bar-tipping algorithm. This means that for a given coordinate (i, j), the sum of $x_{i,j,d}$ $[d = 0, 1, 2(, 3)]$ must take the value 1. Here, the bars in the second and subsequent columns cannot extend to the left by the second constraint. Thus, d takes (0, 1, 2, 3) when $j = 0$, and d takes (0, 1, 2) when $j \geq 1$. The third term is the penalty term for selecting two coordinates of the start and the goal from the coordinates (m, n). This means that for a given coordinate (m, n), the sum of $X_{m,n}$ takes 2. In other words, two coordinates were selected as the start and the goal. The start and the goal are commutative in the maze. They are randomly selected from the two coordinates determined by the third term. $X_{m,n}$ denotes whether or not to set the start and goal at the m-th row and n-th column of options of start and goal coordinates. When the (m, n) coordinate is chosen as the start and goal, $X_{m,n}$ takes 1. Otherwise, it takes 0. There are no relations between $X_{m,n}$ and $x_{i,j,d}$ in Equation (1). This means that the maze structure and the start and goal determination coordinates have no relations. Figure 8 shows the coordinates (m, n) that are the options of the start and the goal. As Figure 8 shows, (m, n) is different from the coordinate setting bars; it is located at the four corners of the bars, where the bars do not extend. $X_{m,n}$ and $x_{i,j,d}$ are different. $X_{m,n}$ are options of start and goal, and $x_{i,j,d}$ are options of coordinates and directions to extend the bars. We have shown the simplest implementation of the maze generation following the bar-tipping algorithm by quantum annealer. Following the above a maze depending on randomness is generated. To generate a unique maze independent of randomness, we add the effect to make the maze more difficult in the cost function, and the difficulty is defined in terms of time (in seconds).

2.2 Update rule

We propose an additional Q_{update} term to increase the time to solve the maze. We introduce a random term that takes random elements to change the maze structure. It is added to the Equation (1). First, Q_{update} term, the additional term which includes the new QUBO matrix Q_{update}, is given by

$$\begin{aligned} &\lambda_{update1} \sum_{i,i'} \sum_{j,j'} \sum_{d,d'} Q_{update(k,k')} x_{i,j,d} x_{i',j',d'} \\ &+\lambda_{update1} \sum_{i} \sum_{j} \sum_{d} \sum_{m} \sum_{n} Q_{update(k,l)} x_{i,j,d} X_{m,n} \\ &+\lambda_{update1} \sum_{i} \sum_{j} \sum_{d} \sum_{m} \sum_{n} Q_{update(l,k)} X_{m,n} x_{i,j,d} \\ &+\lambda_{update2} \sum_{m,m'} \sum_{n,n'} Q_{update(l,l')} X_{m,n} X_{m',n'}, \end{aligned} \quad (3)$$

where

$$\begin{cases} k = d + (3N + 1)i & (j = 0) \\ k = d + 3j + 1 + (3N + 1)i & (j \neq 0) \\ l = (3N + 1)N + (N + 1)m + n. \end{cases} \quad (4)$$

Figure 9 shows the structure of Q_{update} and roles. Here, k', l' are the replacement of i, j, m, n, and d in k, l with i', j', m', n', and d'. N in Equation (4) is the size of the maze. The coefficients $\lambda_{update1}$ and $\lambda_{update2}$ are constants to adjust the effect of terms. The elements of Q_{update} related to the relation between the start and goal determination and the maze generation. This is located in part B, C in Figure 9 and multiplied by the $\lambda_{update1}$. The elements of Q_{update} related to the relation between the start and goal determination and the maze generation, part B, C in Figure 9 is multiplied by the $\lambda_{update1}$. The elements of Q_{update} related to the start and goal determination, part D in Figure 9 is multiplied by the $\lambda_{update2}$.

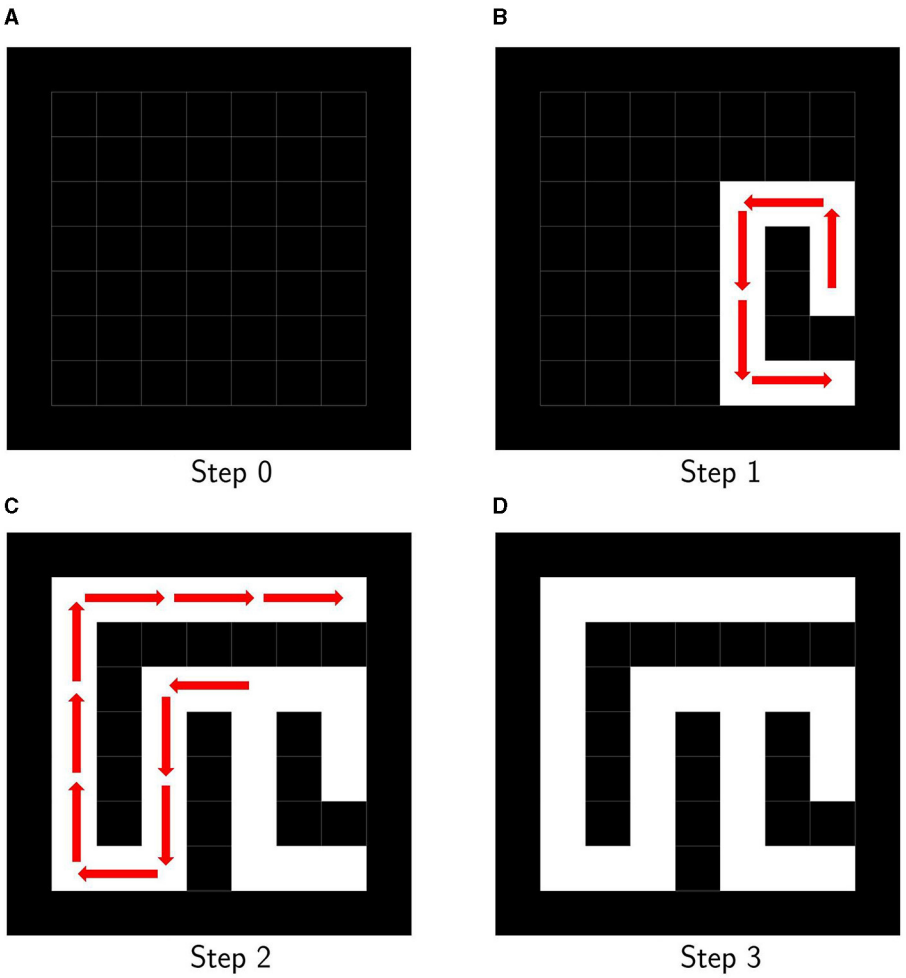


FIGURE 7
(A) Initial state for hunt-and-kill algorithm. **(B)** Step 1 for hunt-and-kill algorithm. **(C)** Step 2 for hunt-and-kill algorithm. **(D)** Maze generated using hunt-and-kill algorithm.

These are to control the maze difficulty without breaking the bar-tipping algorithm's constraints. Equation (3) is represented by the serial number k of each coordinate (i, j) at which bars can extend, and the sum l of the total number of coordinates at which the bars can extend and the serial number of coordinates (m, n), which are options for the start and the goal. Furthermore, The second term and the third term in Equation (3) allow the maze to consider the relation between the structure of the maze and the coordinates of the start and the goal.

Second, Q_{update}, the new QUBO matrix, is given by

$$Q_{update} \coloneqq p(t)Q_{update} + \{1 - p(t)\}Q_{random}, \tag{5}$$

where Q_{random} is a matrix of random elements from -1 to 1 and $p(t)$ depends on time t (in seconds) taken to solve the previous maze and is expressed as follows:

$$p(t) = \frac{1}{1 + e^{-at}}. \tag{6}$$

The Q_{update} is a matrix that was made with the aim of increasing the maze solving time through the maze solving iteration. The initial Q_{update} used in the first maze generation is a random matrix, and the next Q_{update} that is used in the second or subsequent maze generation is updated using Equation (5), the maze solving time t, and the previous Q_{update}. The longer the solving time t of the maze is, the higher the percentage of the previous Q_{update} in the current Q_{update} and the lower the percentage of Q_{random}; inversely, when t is small, the ratio of the previous Q_{update} is small, and the percentage of Q_{random} is significant. In other words, the longer the solving time t of the previous maze, the more characteristics of the previous term Q_{update} remain. Here, a is a constant to adjust the percentage. The $p(t)$ is a function that increases monotonically with t and takes 0 to 1. Thus, Q_{random}, which the random element in Q_{update}, increase as time t increases. After the maze is solved, the next maze QUBO is updated by Equation (5) using the time taken to solve the maze. The update is carried out only once before the maze generation. Repetition of the update will make the maze gradually difficult for individuals.

The sum of Equations (1) and (3) is always used to generate a new maze annealing from a maximally mixed state.

2.3 Experiments

2.3.1 Generation of maze

We generate mazes by optimizing the cost function using DW_2000Q_6. Since the generated maze will not be solved, the update term is excluded for this experiment. $\lambda_1 = 2$ and $\lambda_2 = 2$ were chosen.

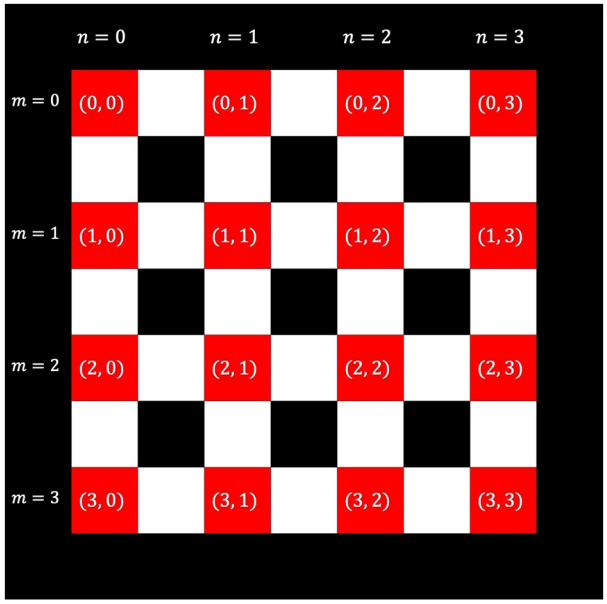


FIGURE 8
Black cells represent outer walls and inner bars (i, j). Red cells represent options of start and goal coordinates (m, n).

2.3.2 Computational cost

We compare the generation times of $N \times N$ maze in DW_2000Q_6 from D-Wave, simulated annealer called SASampler and simulated quantum annealer called SQASampler from OpenJij, D-Wave's quantum-classical hybrid solver called hybrid_binary_quadratic_model_version2 (hereinafter referred to as "Hybrid Solver") and classical computer [MacBook Pro(14-inch, 2021), OS: macOS Monterey Version 12.5, Chip: Apple M1 Pro, Memory: 16 GB] based on bar-tipping algorithm coded with Python 3.11.5 (hereinafter referred to as "Classic"). The update term was excluded from this experiment. We set $\lambda_1 = 2$ and $\lambda_2 = 2$. DW_2000Q_6 was annealed 1,000 times for 20 μs, and its QPU annealed time for maze generation as calculated using time-to-solution (TTS). SASampler and SQASampler were annealed with 1,000 sweeps. These parameters were constant throughout this experiment. Regression curves fitted using the least squares method were drawn from the results to examine the dependence of computation time on maze size.

2.3.3 Effect of update term

The solving time of 9×9 maze generated without Q_{update} and using Q_{update} was measured. This experiment asked 12 human subjects to solve mazes one set (30 times). To prevent the players from memorizing maze structure, they can only observe the limited 5×5 cells. In other words, only two surrounding cells can be observed. The increase rate from the first step of simple moving average of 10 solving times was plotted on the graph. For this experiment, $\lambda_1 = 2$, $\lambda_2 = 2$, $\lambda_{update1} = 0.15$, $\lambda_{update2} = 0.30$, and $a = 0.05$ were chosen. For two λ_{update}, we chose larger values that do not violate the constraints of the bar-tipping algorithm. We chose a value in which Equation (6) will be ∼0.8 (80%) when $t = 30$ s as a constant a.

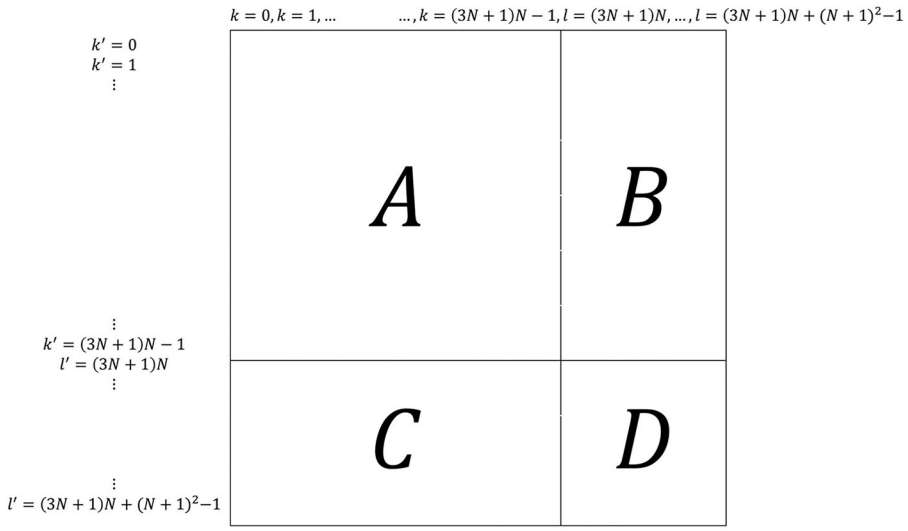


FIGURE 9
Structure of Q_{update}. Part A is related to maze generation. Part B and part C are related to the relation between maze generation and the start and goal determination. Part D is related to the start and goal determination.

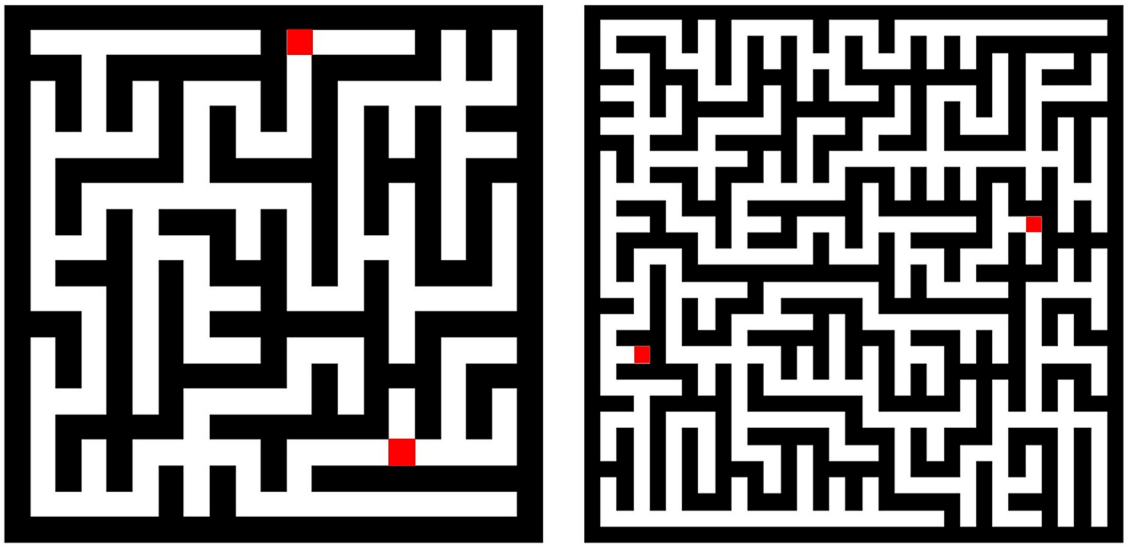


FIGURE 10
(Left) 9 × 9 maze generated by DW_2000Q_6. **(Right)** 15 × 15 maze generated by DW_2000Q_6. Red cells represent a start and a goal for the maze.

2.4 Applicatons

The cost function in this study has many potential applications by generalizing it. For example, it can be applied to graph coloring and traffic light optimization. Graph coloring can be applied by allowing adjacent nodes to have different colors. Traffic light optimization can address the traffic light optimization problem by looking at the maze generation as traffic flow. Roughly speaking, our cost function can be applied to the problem of determining the next state by looking at adjacent states.

Q_{update} can be applied to the problem of determining the difficulty of the next state from the previous result. The selection of personalized educational materials is one of the examples. Based on the solving time of the previously solved problems, the educational materials can be selected at a difficulty suitable for the individual. This is the most fascinating direction in future studies. As described above, we should emphasize that Q_{update} proposed in this study also has potential use in various fields related to training and education.

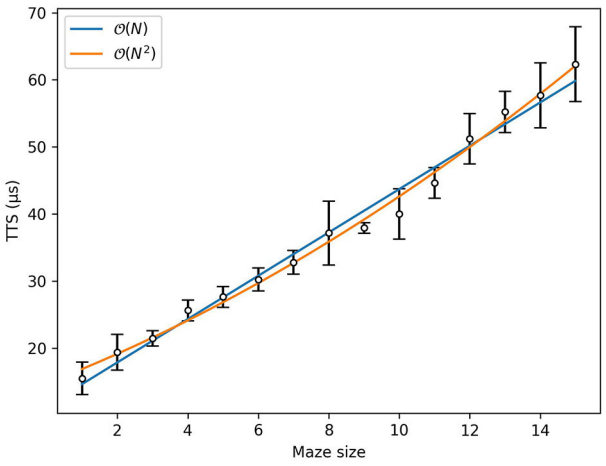


FIGURE 11
Time to reach the ground state with 99% success probability as a function of the maze size in DW_2000Q_6. The error bars represent a 95% confidence interval. The regression curve is given by $[(3.231 \pm 0.076)N + (11.40 \pm 0.69)]$ for linear regression and $[(7.4 \pm 1.8) \cdot 10^{-2}N^2 + (2.05 \pm 0.30)N + (14.8 \pm 1.0)]$ for quadratic regression.

3 Results

3.1 Generation of maze

Figure 10 shows execution examples of 9 × 9 and 15 × 15 mazes generated by optimizing the cost function using DW_2000Q_6.

3.2 Computational cost

Fits of the form $aN^2 + bN + c$ is applied to each of the datasets using the least squares method. The results are as follows. Figure 11 shows the relation between TTS for maze generation and maze size on DW_2000Q_6. DW_2000Q_6 is $\mathcal{O}(N)$ or $\mathcal{O}(N^2)$. Even if it is quadratically dependent on the maze size, its deviation is smaller than the other solvers. Figure 12 shows the relation between maze generation time and maze size on Classic, SASampler, and SQASampler. Classic $\left[(0.855 \pm 0.090)N^2 + (0.6 \pm 1.5)N + (2.2 \pm 5.1)\right]$, SASampler $\left[(28.8 \pm 1.2)N^2 + (36 \pm 20)N + (129 \pm 71)\right]$, and SQASampler $\left[(172.8 \pm 4.4)N^2 + (287 \pm 73)N - (1.5 \pm 2.5) \cdot 10^2\right]$ exhibit quadratic dependence on the maze size $\mathcal{O}(N^2)$. Most of the

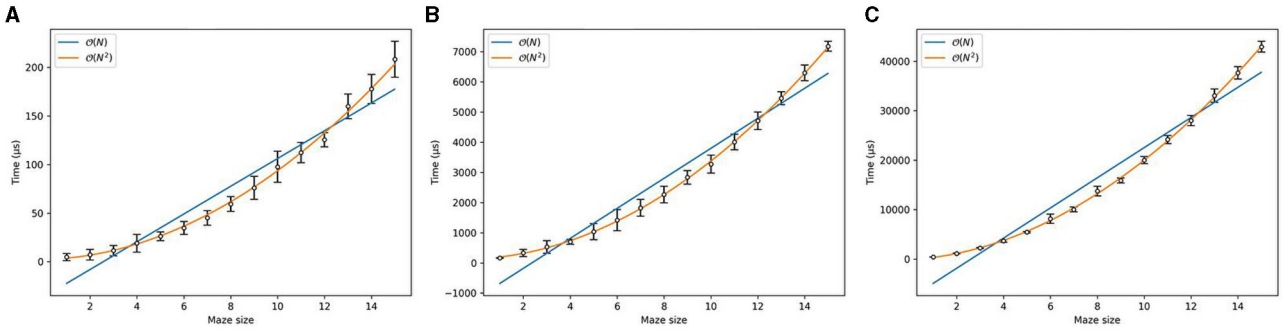


FIGURE 12
(A) The time to reach the ground state as a function of the maze size in Classic. The error bars represent a 95% confidence interval. The regression curve is $[(0.855 \pm 0.090)N^2 + (0.6 \pm 1.5)N + (2.2 \pm 5.1)]$. **(B)** Time to reach the ground state as a function of the maze size in SASampler. The error bars represent a 95% confidence interval. The regression curve is $[(28.8 \pm 1.2)N^2 + (36 \pm 20)N + (129 \pm 71)]$. **(C)** Time to reach the ground state as a function of the maze size in SQASampler. The error bars represent a 95% confidence interval. The regression curve is $[(172.8 \pm 4.4)N^2 + (287 \pm 73)N - (1.5 \pm 2.5) \cdot 10^2]$.

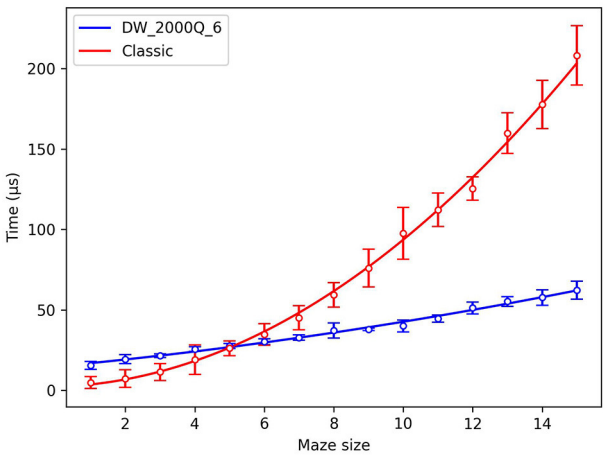


FIGURE 13
Comparison of maze generation time between DW_2000Q_6 and classic.

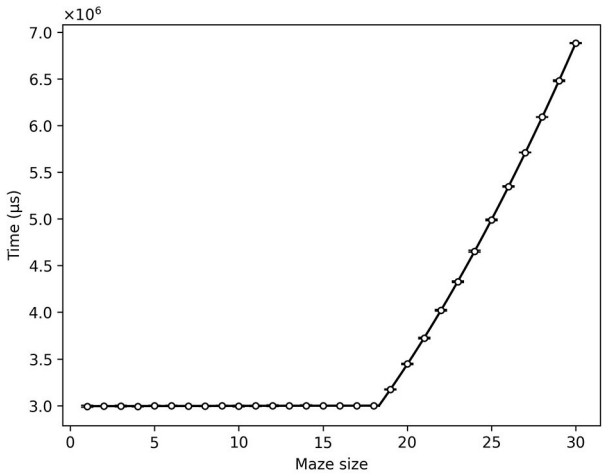


FIGURE 14
Time to reach the ground state as a function of maze size in the Hybrid Solver. The error bars represent a 95% confidence interval.

solvers introduced here are $\mathcal{O}(N^2)$ since they are extending $N \times N$ bars to generate a maze. Figure 13 shows the comparison of maze generation time between DW_2000Q_6 and Classic. DW_2000Q_6 has a smaller coefficient N^2 than the classical algorithm, and after $N = 5$, DW_2000Q_6 shows an advantage over Classic in the maze generation problem. The improvement using quantum annealing occurred because it determines the direction of $N \times N$ bars at once. Figure 14 shows the relation between maze generation time and maze size on Hybrid Solver. Linear and quadratic fits applied to the dataset indicate that the Hybrid Solver is $\mathcal{O}(1)$ or $\mathcal{O}(N)$ $\left[(3.29 \pm 0.83) \cdot 10^2 N + (2.99325 \pm 0.00090) \cdot 10^6\right]$ between $N = 1$ and $N = 18$ and then shifted to $\mathcal{O}(N^2)$ $\left[(6.899 \pm 0.065) \cdot 10^3 N^2 - (0.4 \pm 3.2) \cdot 10^3 N + (6.90 \pm 0.39) \cdot 10^5\right]$. The shift in the computational cost of Hybrid Solver may have resulted from a change in its algorithm.

3.3 Effect of update term

Here, 12 human subjects are asked to solve the maze one set (30 times), and the maze is shown to increase in difficulty as it adapts to each human subject. Figure 15A shows the increase rate from the first step of simple moving average of 10 solving time of maze generated without Q_{update} and individual increase rate. The solving time of the maze without Q_{update} was slightly getting shorter overall. Figure 15B shows the increase rate from the first step of simple moving average of 10 solving time of maze generated using Q_{update} and individual increase rate. The solving time of the maze using Q_{update} was getting longer overall. Most of the players increased their solving time, but some players decreased or did not change their solving time. In addition, nine players' average of the solving time of the maze generated using Q_{update}

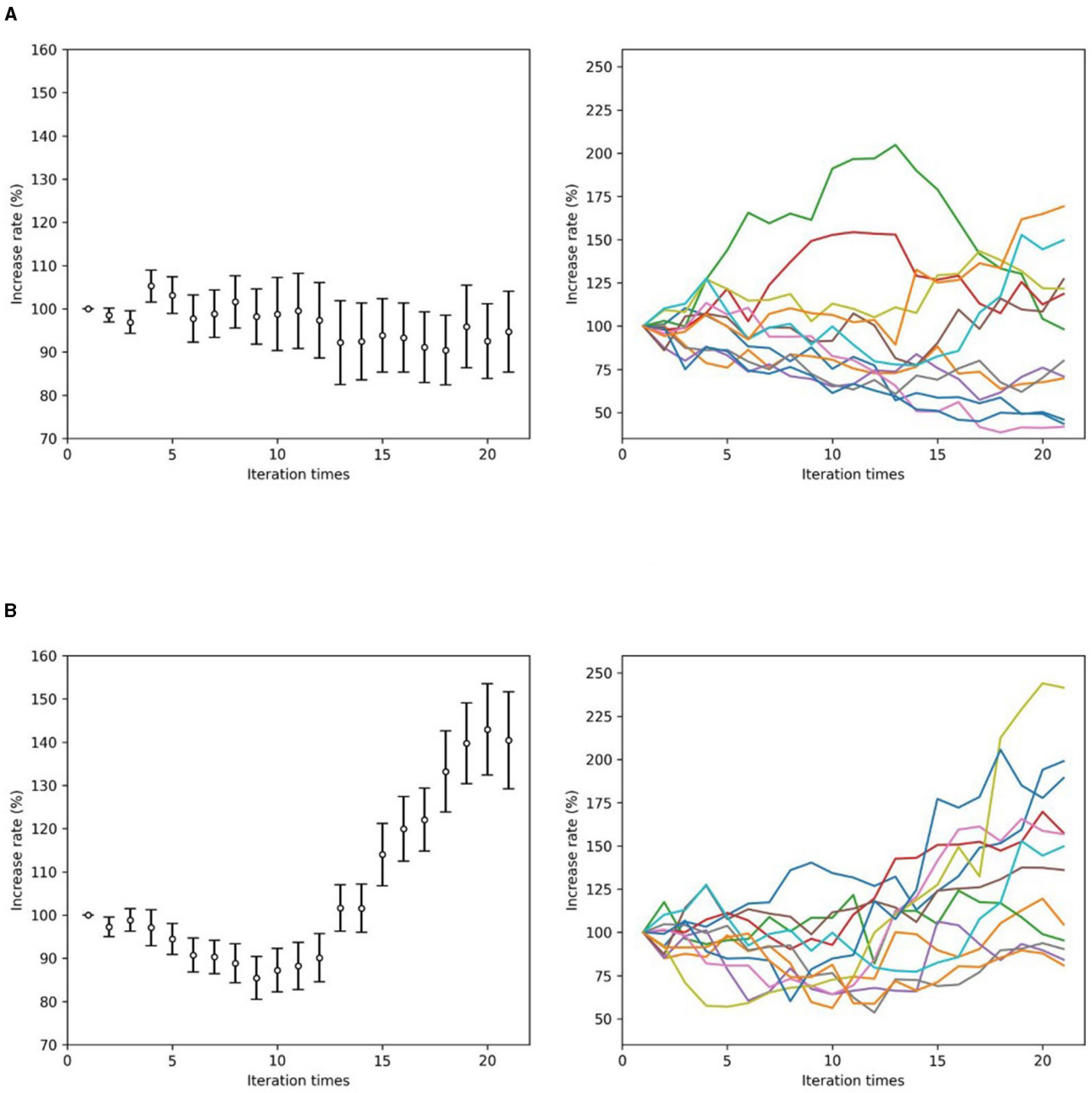


FIGURE 15
(A) Left: Increase rate from the first step of simple moving average of 10 solving time of 9 × 9 maze generated without Q_{update}. The error bars represent standard errors. Right: All players' increase rate from the first step of simple moving average of 10 solving time of 9 × 9 maze generated without Q_{update}. **(B)** Left: Increase rate from the first step of simple moving average of 10 solving time of 9 × 9 maze generated using Q_{update}. The error bars represent standard errors. Right: All players' increase rate from the first step of simple moving average of 10 solving time of 9 × 9 maze generated using Q_{update}.

increased than that of the maze generated without Q_{update}. These show that Q_{update} has the potential to increase the difficulty of the mazes.

4 Discussion

In this study, we show that generating difficult (longer the maze solving time) mazes using the bar-tipping algorithm is also possible with quantum annealing. By reformulating the bar-tipping algorithm as the combinatorial optimization problem, we generalize it more flexibly to generate mazes. In particular, our approach is simple but can adjust the difficulty in solving mazes by quantum annealing.

In Section 3.2, regarding comparing computational costs to solve our approach to generating mazes using TTS, DW_2000Q_6 has a smaller coefficient of N^2 than the classical counterpart. Therefore, as N increases, the computational cost of DW_2000Q_6

can be expected to be lower than that of the classical simulated annealing for a certain time. Unfortunately, since the number of qubits in the D-Wave quantum annealer is finite, the potential power of generating mazes by quantum annealing is limited. However, our insight demonstrates some advantages of quantum annealing against its classical counterpart. In addition, we observed that the hybrid solver's computational cost was constant up to $N = 18$. This indicates that hybrid solvers will be potentially effective if they are developed to deal with many variables in future.

In Section 3.3, we proposed Q_{update} to increase the solving time using quantum annealing. We demonstrated that introducing Q_{update} increased the time to solve the maze and changed the difficulty compared with the case where Q_{update} was not introduced. At this time, the parameters ($\lambda_{update1}$, $\lambda_{update2}$, and a) were fixed. Difficult maze generation for everyone may be possible by adjusting the parameters individually.

One of the directions in the future study is in applications of our cost function in various realms. We should emphasize that Q_{update} proposed in this study also has the potential use in various fields related to training and education. The powerful computation of quantum annealing and its variants open the way to such realms with high-speed computation and various solutions.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

Ethics statement

Ethical approval was not required for the studies involving humans because the data was obtained through collaboration among the participants of a group workshop developing an application using quantum annealing. The studies were conducted in accordance with the local legislation and institutional requirements. Written informed consent for participation was not required from the participants or the participants' legal guardians/next of kin in accordance with the national legislation and institutional requirements because the data was obtained through collaboration among the participants of a group workshop developing an application using quantum annealing.

Author contributions

YI: Writing – original draft, Writing – review & editing, Conceptualization, Data curation, Formal analysis, Investigation, Software, Validation, Visualization, Methodology. TY: Writing – original draft, Writing – review & editing, Conceptualization, Validation, Methodology. KO: Writing – original draft, Writing – review & editing, Conceptualization, Validation, Methodology. MO: Supervision, Writing – review & editing, Funding acquisition, Project administration.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. The authors thank financial support from the MEXT-Quantum Leap Flagship Program Grant No. JPMXS0120352009, as well as Public\Private R&D Investment Strategic Expansion PrograM (PRISM) and programs for Bridging the gap between R&D and the IDeal society (society 5.0) and Generating Economic and social value (BRIDGE) from Cabinet Office.

Acknowledgments

The authors thank the fruitful discussion with Reo Shikanai and Yoshihiko Nishikawa on applications of our approach to another application. This study is the result of research developed from an exercise class held at Tohoku University in Japan in the past called Quantum Annealing for You, 2nd party!. We want to thank one of the supporters, Rumiko Honda, for supporting the operations. The participants were a diverse group, ranging from high school students to university students, graduate students, technical college students, and working adults. As we can see from the authors' affiliations, this is a good example of a leap from the diversity of the participants to the creation of academic and advanced content.

Conflict of interest

MO is employed by Sigma-i.

The remaining authors declare that the research was conducted without any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Alg (2023a). Available online at:https://algoful.com/Archive/Algorithm/MazeBar (accessed August 13, 2023).

Alg (2023b). Available online at: https://algoful.com/Archive/Algorithm/MazeExtend (accessed August 13, 2023).

Alg (2023c). Available online at: https://algoful.com/Archive/Algorithm/MazeDig (accessed August 13, 2023).

Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B., and Melko, R. (2018). Quantum Boltzmann machine. *Phys. Rev.* X, 8. doi: 10.1103/PhysRevX.8.021050

Arai, S., Ohzeki, M., and Tanaka, K. (2021a). Mean field analysis of reverse annealing for code-division multiple-access multiuser detection. *Phys. Rev. Res.* 3, 033006. doi: 10.1103/PhysRevResearch.3.033006

Arai, S., Ohzeki, M., and Tanaka, K. (2021b). Teacher-student learning for a binary perceptron with quantum fluctuations. *J. Phys. Soc. Jpn.* 90, 074002. doi: 10.7566/JPSJ.90.074002

Bando, Y., and Nishimori, H. (2021). Simulated quantum annealing as a simulator of nonequilibrium quantum dynamics. *Phys. Rev. A* 104, 022607. doi: 10.1103/PhysRevA.104.022607

Bando, Y., Susa, Y., Oshiyama, H., Shibata, N., Ohzeki, M., Gómez-Ruiz, F. J., et al. (2020). Probing the universality of topological defect formation in a quantum annealer: Kibble-zurek mechanism and beyond. *Phys. Rev. Res.* 2, 033369. doi: 10.1103/PhysRevResearch.2.033369

Ding, Y., Chen, X., Lamata, L., Solano, E., and Sanz, M. (2021). Implementation of a hybrid classical-quantum annealing algorithm for logistic network design. *SN Comp. Sci.* 2, 1–9. doi: 10.1007/s42979-021-00466-2

Feld, S., Roch, C., Gabor, T., Seidel, C., Neukart, F., Galter, I., et al. (2019). A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer. *Front. ICT* 6, 13. doi: 10.3389/fict.2019.00013

Goto, T., and Ohzeki, M. (2023). Online calibration scheme for training restricted boltzmann machines with quantum annealing. *arXiv [Preprint]*. arXiv:2307.09785

Haba, R., Ohzeki, M., and Tanaka, K. (2022). Travel time optimization on multi-agv routing by reverse annealing. *Sci. Rep.* 12, 17753. doi: 10.1038/s41598-022-22704-0

Hasegawa, Y., Oshiyama, H., and Ohzeki, M. (2023). Kernel learning by quantum annealer. *arXiv [Preprint]*. arXiv:2304.10144

Hirama, S., and Ohzeki, M. (2023). Efficient algorithm for binary quadratic problem by column generation and quantum annealing. *arXiv*. doi: 10.7566/JPSJ.92.113002

Hussain, A., Bui, V.-H., and Kim, H.-M. (2020). Optimal sizing of battery energy storage system in a fast ev charging station considering power outages. *IEEE Transact. Transport. Electr.* 6, 453–463. doi: 10.1109/TTE.2020.2980744

Ide, N., Asayama, T., Ueno, H., and Ohzeki, M. (2020). "Maximum likelihood channel decoding with quantumannealingmachine," in *2020 International Symposium on Information Theory and Its Applications (ISITA)*, 91–95.

Inoue, D., Okada, A., Matsumori, T., Aihara, K., and Yoshida, H. (2021). Traffic signal optimization on a square lattice with quantum annealing. *Sci. Rep.* 11, 1–12. doi: 10.1038/s41598-021-82740-0

Kadowaki, T., and Nishimori, H. (1998). Quantum annealing in the transverse ising model. *Phys. Rev. E* 58, 5355–5363. doi: 10.1103/PhysRevE.58.5355

Khoshaman, A., Vinci, W., Denis, B., Andriyash, E., Sadeghi, H., and Amin, M. H. (2018). Quantum variational autoencoder. *Quant. Sci. Technol.* 4, 014001. doi: 10.1088/2058-9565/aada1f

King, A. D., Suzuki, S., Raymond, J., Zucca, A., Lanting, T., Altomare, F., et al. (2022). Coherent quantum annealing in a programmable 2,000 qubit ising chain. *Nat. Phys.* 18, 1324–1328. doi: 10.1038/s41567-022-01741-6

Koshikawa, A. S., Ohzeki, M., Kadowaki, T., and Tanaka, K. (2021). Benchmark test of black-box optimization using d-wave quantum annealer. *J. Phys. Soc. Jpn.* 90, 064001. doi: 10.7566/JPSJ.90.064001

Kumar, V., Bass, G., Tomlin, C., and Dulny, J. (2018). Quantum annealing for combinatorial clustering. *Quant. Inf. Process.* 17, 39. doi: 10.1007/s11128-017-1809-2

Maruyama, N., Ohzeki, M., and Tanaka, K. (2021). Graph minor embedding of degenerate systems in quantum annealing. *arXiv [Preprint]*. arXiv:2110.10930

mok Bae, C., Kim, E. K., Lee, J., joong Kim, K., and Na, J.-C. (2015). *Generation of an Arbitrary Shaped Large Maze by Assembling Mazes*. New York, NY: IEEE. doi: 10.1109/CIG.2015.7317901

Neukart, F., Compostella, G., Seidel, C., Von Dollen, D., Yarkoni, S., and Parney, B. (2017). Traffic flow optimization using a quantum annealer. *Front. ICT* 4, 29. doi: 10.3389/fict.2017.00029

Neven, H., Denchev, V. S., Rose, G., and Macready, W. G. (2012). "Qboost: large scale classifier training withadiabatic quantum optimization," in *Asian Conference on Machine Learning (PMLR)* (New York, NY), 333–348.

Nishimura, N., Tanahashi, K., Suganuma, K., Miyama, M. J., and Ohzeki, M. (2019). Item listing optimization for e-commerce websites based on diversity. *Front. Comput. Sci.* 1, 2. doi: 10.3389/fcomp.2019.00002

OḾalley, D., Vesselinov, V. V., Alexandrov, B. S., and Alexandrov, L. B. (2018). Nonnegative/binary matrix factorization with a d-wave quantum annealer. *PLoS ONE* 13, e0206653. doi: 10.1371/journal.pone.0206653

Ope (2023). Available online at: https://www.openjij.org/ (accessed October 10, 2023).

Orús, R., Mugel, S., and Lizaso, E. (2019). Forecasting financial crashes with quantum computing. *Phys. Rev. A* 99, 060301. doi: 10.1103/PhysRevA.99.060301

Oshiyama, H., and Ohzeki, M. (2022). Benchmark of quantum-inspired heuristic solvers for quadratic unconstrained binary optimization. *Sci. Rep.* 12, 2146. doi: 10.1038/s41598-022-06070-5

Pakin, S. (2017). "Navigating a maze using a quantum annealer," in *ITiCSE-WGR 2017 - Proceedings of the 2017 ITiCSE Conference onWorking Group Reports* (New York, NY). doi: 10.1145/3149526.3149532

Rosenberg, G., Haghnegahdar, P., Goddard, P., Carr, P., Wu, K., and De Prado, M. L. (2016). Solving the optimal trading trajectory problem using a quantum annealer. *IEEE J. Sel. Top. Signal Process.* 10, 1053–1060. doi: 10.1109/JSTSP.2016.2574703

Sato, T., Ohzeki, M., and Tanaka, K. (2021). Assessment of image generation by quantum annealer. *Sci. Rep.* 11, 13523. doi: 10.1038/s41598-021-92295-9

Shu Tanaka, M. Y., and Seki, Y. (2022). Black-box optimization by anneling machines. *J. Neural Circ. Soc. Jpn.* 29, 164–173. doi: 10.3902/jnns.29.164

Takabayashi, T., and Ohzeki, M. (2023). Hybrid algorithm of linear programming relaxation and quantum annealing. *arXiv [Preprint]*. arXiv:2308.10765

Tanaka, T., Sako, M., Chiba, M., Lee, C., Cha, H., and Ohzeki, M. (2023). Virtual screening of chemical space based on quantum annealing. *J. Phys. Soc. Jpn.* 92, 023001. doi: 10.7566/JPSJ.92.023001

Urushibata, M., Ohzeki, M., and Tanaka, K. (2022). Comparing the effects of boltzmann machines as associative memory in generative adversarial networks between classical and quantum samplings. *J. Phys. Soc. Jpn.* 91, 074008. doi: 10.7566/JPSJ.91.074008

Venturelli, D., and Kondratyev, A. (2019). Reverse quantum annealing approach to portfolio optimization problems. *Quant. Mach. Intell.* 1, 17–30. doi: 10.1007/s42484-019-00001-w

Venturelli, D., Marchand, D. J. J., and Rojo, G. (2016). Quantum annealing implementation of job-shop scheduling. *arXiv [Preprint]*. arXiv:1506.08479

Yamamoto, M., Ohzeki, M., and Tanaka, K. (2020). Fair sampling by simulated annealing on quantum annealer. *J. Phys. Soc. Jpn.* 89, 025002. doi: 10.7566/JPSJ.89.025002

Yonaga, K., Miyama, M., Ohzeki, M., Hirano, K., Kobayashi, H., and Kurokawa, T. (2022). Quantum optimization with lagrangian decomposition for multiple-process scheduling in steel manufacturing. *ISIJ Int.* 62, 1874–1880. doi: 10.2355/isijinternational.ISIJINT-2022-019

frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 12 December 2023
DOI 10.3389/fcomp.2023.1286226

OPEN ACCESS

EDITED BY
David Esteban Bernal Neira,
Purdue University, United States

REVIEWED BY
Ma Hongyang,
Qingdao University of Technology, China
Farshud Sorourifar,
The Ohio State University, United States

*CORRESPONDENCE
Mikiya Doi
✉ mikiya.doi.t2@dc.tohoku.ac.jp

RECEIVED 31 August 2023
ACCEPTED 06 November 2023
PUBLISHED 12 December 2023

CITATION
Doi M, Nakao Y, Tanaka T, Sako M and Ohzeki M (2023) Exploration of new chemical materials using black-box optimization with the D-wave quantum annealer.
Front. Comput. Sci. 5:1286226.
doi: 10.3389/fcomp.2023.1286226

Exploration of new chemical materials using black-box optimization with the D-wave quantum annealer

Mikiya Doi[1]*, Yoshihiro Nakao[2], Takuro Tanaka[2], Masami Sako[2] and Masayuki Ohzeki[1,3,4,5]

[1]Graduate School of Information Sciences, Tohoku University, Sendai, Japan, [2]LG Japan Lab Ltd., Yokohama, Japan, [3]International Research Frontier Initiative, Tokyo Institute of Technology, Tokyo, Japan, [4]Department of Physics, Tokyo Institute of Technology, Tokyo, Japan, [5]Sigma-i Co. Ltd., Tokyo, Japan

In materials informatics, searching for chemical materials with desired properties is challenging due to the vastness of the chemical space. Moreover, the high cost of evaluating properties necessitates a search with a few clues. In practice, there is also a demand for proposing compositions that are easily synthesizable. In the real world, such as in the exploration of chemical materials, it is common to encounter problems targeting black-box objective functions where formalizing the objective function in explicit form is challenging, and the evaluation cost is high. In recent research, a Bayesian optimization method has been proposed to formulate the quadratic unconstrained binary optimization (QUBO) problem as a surrogate model for black-box objective functions with discrete variables. Regarding this method, studies have been conducted using the D-Wave quantum annealer to optimize the acquisition function, which is based on the surrogate model and determines the next exploration point for the black-box objective function. In this paper, we address optimizing a black-box objective function containing discrete variables in the context of actual chemical material exploration. In this optimization problem, we demonstrate results obtaining parameters of the acquisition function by sampling from a probability distribution with variance can explore the solution space more extensively than in the case of no variance. As a result, we found combinations of substituents in compositions with the desired properties, which could only be discovered when we set an appropriate variance.

KEYWORDS

quantum annealing, quantum computing, black-box optimization, combinatorial optimization problem, materials informatics

1 Introduction

Black-box optimization is a method to optimize a function that does not have an explicit objective function in the mathematical form. In the real world, this optimization problem appears in various fields, including material informatics, robotics (Deisenroth, 2011), machine learning (Snoek et al., 2012), and recommendation systems (Vanchinathan et al., 2014). Bayesian optimization is one of the solutions for black-box optimization problems (Jones et al., 1998). Taking the exploration of chemical materials as an example, a surrogate model is constructed using an existing dataset to predict the relationship between the combinations of substituents in the chemical materials and the corresponding property

values. Based on this surrogate model, an acquisition function is defined. The combination of substituents obtained through optimizing this acquisition function is then used as the next input point for the black-box objective function, enabling the evaluation of the actual property values. The relationship between the inputted combination of substituents and the actual property value is then added to the existing dataset, then updating the surrogate model. Repeating this process is to explore the combinations of substituents that yield the desired property values. Especially for black-box optimization problems involving discrete variables, discrete variables are included in both the surrogate model and the acquisition function. Therefore, even optimizing the acquisition function often proves to be NP-hard, and the solutions obtained through optimization are generally approximate. In a previous study, Bayesian optimization of combinatorial structures (BOCS) was proposed as the promising algorithm for such problems (Baptista and Poloczek, 2018). In this algorithm, the acquisition function was assumed as quadratic unconstrained binary optimization (QUBO) problem.

Quantum annealing (Kadowaki and Nishimori, 1998) is a heuristic algorithm to solve QUBO problems by driving binary variables through quantum fluctuations. Many well-known combinatorial optimization problems can be encoded into QUBO problems (Lucas, 2014). Practical applications of quantum annealing can be found in various fields, including traffic flow optimization (Neukart et al., 2017; Inoue et al., 2021; Shikanai et al., 2023), manufacturing (Ohzeki et al., 2019; Haba et al., 2022), finance (Rosenberg et al., 2016; Venturelli and Kondratyev, 2019), steel manufacturing (Yonaga et al., 2022), decoding problems (Ide et al., 2020; Arai et al., 2021), and algorithms in machine learning (Amin et al., 2018; O'Malley et al., 2018; Urushibata et al., 2022; Goto and Ohzeki, 2023; Hasegawa et al., 2023). Furthermore, quantum annealing, which utilizes the quantum tunneling effect, is expected to find the optimal solution for several combinatorial optimization problems more rapidly than algorithms such as simulated annealing (Kirkpatrick et al., 1983). This advantage is investigated from the perspective of energy landscape characteristics (Das and Chakrabarti, 2008) and through numerical computation (Denchev et al., 2016). In addition, there are discussions about the characteristics of solutions obtained in cases where multiple optimal solutions exist (Yamamoto et al., 2020; Maruyama et al., 2021). With these backgrounds, quantum annealing has recently attracted attention, both for its potential applications and for validating the fundamental aspects of quantum effects.

Studies that employ quantum annealing in some algorithms for black-box optimization problems involving discrete variables exist. These include benchmark tests (Koshikawa et al., 2021) that have examined the presence or absence of quantum superiority in optimizing acquisition functions. In terms of practical applications, there are case studies that have achieved significant screening in the exploration of chemical materials within the search chemical space (Hatakeyama-Sato et al., 2021; Tanaka et al., 2023), as well as instances of designing complex metamaterials (Kitai et al., 2020).

In the exploration of chemical materials, it is necessary not only to discover molecules with the desired property values but also to be concerned about scenarios in actual synthesis where molecules with specific substructures may become entirely unfeasible to synthesize. Drawing inspiration from previous studies and practical needs, we demonstrate a method for proposing diverse compositions of chemical materials with desired properties, targeting a black-box optimization problem that includes discrete variables in actual chemical material exploration. In more detail, we show results that by obtaining parameters of the surrogate model and acquisition function from sampling a probability distribution with an appropriate variance and optimizing the acquisition function, we explored the solution space more extensively while optimizing the black-box objective function. The method used in this paper is generally referred to as Thompson sampling (Thompson, 1933; Chapelle and Li, 2011). In this sense, it can be said that our research results evaluate the impact of the magnitude of the variance of the posterior probability distribution in Thompson sampling.

The remaining sections of this paper are organized as follows: In the next section, Section 2, we explain the problem setting in this paper and the method we propose. In Section 3, we demonstrate the results of the experiments related to the actual exploration of chemical materials. Finally, Section 4 summarizes our research and discusses this paper and future research directions.

2 Materials and methods

In this section, we introduce the problem settings based on the search for chemical materials, which is the focus of this paper. Subsequently, in Bayesian optimization, we explain the construction of the surrogate model in the QUBO form, which is well-known in prior research, along with the construction of the acquisition function. We provide this explanation in conjunction with our method aim.

2.1 Problem settings

In this paper, we define the binding of substituents to specific sites of the molecular frame as the composition of chemical materials. We aim to propose various combinations of substituents through Bayesian optimization while maximizing a target material property value. To align our description with other literature focusing on black-box optimization problems, we define our goal as a minimization problem, utilizing the fact that maximization and minimization problems can be transformed into each other by reversing the sign of the objective function.

2.2 Methods

We express the assignment of substituents using a binary vector. In particular, for substituents that can bind to each site, we encode them by converting the 0-indexed substituent number to binary. Thus, we set a binary vector $\vec{x}^{(\mu)} \in \{0,1\}^N$ as input, and the corresponding target material property value $y^{(\mu)}$ as output. We aim to find $\vec{x}$ that minimizes a black-box objective function. Since we cannot know an explicit form of the black-box objective function, we construct a surrogate model as

QUBO form following the previous studies. We utilize an existing dataset $\mathcal{D} = \{\vec{x}^{(\mu)}, y^{(\mu)}\}_{\mu=1}^{D}$ to sample the parameters of the surrogate model from a probability distribution we discuss later and construct it. Based on the surrogate model, we construct an acquisition function and propose a combination of substituents that optimize the acquisition function using the D-Wave quantum annealer. Subsequently, we input the proposed combination of substituents as the next exploration point $\vec{x}^{(new)}$ and obtain output $y^{(new)}$ from the black-box objective function. Then we append $\{\vec{x}^{(new)}, y^{(new)}\}$ to the existing dataset as new data and reconstruct the surrogate model. By repeating this process, we aim to obtain diverse combinations of substituents with desired target material property values.

2.2.1 Construction of surrogate model function

We construct the surrogate model $f_{surrogate}(\vec{x})$ in the QUBO form in this paper.

$$f_{surrogate}(\vec{x}) = \alpha_0 + \sum_i \alpha_i x_i + \sum_{i<j} \alpha_{ij} x_i x_j \tag{1}$$

For simplicity, we set the surrogate model parameters $\{\alpha_i, \alpha_{ij}\} = \vec{\alpha} \in \mathbb{R}^p$. Note that $p = 1 + N + N(N-1)/2$. Defining $X \in \{0,1\}^{D\times p}$ as the design matrix and denoting the μ-th row in the design matrix X as $X^{(\mu)}$, we have the following expression $X^{(\mu)} = \left(1, x_1^{(\mu)}, ..., x_N^{(\mu)}, x_1^{(\mu)}x_2^{(\mu)}, x_1^{(\mu)}x_3^{(\mu)}, ..., x_{N-1}^{(\mu)}x_N^{(\mu)}\right)$. Furthermore, we set the output vector $\vec{y} \in \mathbb{R}^D$ and I as the identity matrix. Then, we assume a prior distribution of surrogate model parameters $P(\vec{\alpha})$ with a variance $\sigma_\alpha^2 I$ and a likelihood function over the surrogate model parameters $\vec{\alpha}$ with a variance $\sigma_y^2 I$. We give the prior distribution and likelihood function as following multivariate Gaussian distributions.

$$P(\vec{\alpha}) = \mathcal{N}(\vec{0}, \sigma_\alpha^2 I) \tag{2}$$

$$P(\vec{y}|\vec{\alpha}, X) = \mathcal{N}(X\vec{\alpha}, \sigma_y^2 I) \tag{3}$$

At this time, the posterior distribution of the surrogate model parameters $\vec{\alpha}$ is computed and given by a multivariate Gaussian distribution, similar to the prior distribution and the likelihood function.

$$\begin{aligned} \vec{\alpha}|\vec{y}, X &\sim \mathcal{N}(\vec{\mu}, \Sigma) \\ \vec{\mu} &= (X^T X + \lambda I)^{-1} X^T \vec{y} \\ \Sigma &= \sigma^2 (X^T X + \lambda I)^{-1} \\ s.t.\ \sigma^2 &= \sigma_y^2\ ,\ \lambda = \frac{\sigma_y^2}{\sigma_\alpha^2} \end{aligned} \tag{4}$$

We sample the surrogate model parameters $\vec{\alpha} \in \mathbb{R}^p$ from the multivariate Gaussian distribution described in (4). σ^2 is a hyperparameter indicating the magnitude of fluctuations from the mean vector $\vec{\mu}$ when sampling the surrogate model parameters. λ is also a hyperparameter. Note that λ corresponds to the coefficient of the regularization term during ridge regression.

2.2.2 Construction of acquisition function

The acquisition function $f_{acquisition}(\vec{x})$ is constructed in the same QUBO form as the surrogate model, and the next exploration point $\vec{x}^{(new)}$ is proposed by optimizing the acquisition function.

$$\vec{x}^{(new)} = \arg\min_{\vec{x}} \{f_{acquisition}(\vec{x})\} \tag{5}$$

$f_{acquisition}(\vec{x})$ is a function with modified specific parameters from the surrogate model $f_{surrogate}(\vec{x})$ described in 2.2.1. This modification is like a penalty method, designed to ensure that binary vectors with substituent numbers that do not exist at each site do not become the optimal points of the acquisition function. Parameters that are not modified are identical to those in the surrogate model $f_{surrogate}(\vec{x})$. For example, when six potential substituents can bind at a specific site, representing the 0-indexed substituent numbers in binary requires three bits (x_1, x_2, x_3). In this context, $x_1x_2x_3 = (000, 001, 010, 011, 100, 101)_2$ corresponds to valid substituent numbers from 0 to 5 in decimal. However, each combination $x_1x_2x_3 = (110, 111)_2$ is equivalent to substituent numbers 6–7 in decimal, rendering them inappropriate as optimal point candidates. To prevent the substituent combinations with substituent numbers 6–7 at this site from being proposed as the optimal points of the acquisition function, we adjust the surrogate model parameters. In this example, we modify the coefficient of x_1x_2 in the surrogate model function to a positive constant C, and the other coefficients are kept the same as in the surrogate model. The next exploration point of the black-box objective function is determined by the optimization of the acquisition function $f_{acquisition}(\vec{x})$.

The search space explored varies greatly depending on how the acquisition function is constructed and how the acquisition function is optimized. As described, our method samples the parameters of the surrogate model and the acquisition function from a probability distribution with variance. The hyperparameter σ^2 indicates the magnitude of the variance. The larger this hyperparameter σ^2 is, the more significant the variance of the acquisition function, potentially allowing for exploration across a broader solution space and avoiding resampling the previously explored points.

3 Results

In this section, we describe detailed problem settings and experimental conditions and then show the experimental results obtained by applying our method. In particular, we compare and discuss based on the magnitude of the hyperparameter σ^2. Our discussion centers on two main points of interest in this paper. The first point is whether our method has brought diversity to the proposed substituent combinations. The second point is whether our method has optimized the black-box objective function.

3.1 Detailed problem settings and experimental conditions

We set the number of substituent binding sites as four, and for convenience in the description, we call each binding site R1,

R2, R3, and R4, respectively. The number of possible substituents that can bind at each site is R1: 6, R2: 29, R3: 64, and R4: 64, respectively. Therefore, the size of the chemical space is calculated as $6 \times 29 \times 64 \times 64 = 712704$. Moreover, the number of bits necessary to represent the number of each substituent is R1: 3, R2: 5, R3: 6, and R4: 6. Consequently, the binary vector $\vec{x}$ dimension is calculated as $N = 3 + 5 + 6 + 6 = 20$. The substituent number at R1 is represented in 0-indexed form using x_1 to x_3, similarly, x_4 to x_8 represent the substituent number at R2, x_9 to x_{14} represent the substituent number at R3, and x_{15} to x_{20} represent the substituent number at R4. To illustrate with a concrete encoding example, suppose the substituent numbers at each site are R1: 0, R2: 2, R3: 10, and R4: 63. In this case, the binary vector $\vec{x}$ would be represented as $\vec{x} = (0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1)$. We set the hyperparameter λ at 10^{-2} and the hyperparameter σ^2, which indicates the magnitude of fluctuation from the mean vector $\vec{\mu}$ when sampling surrogate model parameters, to $\{0, 4 \times 10^{-3}, 8 \times 10^{-3}, 12 \times 10^{-3}\}$. We set the surrogate model's parameter correction for R1 in the acquisition function as $C = \alpha_{12} = 2 \times max(\vec{\alpha})$ at each after sampling $\vec{\alpha}$. We used D-Wave Advantage 4.1 as the quantum annealer, setting the annealing time to 2,000 μs, and the number of samples is 300. The quantum adiabatic theorem ensures that it is possible to find the nontrivial ground state at the end of the quantum annealing if the transverse field changes sufficiently slowly (Suzuki and Okada, 2005; Morita and Nishimori, 2008; Ohzeki and Nishimori, 2011). On the other hand, when quantum annealing is carried out on a physical device D-wave quantum annealer, it operates at a finite temperature and is subject to external noise. Due to these factors, the annealing time is often short in many studies. Considering these theoretical and experimental backgrounds, we set the annealing time to be longer in our setting because we observed a tendency for the results to stabilize, possibly due to the effects of ambient temperature. The number of samples in the initial dataset is 992. For comparison as a baseline, we also conducted an experiment where the optimization part of the acquisition function was replaced with random sampling. Due to the nature of this study, which is conducted in the context of actual chemical material exploration, the computational cost of the black-box objective function is exceptionally high, resulting in an experiment of only one instance. We defined one loop as carrying out the following steps (i) through (v), and we performed 20 loops.

(i) By sampling the surrogate model parameters $\vec{\alpha}$ from a multivariate Gaussian distribution $\mathcal{N}(\vec{\mu}, \Sigma)$ described in (4), construct the surrogate model.
(ii) Construct the acquisition function by partially correcting the surrogate model parameters as explained in 2.2.2
(iii) Optimize the acquisition function by quantum annealing and select the top 10 points of the acquisition function as the next exploration points for the black-box objective function. In the random sampling used as a baseline, 10 sampling points are randomly selected. Note that at this time, the top 10 points exclude combinations of substituents that are already present in the existing dataset and combinations of substituents that include non-existent substituent numbers, such as substituent numbers 6-7 in R1 and substituent numbers 29–31 in R2, through screening.
(iv) Take the next exploration points obtained in (iii) as inputs and get outputs, carrying out the evaluation of target material property values, which is the computation of the black-box objective function, through DFT (Density Functional Theory) calculations. The detailed calculation method is described in the Additional Requirements.
(v) Append the new samples $\{\vec{x}^{(new)}, y^{(new)}\}$ obtained in (iv) to the existing dataset and return to (i).

3.2 Experimental results

3.2.1 Histogram of substituent numbers in combinations added by end of the experiment

We show the histogram of substituent numbers at the binding sites R1, R2, R3, and R4 for the combinations of substituents added to the dataset by the end of the experiment in Figure 1. In the case of $\sigma^2 = 0$, we observed a tendency in R3 and R4 where specific substituent numbers were frequently proposed. However, as σ^2 increases, it can be observed that diversity is brought into the combinations of substituents proposed for R3 and R4. This difference is particularly pronounced when comparing $\sigma^2 = 0$ and $\sigma^2 = 12 \times 10^{-3}$. From these results, we can infer that we realized the proposal of various combinations of substituents by sampling parameters of the surrogate model and the acquisition function from probability distributions with variance. By sampling parameters from probability distributions with larger variances, the optimal points and the shape of the acquisition function change significantly in each loop. We believe that this approach allowed us to explore the solution space without getting trapped by some specific approximate solutions and without resampling the previously explored points.

3.2.2 Relationship between the number of loops and the R^2 of the surrogate model

We show the transition of the coefficient of determination R^2 in the surrogate model at each loop in Figure 2. The coefficient of determination R^2 is calculated from the initial dataset sample points, 992 points, and the sample points appended up to each loop. Note that R^2, plotted in Figure 2, represents the results of mean-based regression. This result is equivalent to the regression of the maximum a posteriori (MAP) estimation. As σ^2 becomes larger, a tendency for R^2 at each loop to become smaller was observed. We speculate that we can attribute this result to the tendency shown in Figure 1, where the larger σ^2 is, the more diverse the combinations of substituents that the optimization of the acquisition function proposes become. When σ^2 is small, R^2 improves by fitting to similar input vectors and outputs. However, to improve R^2 when σ^2 is large, it is necessary to fit diverse input vectors and outputs. We speculate that this difficulty is why there was the tendency for the coefficient of determination, R, to be smaller when σ^2 is larger.

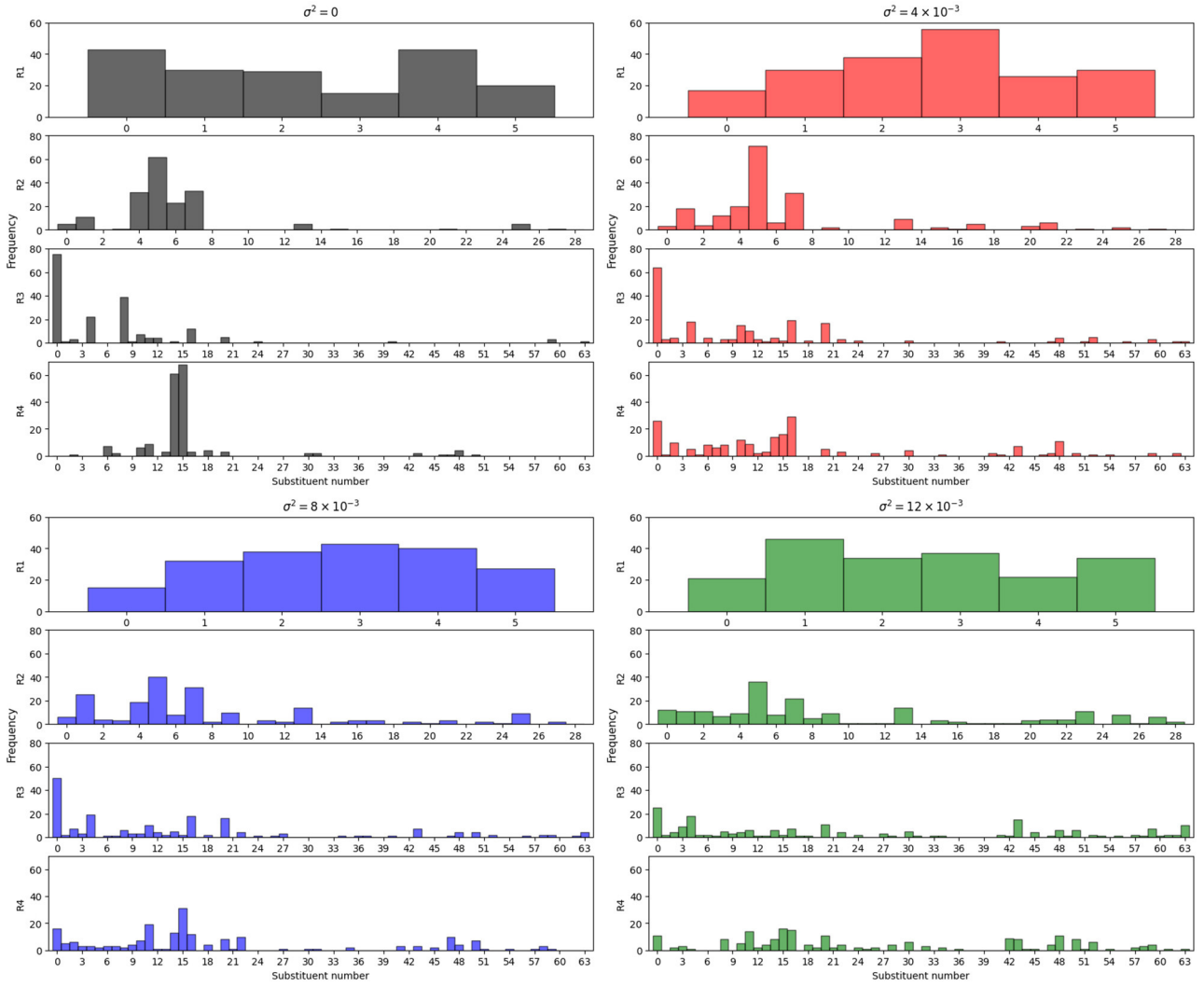


FIGURE 1
Histogram of substituent numbers for combinations of substituents added to the dataset by the end of the experiment each σ^2. Top left is $\sigma^2 = 0$, **top right** is $\sigma^2 = 4 \times 10^{-3}$, **bottom left** is $\sigma^2 = 8 \times 10^{-3}$, bottom right is $\sigma^2 = 12 \times 10^{-3}$.

3.2.3 Analysis of target material property values

Finally, we show the target material property values evaluated by DFT calculations, corresponding to the combinations of substituents proposed through the optimization of the acquisition function as the next exploration point of the black-box objective function in Figures 3, 4. In Figure 3, we plot the transition of the best target material property values in the existing dataset up to each loop. Although we could only experiment once because of the extremely high computational cost of the black-box objective function, in the case of optimizing the acquisition functions, we confirm that it is possible to search for combinations of substituents with higher target material property values than the best value in the initial dataset. Under the conditions set in this study, using random sampling in the optimization part of the acquisition function, we could not find any combination of substituents that exhibited a property value exceeding the best target material property value in the initial dataset. In Figure 4, we show the histograms of the target material property values for all combinations of substituents in the initial dataset and those added to the dataset by the end of each experiment.

To reiterate, the objective of black-box optimization in this study was to maximize the target material property value while bringing diversity to the combinations of substituents. Therefore, we listed the combinations of substituents whose target material property values exceeded our criteria of 0.880 or higher in Tables 1–4. From the perspective of the number of combinations of substituents with property values that exceed our criteria, the number of proposed combinations was the highest at 25 combinations when $\sigma^2 = 0$. However, considering the diversity of proposed combinations of substituents, which is one of the aims of this paper, the advantage can be found when $\sigma^2 \neq 0$. Especially in the case of $\sigma^2 = 4 \times 10^{-3}$, it was possible to discover combinations of substituents with the property values that exceed our criteria, which have the substituent number of R4:0, a combination not discovered in case of $\sigma^2 = 0$.

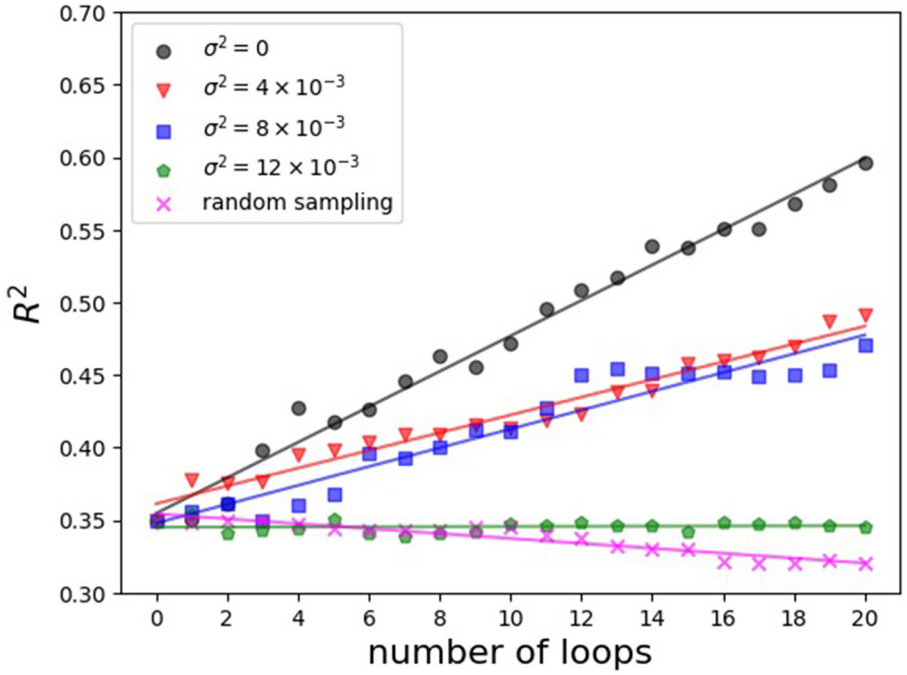


FIGURE 2
Relationship between the number of loops and the coefficient of determination R^2 in the surrogate model each σ^2 and random sampling.

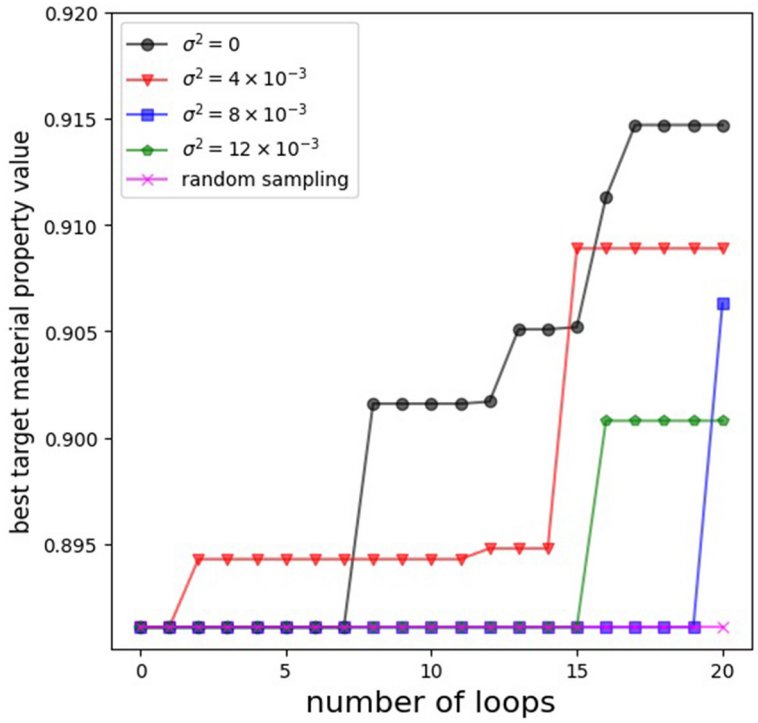


FIGURE 3
The transition of the best target material property values in the existing dataset up to each loop.

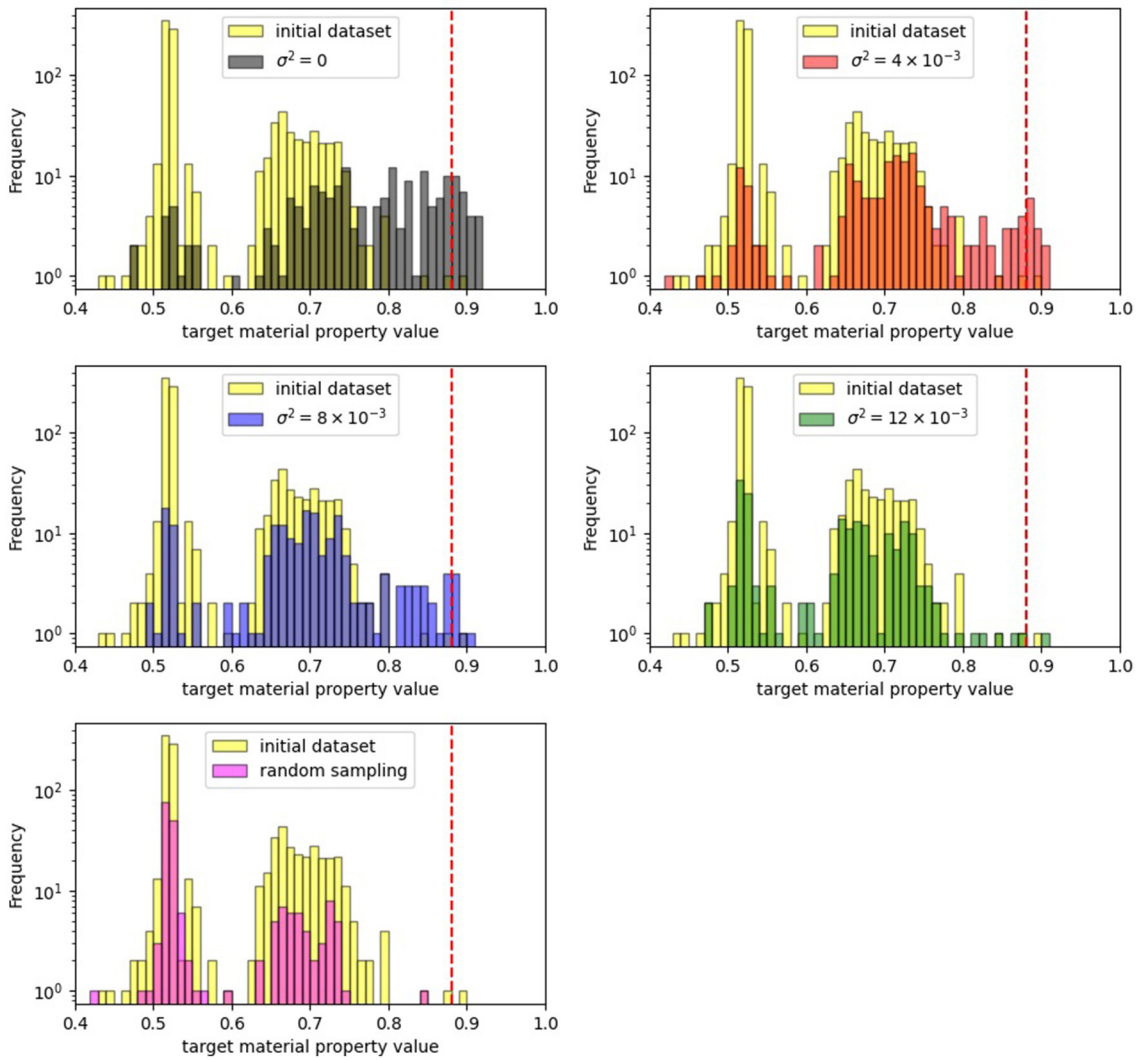


FIGURE 4
The histogram of the target material property values for the combinations of substituents in the initial dataset and those added to the dataset by the end of each experiment. The red dotted line shows the cutoff value (0.880), which we defined as a desired target material property value.

4 Discussion

In this study, we achieved the exploration of diverse approximate solutions in black-box optimization, which has the background of new chemical material discovery, by considering appropriate fluctuations in the parameters of the surrogate model and the acquisition function. Although the validity of the result is debatable because of the one-instance experiment, our result indicates that quantum annealing can accelerate the discovery of diverse chemical materials with desired material property values in materials informatics. More generally, our results demonstrate the advantages and disadvantages of varying the magnitude of the variance when sampling the parameters of the surrogate model from a probability distribution in optimizing a black-box objective function. In this paper, we explored a broader solution space by devising the construction of the surrogate model and the acquisition function. As an alternative approach, we are considering optimizing the acquisition function using a different method from quantum annealing, such as simulated annealing. Our method in this paper, which encodes combinations of substituents as a binary vector, can be applied even in a more vast chemical space. Future challenges include verifying the performance in such cases and investigating the computational time advantage of using quantum annealing.

5 Additional requirements

5.1 DFT (Density Functional Theory) calculations

For the proposed substituents by the D-Wave quantum annealer, the energy value of ground and excited states were calculated by optimizing the geometry based on DFT calculation. DFT calculations were performed using the supercomputer TSUBAME 3.0 with Gaussian16, Revision C.01 software (Frisch

TABLE 1 Desired target material property value in $\sigma^2 = 0$.

R1	R2	R3	R4	Target material property value
2	5	0	15	0.881
4	7	8	15	0.889
4	5	8	15	0.889
4	5	0	14	0.902
0	5	0	14	0.897
4	6	8	15	0.894
0	5	8	15	0.887
2	5	8	15	0.902
1	5	0	14	0.905
0	7	8	14	0.882
0	6	0	14	0.898
0	7	0	14	0.905
0	5	14	14	0.911
2	5	0	14	0.88
4	6	0	14	0.915
2	7	0	14	0.888
2	7	8	15	0.892
1	6	0	14	0.91
2	6	0	14	0.89
3	6	0	14	0.884
3	5	8	15	0.894
1	7	8	14	0.887
1	6	8	14	0.891
3	7	8	15	0.891
1	7	0	14	0.914

TABLE 2 Desired target material property value in $\sigma^2 = 4 \times 10^{-3}$.

R1	R2	R3	R4	Target material property value
0	5	0	15	0.894
3	5	0	14	0.882
3	5	0	15	0.88
2	5	0	15	0.881
4	7	0	0	0.895
3	5	0	0	0.882
1	5	0	14	0.909
0	5	0	14	0.901
3	6	8	15	0.899
3	7	0	0	0.881
2	7	0	0	0.883

et al., 2019), with the functional B3LYP and basis functions 6-31G.19 parameters from the DFT calculation were used to reproduce the experimental values.Here, a prediction model was created using random forest regression.

TABLE 3 Desired target material property value in $\sigma^2 = 8 \times 10^{-3}$.

R1	R2	R3	R4	Target material property value
2	5	0	15	0.881
2	5	0	14	0.88
4	5	0	15	0.88
4	5	8	15	0.886
0	5	0	14	0.897
1	5	0	14	0.906

TABLE 4 Desired target material property value in $\sigma^2 = 12 \times 10^{-3}$.

R1	R2	R3	R4	Target material property value
2	7	14	14	0.901

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

MD: Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing—original draft, Writing—review & editing. YN: Data curation, Funding acquisition, Investigation, Project administration, Resources, Software, Visualization, Writing—review & editing, Methodology. TT: Data curation, Funding acquisition, Investigation, Project administration, Resources, Software, Visualization, Writing—review & editing, Methodology. MS: Investigation, Methodology, Resources, Software, Writing—review & editing. MO: Conceptualization, Funding acquisition, Methodology, Project administration, Resources, Supervision, Writing—review & editing.

Funding

Our study receives financial support from the MEXT-Quantum Leap Flagship Program Grant No. JPMXS0120352009, as well as Public\Private R&D Investment Strategic Expansion PrograM (PRISM) and programs for Bridging the gap between R&D and the IDeal society (society 5.0) and Generating Economic and social value (BRIDGE) from Cabinet Office.

Acknowledgments

This study used the TSUBAME3.0 supercomputer at Tokyo Institute of Technology. The authors thank the fruitful discussions with Y. Nishikawa.

Conflict of interest

YN, TT, and MS are employed by LG Japan Lab Ltd. MO is employed by Sigma-i.
The remaining author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B., and Melko, R. (2018). Quantum Boltzmann machine. *Phys. Rev. X* 8, 021050. doi: 10.1103/PhysRevX.8.021050

Arai, S., Ohzeki, M., and Tanaka, K. (2021). Mean field analysis of reverse annealing for code-division multiple-access multiuser detection. *Phys. Rev. Res.* 3, 033006. doi: 10.1103/PhysRevResearch.3.033006

Baptista, R., and Poloczek, M. (2018). "Bayesian optimization of combinatorial structures," in *Proceedings of the 35th International Conference on Machine Learning* (PMLR) (Stockholm: ICML), 462–471.

Chapelle, O., and Li, L. (2011). "An empirical evaluation of Thompson sampling," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, eds J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger (Red Hook, NY: Curran Associates, Inc.), 2249–2257.

Das, A., and Chakrabarti, B. K. (2008). Colloquium: Quantum annealing and analog quantum computation. *Rev. Modern Phys.* 80, 1061–1081. doi: 10.1103/RevModPhys.80.1061

Deisenroth, M. P. (2011). A survey on policy search for robotics. *Found. Trends Robot.* 2, 1–142. doi: 10.1561/2300000021

Denchev, V. S., Boixo, S., Isakov, S. V., Ding, N., Babbush, R., Smelyanskiy, V., et al. (2016). What is the computational value of finite-range tunneling? *Phys. Rev. X* 6, 031015. doi: 10.1103/PhysRevX.6.031015

Frisch, M. J., Trucks, G. W., Schlegel, H. B., Scuseria, G. E., Robb, M. A., Cheeseman, J. R., et al. (2019). *Gaussian 16 Revision C.01*. Wallingford, CT: Gaussian Inc.

Goto, T., and Ohzeki, M. (2023). Online calibration scheme for training restricted Boltzmann machines with quantum annealing. *arXiv preprint arXiv:2307.09785*. doi: 10.48550/ARXIV.2307.09785

Haba, R., Ohzeki, M., and Tanaka, K. (2022). Travel time optimization on multi-AGV routing by reverse annealing. *Sci. Rep.* 12, 17753. doi: 10.1038/s41598-022-22704-0

Hasegawa, Y., Oshiyama, H., and Ohzeki, M. (2023). Kernel Learning by quantum annealer. *arXiv preprint arXiv:2304.10144*. doi: 10.48550/ARXIV.2304.10144

Hatakeyama-Sato, K., Kashikawa, T., Kimura, K., and Oyaizu, K. (2021). Tackling the challenge of a huge materials science search space with quantum-inspired annealing. *Adv. Intell. Syst.* 3, 2000209. doi: 10.1002/aisy.202000209

Ide, N., Asayama, T., Ueno, H., and Ohzeki, M. (2020). "Maximum likelihood channel decoding with quantum annealing machine," in *2020 International Symposium on Information Theory and Its Applications (ISITA)* (IEEE).

Inoue, D., Okada, A., Matsumori, T., Aihara, K., and Yoshida, H. (2021). Traffic signal optimization on a square lattice with quantum annealing. *Sci. Rep.* 11, 1–12. doi: 10.1038/s41598-021-82740-0

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *J. Glob. Opt.* 13, 455–492. doi: 10.1023/A:1008306431147

Kadowaki, T., and Nishimori, H. (1998). Quantum annealing in the transverse Ising model. *Phys. Rev E* 58, 5355–5363.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science* 220, 671–680.

Kitai, K., Guo, J., Ju, S., Tanaka, S., Tsuda, K., Shiomi, J., et al. (2020). Designing metamaterials with quantum annealing and factorization machines. *Phys. Rev. Res.* 2, 013319. doi: 10.1103/PhysRevResearch.2.013319

Koshikawa, A. S., Ohzeki, M., Kadowaki, T., and Tanaka, K. (2021). Benchmark test of black-box optimization using D-wave quantum annealer. *J. Phys. Soc. Japan* 90, 064001. doi: 10.7566/JPSJ.90.064001

Lucas, A. (2014). Ising formulations of many NP problems. *Front. Phys.* 2, 5. doi: 10.3389/fphy.2014.00005

Maruyama, N., Ohzeki, M., and Tanaka, K. (2021). Graph minor embedding of degenerate systems in quantum annealing. *arXiv preprint arXiv:2110.10930*.

Morita, S., and Nishimori, H. (2008). Mathematical foundation of quantum annealing. *J. Math. Phys.* 49, 125210. doi: 10.1063/1.2995837

Neukart, F., Compostella, G., Seidel, C., Von Dollen, D., Yarkoni, S., and Parney, B. (2017). Traffic flow optimization using a quantum annealer. *Front. ICT* 4, 29. doi: 10.3389/fict.2017.00029

Ohzeki, M., Miki, A., Miyama, M. J., and Terabe, M. (2019). Control of automated guided vehicles without collision by quantum annealer and digital devices. *Front. Comput. Sci.* 1, 9. doi: 10.3389/fcomp.2019.00009

Ohzeki, M., and Nishimori, H. (2011). Quantum annealing: an introduction and new developments. *J. Comput. Theoret. Nanosci.* 8, 963–971. doi: 10.1166/jctn.2011.1776963

O'Malley, D., Vesselinov, V. V., Alexandrov, B. S., and Alexandrov, L. B. (2018). Nonnegative/binary matrix factorization with a D-wave quantum annealer. *PLoS ONE* 13, e0206653. doi: 10.1371/journal.pone.0206653

Rosenberg, G., Haghnegahdar, P., Goddard, P., Carr, P., Wu, K., and de Prado, M. L. (2016). Solving the optimal trading trajectory problem using a quantum annealer. *IEEE J. Select. Top. Signal Processs.* 10, 1053–1060. doi: 10.1109/JSTSP.2016.2574703

Shikanai, R., Ohzeki, M., and Tanaka, K. (2023). Traffic signal optimization using quantum annealing on real map. *arXiv preprint arXiv:2308.14462*. doi: 10.48550/ARXIV.2308.14462

Snoek, J., Larochelle, H., and Adams, R. P. (2012). "Practical Bayesian optimization of machine learning algorithms," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2* (Red Hook, NY: Curran Associates, Inc.), 2951–2959.

Suzuki, S., and Okada, M. (2005). Residual energies after slow quantum annealing. *J. Phys. Soc. Japan* 74, 1649–1652. doi: 10.1143/JPSJ.74.1649

Tanaka, T., Sako, M., Chiba, M., Lee, C., Cha, H., and Ohzeki, M. (2023). Virtual screening of chemical space based on quantum annealing. *J. Phys. Soc. Japan* 92, 023001. doi: 10.7566/JPSJ.92.023001

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 285–294.

Urushibata, M., Ohzeki, M., and Tanaka, K. (2022). Comparing the effects of Boltzmann machines as associative memory in generative adversarial networks between classical and quantum samplings. *J. Phys. Soc. Japan* 91, 074008. doi: 10.7566/JPSJ.91.074008

Vanchinathan, H. P., Nikolic, I., Bona, F. D., and Krause, A. (2014). "Explore-exploit in top-n recommender systems via Gaussian processes," in *Proceedings of the 8th ACM Conference on Recommender Systems* (New York, NY: ACM).

Venturelli, D., and Kondratyev, A. (2019). Reverse quantum annealing approach to portfolio optimization problems. *Quant. Mach. Intell.* 1, 17–30. doi: 10.1007/s42484-019-00001-w

Yamamoto, M., Ohzeki, M., and Tanaka, K. (2020). Fair sampling by simulated annealing on quantum annealer. *J. Phys. Soc. Japan* 89, 025002. doi: 10.7566/JPSJ.89.025002

Yonaga, K., Miyama, M., Ohzeki, M., Hirano, K., Kobayashi, H., and Kurokawa, T. (2022). Quantum optimization with Lagrangian decomposition for multiple-process scheduling in steel manufacturing. *ISIJ Int.* 62, 1874–1880. doi: 10.2355/isijinternational.ISIJINT-2022-019

frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 05 January 2024
DOI 10.3389/fcomp.2023.1286657

OPEN ACCESS

EDITED BY
Susan Mniszewski,
Los Alamos National Laboratory (DOE),
United States

REVIEWED BY
Nga Nguyen-Fotiadis,
Los Alamos National Laboratory (DOE),
United States
Hayato Ushijima Mwesigwa,
Fujitsu Laboratories, United States

*CORRESPONDENCE
Sridhar Tayur
✉ stayur@andrew.cmu.edu

RECEIVED 31 August 2023
ACCEPTED 07 November 2023
PUBLISHED 05 January 2024

CITATION
Guddanti SS, Padhye A, Prabhakar A and Tayur S (2024) Pneumonia detection by binary classification: classical, quantum, and hybrid approaches for support vector machine (SVM). *Front. Comput. Sci.* 5:1286657. doi: 10.3389/fcomp.2023.1286657

Pneumonia detection by binary classification: classical, quantum, and hybrid approaches for support vector machine (SVM)

Sai Sakunthala Guddanti[1], Apurva Padhye[2], Anil Prabhakar[3] and Sridhar Tayur[4]*

[1]Centre for Quantum Information Communication and Computing, Indian Institute of Technology Madras, Chennai, India, [2]Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, United States, [3]Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai, India, [4]Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, United States

Early diagnosis of pneumonia is crucial to increase the chances of survival and reduce the recovery time of the patient. Chest X-ray images, the most widely used method in practice, are challenging to classify. Our aim is to develop a machine learning tool that can accurately classify images as belonging to normal or infected individuals. A support vector machine (SVM) is attractive because binary classification can be represented as an optimization problem, in particular as a Quadratic Unconstrained Binary Optimization (QUBO) model, which, in turn, maps naturally to an Ising model, thereby making annealing—classical, quantum, and hybrid—an attractive approach to explore. In this study, we offer a comparison between different methods: (1) a classical state-of-the-art implementation of SVM (LibSVM); (2) solving SVM with a classical solver (Gurobi), with and without decomposition; (3) solving SVM with simulated annealing; (4) solving SVM with quantum annealing (D-Wave); and (5) solving SVM using Graver Augmented Multi-seed Algorithm (GAMA). GAMA is tried with several different numbers of Graver elements and a number of seeds using both simulating annealing and quantum annealing. We found that simulated annealing and GAMA (with simulated annealing) are comparable, provide accurate results quickly, competitive with LibSVM, and superior to Gurobi and quantum annealing.

KEYWORDS

quantum annealing, quantum machine learning, binary classification, Graver Augmented Multi-seed Algorithm, support vector machine

1 Introduction

Pneumonia is a major disease which is prevalent across the globe. Caused by the bacteria and viruses in the air we breathe, the illness affects one or both of the lungs, creating difficulty in breathing. Pneumonia accounts for more than 15% of deaths in children younger than 5 years old (World Health Organization, 2022). Therefore, early and accurate diagnosis of pneumonia is crucial to prevent death and ensure better treatment.

There are many widely used tests to diagnose pneumonia, such as chest X-rays, chest MRI, and needle biopsy of the lung. Chest X-ray imaging is the most commonly used method, as it is relatively inexpensive and non-invasive. Figure 1 shows examples of healthy and pneumonic lung X-rays. However, the examination of chest X-rays is challenging and sensitive to subjective variability. Machine learning (ML) techniques have gained popularity

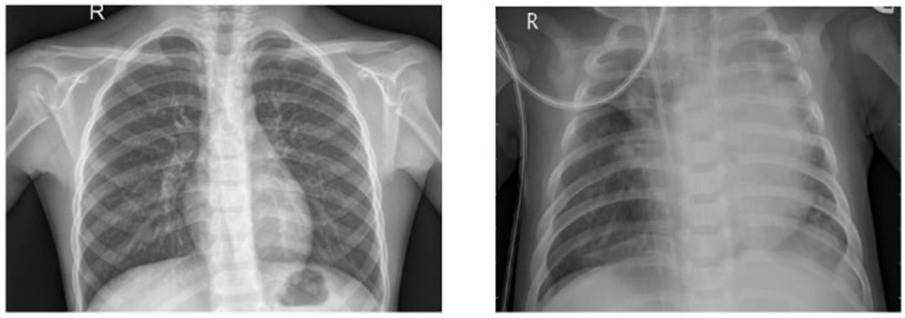


FIGURE 1
The image on the **left** shows a normal chest X-ray, whereas the one on the **right** shows lungs with pneumonia opacity (Breviglieri, 2019).

for solving the image classification problem and have found their use in pneumonia diagnosis as well. Support vector machine (SVM) is a widely used method for classification. We have the added advantage of being able to reframe the SVM as a Quadratic Unconstrained Binary Optimization (QUBO) problem, making it especially suitable for studying annealing methods. In this study, we computationally evaluate a variety of SVM methods, in the context of X-ray imaging for pneumonia, and compare our results against LibSVM, a state-of-the art implementation of SVM. Our main contributions include:

1. Studying a QUBO formulation of an SVM using simulated annealing (SA) and quantum annealing (QA).
2. Solving a QUBO with Gurobi and comparing with annealing methods.
3. Combining multiple weak SVMs to get a strong classification model to accommodate fewer qubits on NISQ quantum annealers.
4. Studying a hybrid quantum-classical optimization heuristic technique, Graver Augmented Multi-seed Algorithm (GAMA).

2 Related work with CNNs and SVMs

Nagashree and Mahanand (2023) compared the performance of an SVM with a few other classification algorithms, such as decision tree, naïve Bayes, and K nearest neighbor. The comparison results indicate a better performance of SVMs for diagnosing pneumonia. Darici et al. (2020) and Kundu et al. (2021) developed an ensemble framework and implemented it with deep learning models to boost their individual performance.

Many researchers have explored, using different data sets, comparing between classical and quantum machine learning algorithms. Willsch et al. (2020) introduced a method to train an SVM on a D-Wave quantum annealer and studied its performance in comparison to classical SVMs for both synthetic data and real data obtained from biology experiments. Wang et al. (2022) implemented an SVM, enhanced with quantum annealing, for two fraud detection data sets. They observed a potential advantage of using an SVM with quantum annealing, over other classical approaches, for bank loan time series data. Delilbasic et al. (2021) implemented two formulations of a quantum support vector machine (QSVM) using IBM quantum computers and D-Wave quantum annealers and compared the results for remote sensing (RS) images. Bhatia and Phillipson (2021) compared classical approach, simulated annealing, hybrid solver, and fully quantum implementations for public Banknote Authentication dataset and the Iris Dataset.

Researchers have also studied convolutional neural networks (CNN) in this context. Although it is not the focus of our study, we mention the related literature. Sirish Kaushik et al. (2020) implemented four models of CNNs and reached an accuracy of 92.3%. Nakrani et al. (2020) and Youssef et al. (2020) implemented deep learning models (different types of CNNs) to classify the data. Madhubala et al. (2021) extended the classification to more than two types of pneumonia. They used CNNs for classification and later performed augmentation to obtain the final results. Ibrahim et al. (2021) considered bacterial pneumonia, non-COVID viral pneumonia, and COVID-19 pneumonia chest X-ray images. They performed multiple experiments with binary and multi-class classification and achieved a better accuracy in identifying COVID-19 (99%) than normal pneumonia (94%).

3 Background information

3.1 QUBO formulation of SVM

Recalling that SVM is a supervised machine learning model. The hyperplane produced by the SVM maximizes its distance between the two classes. Figure 2 shows the support vectors, and the hyperplane classifies data into two classes (labels +1 and −1).

Given training data $X \in \mathbb{R}^{N\times d}$ and training labels $Y \in \{-1,+1\}^N$, where N is the number of training data points, we look for a hyperplane determined by weights, $w \in \mathbb{R}^d$, and bias, $b \in \mathbb{R}$, to separate the training data into two classes. Mathematically, the SVM is expressed as (Date et al., 2021) follows:

$$\min_{w,b} \frac{1}{2}\|w\|^2, \tag{1}$$
$$\text{subject to } y_i(w^T x_i + b) \geq 1, \quad \forall i = 1, 2, \ldots, N.$$

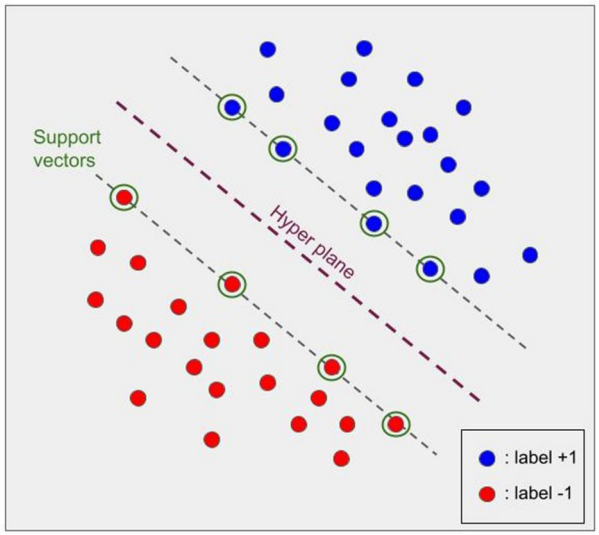


FIGURE 2
Representation of hyperplane in SVM separating two classes of data.

where, x_i is the i-th row vector in X and y_i is the i-th element in Y. The Lagrangian function of this optimization problem is as follows:

$$\mathcal{L}(w, b, \lambda) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \lambda_i [y_i(w^T x_i + b) - 1], \tag{2}$$

where λ is the vector containing all the Lagrangian multipliers, that is, $\lambda = [\lambda_1, \ldots, \lambda_N]^T$, with $\lambda_i \geq 0, \ \forall i$. Each Lagrange multiplier or support vector corresponds to one image and represents the significance of that particular image in determining the hyperplane. Converting the above primal problem to its dual form yields a QUBO (Date et al., 2021)

$$\min_{\lambda} \mathcal{L}(\lambda) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j (x_i^T x_j) - \sum_{i=1}^{N} \lambda_i, \tag{3}$$

with the final weights determined as

$$w = \sum_{i=1}^{N} \lambda_i y_i x_i, \tag{4}$$

$$\sum_{i=1}^{N} \lambda_i y_i = 0, \tag{5}$$

and $\lambda_i, \lambda_j \geq 0, \forall i, j$. Since the data are linearly inseparable, we use a kernel function to plot the input data to higher dimensions and use the SVM on the higher dimensional data. The kernel matrix is defined as follows

$$K_{ij} = \phi(x_i)\phi(x_j), \quad \forall i, j, \tag{6}$$

where $\phi(x_i)$ is some function of the input vector x_i. In this study, we have used the radial basis function (RBF) as it can project data efficiently. Mathematically, the RBF is defined as follows:

$$K(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right). \tag{7}$$

The value of σ was chosen as 50 by trial. Substituting the RBF from (7) in (3) yields the QUBO as follows:

$$\min_{\lambda} \mathcal{L}(\lambda) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j (K_{ij}) - \sum_{i=1}^{N} \lambda_i. \tag{8}$$

The Lagrange multipliers should also satisfy the condition in (5). Writing (8) as a matrix yields

$$\min_{\lambda} \mathcal{L}(\lambda) = \frac{1}{2}\lambda^T (K \odot YY^T)\lambda - \lambda^T \mathbb{1}_N, \ \lambda \geq \mathbb{0}_N. \tag{9}$$

where, K is the kernel matrix whose elements are defined by (6). $\mathbb{1}_N$ and $\mathbb{0}_N$ represent N-dimensional vectors of ones and zeros, respectively, and $\odot$ is the element-wise multiplication operation. This QUBO matrix becomes the input to an annealer (Ising solver) that solves the minimization objectives and returns the Lagrange multipliers (binary) or the support vectors.

The precision vector is introduced to have integer support vectors instead of only binary, and the dimension of the precision vector depends on the range of integer values for the support vector. The precision vector has powers of 2 as elements, and here, we use $p = [2^0, 2^1]$ to get the final QUBO matrix. Now, the dimensions of the QUBO have doubled, and our support vectors can be four integers (0,1,2,3) instead of just being binary. Let $\hat{\lambda} = [\lambda_{11}, \lambda_{12}, \ldots, \lambda_{N1}, \lambda_{N2}]$ be the expanded Lagrange multiplier vector, which gives us our final QUBO. We pass the QUBO matrix to an annealer (Ising solver). The final $\hat{\lambda}$ vector obtained minimizes the QUBO

$$\min_{\hat{\lambda}} \mathcal{L}(\hat{\lambda}) = \frac{1}{2}\hat{\lambda}^T P^T (K \odot YY^T) P\hat{\lambda} - \hat{\lambda}^T P^T \mathbb{1}_N, \tag{10}$$

where $P = I_n \otimes p$ and $\lambda = P\hat{\lambda}$. The annealer returns expanded Lagrange multipliers $\hat{\lambda}$, which we use to calculate support vectors λ. We can predict the labels for unseen data using λ as follows:

$$\text{label}(x) = \text{sign}\left(\sum_{i=1}^{N} \lambda_i y_i (K_{xi}) + b\right), \tag{11}$$

$$b = \text{mean}(y_i - w^T x_i), \quad \text{where } i \in [0, \ldots, N], \tag{12}$$

$$w^T x_i = \sum_{j=1}^{N} \lambda_j y_j K_{ji},$$

with K_{xi} being the kernel between the new test point x and training data point i as defined in (6).

3.2 Graver Augmented Multi-seed Algorithm (GAMA)

Let our binary optimization problem be of the form:

$$\text{objective function: } \min f(x)$$
$$\text{constraints: } Ax = b.$$

Alghassi et al. (2019a) introduced a novel fusion of quantum and classical methodologies for computation of Graver basis. In

the study by Alghassi et al. (2019b), the heuristic was given the acronym GAMA—Graver Augmented Multiseed Algorithm—and the authors studied the application of Graver basis (computed classically) as a means to attain good solutions. In this article, we explore the performance of GAMA in the context of solving an SVM.

GAMA is a heuristic algorithm, in which we compute a partial Graver basis and obtain many feasible solutions using Ising solvers. The motivation for GAMA comes from the theoretical foundation that a complete Graver basis is a *Test-Set* for a wide variety of objective functions (Graver, 1975; Murota et al., 2004; De Loera et al., 2009; Lee et al., 2010; Hemmecke et al., 2011). Of course, for most realistic size problems, it is not possible to identify a complete Graver basis (Pottier, 1996), but in some cases, it is much simpler to establish a partial Graver basis, especially when QUBOs are solved using Ising solvers. We therefore rely on the existence of several feasible solutions to compensate for this incompleteness of the Graver basis. Consequently, the GAMA heuristic selects the best among the (possibly) local optimal solutions by performing a (partial) Graver walk from each of the possible solutions as the seed (hence the term "multiseed"). For finding the Graver bases, we consider the QUBO form of the constraint matrix $Ax = 0$. The Ising solver gives us many kernel elements, and performing conformal filtration on these kernel elements gives us the partial Graver bases. To get feasible solutions, we take the QUBO form of the constraint matrix $Ax = b$ (and solve it using an Ising solver). An alternative is to find kernel elements as differences of the feasible solutions and thus partial Graver bases and augment every feasible solution using the Graver bases to obtain solutions that are likely only a local optimum. To be clear, we have the following steps:

1. Find (partial) Graver basis (either by finding several kernel elements by solving a QUBO for $Ax = 0$ or taking differences of feasible solutions found in step 2);
2. Find feasible solutions by solving a QUBO for $Ax = b$;
3. Augment the feasible solutions using partial Graver basis elements, computing the objective function value $f(x)$ at each step, and choosing the best solution among all (potentially) local optimal solutions.

4 Data and pre-processing

The data set used is from Kaggle (Breviglieri, 2019) (Kaggle, RRID:SCR_013852): 1,000 images from each of the normal and opacity classes are used for training the SVM, while 267 images from the normal class and 1,000 images from the opacity class are used to test the trained model for evaluation of performance. Originally, the images are of different sizes and dimensions. Therefore, the images are first resized to 200 × 200 pixels. The resized images are then flattened to give 1-dimensional arrays of 40,000 pixels.

Although the original data set in Kaggle contains more than 4,000 images, we have considered only 2,000 training images. In the dataset, we observed 1,082 normal images available for training, while there are more than 3,000 images with signs of pneumonia. To get unbiased results from the ML models, we began our training with a balanced dataset. Thus, we considered 1,000 normal images and 1,000 opacity images as the data set in our studies.

5 Methods

We begin with a discussion of each method.

5.1 Method 1: LibSVM (benchmark)

LibSVM is a state-of-the-art library that implements support vector machine (Chang and Lin, 2011) using the input data sets directly, without going through the formulation of a QUBO. The results from LibSVM are typically considered to be a benchmark to compare other newer methods.

5.2 Method 2: SVM using Gurobi

An SVM modeled as QUBO, as in (10), can be solved using a state-of-the-art classical solver, such as Gurobi (version 9.5.0). This is implemented in two ways as follows:

1. All 2,000 training images are taken at once and incorporated into the QUBO. The solver returns expanded Lagrange multipliers as an array of 4,000 elements, using which we construct 2,000 support vector values and make predictions on test data.
2. The training set is divided into 40 sets, each of 50 images. Every set represents an SVM. The 40 SVMs are solved separately and combined using majority voting bagging (Kim et al., 2002). This approach is discussed in detail in Section 5.3.

5.3 Method 3: SVM using annealing

We used the D-Wave neal simulated annealer, digital annealer from IITM, and the Advantage_system 6.2 from D-Wave with 5614 qubits with the Pegasus connectivity between them (Dattani et al., 2019) as our three Ising solver options. Among these, the first two are simulated annealers, while the latter is a quantum annealer.

With additional lenience given for the Lagrange multipliers using a precision vector, the QUBO matrix for 2,000 input images has a size of 4000 × 4000. This is beyond the processing capacity of simulated annealing using D-Wave neal and D-Wave quantum annealing. To overcome this, we opted to partition the images into 20 distinct sets, each comprising 100 images, giving a QUBO matrix of size 200 × 200, which can be solved with simulated annealers while still remaining challenging for quantum annealing platforms.

Subsequently, we refined our strategy by further dividing the images into 40 sets, each encompassing 50 images (25 from each class). As a result, there are 40 SVMs (40 QUBO matrices) of size 100 × 100. These 40 SVMs are trained separately, and their outputs are combined using the majority voting bagging technique (Kim et al., 2002) to obtain the final decision boundary for classification. This framework is presented in Figure 3.

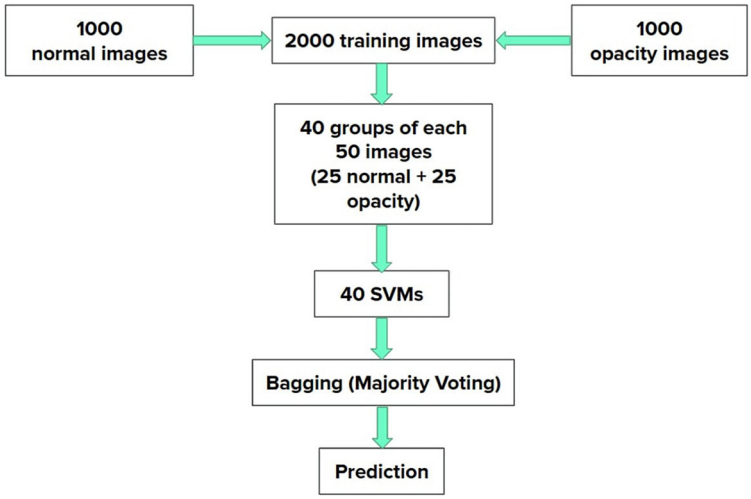


FIGURE 3
Flowchart of the steps involved in our proposed method for utilizing SVM using annealing.

5.3.1 Method 3(a): simulated annealing

5.3.1.1 Simulated annealing using the D-Wave neal package

The 40 QUBOs corresponding to 40 SVMs are solved individually using a simulated annealer, with 1,000 iterations each per SVM. The output of the annealer is the set of expanded Lagrange multipliers for all the 1,000 iterations. We filter the one which gives the minimum energy among 1,000 iterations for every SVM and thus obtain 40 sets of expanded Lagrange multipliers for 40 SVMs, using which we get our final support vectors. The 40 SVMs are combined using the majority voting bagging technique, and the prediction of unseen test data is carried out by (11). The simulated annealer was configured using the default parameter values specified by D-Wave neal in our study.

5.3.1.2 Simulated annealing using the digital annealer of IITM

In the utilization of the Digital Annealer for simulated annealing, it was essential to designate parameter values, that is, the starting and ending temperature and iterations to perform at every temperature while descending. We converted all 40 QUBOs to Ising formulations and gave them as input to the digital annealer. The annealer performs one round of annealing from starting temperature to ending temperature with a specified number of iterations at every step. We took the initial temperature to be 6.4K, the final temperature to be 0.001K, and iterations at every step to be 20. The output we get would be the final spin values of the Ising formulation and its final energy value. We take the spin values output for all 40 SVMs which are expanded Lagrange multipliers and calculate support vectors. These are combined using majority voting, and prediction for unseen test data is done by using equation (11).

5.3.2 Method 3(b): quantum annealing

The procedure resembles that of simulated annealing with D-Wave neal. Here, instead of 1000, we have taken 500 iterations of the D-Wave quantum annealer. It is important to note that, unlike simulated annealing, quantum annealers often have substantial queue times.

5.4 Method 4: SVM using GAMA

GAMA can be a very efficient method when the objective function is complex but the constraints are simple (Alghassi et al., 2019b). We give the simpler constraints to the annealer, obtain partial Graver elements and feasible solutions, and do a walkback using the initial objective function to obtain a final solution. The constraint equation is given in (5).

To ensure that the algorithm does not get stuck in a local minimum while performing augmentation, we implement a Metropolis-Hastings version of GAMA. In this case, we consider the probability of moving in any of the directions according to the ratio in the objective function value and not just in the direction of improvement. We end the augmentation iterations if the change in objective function value remains constant for more than ten iterations.

5.4.1 Method 4(a): GAMA using simulated annealing

We tested simulated annealing from D-Wave and the Digital Annealer from IITM. Similar to the method 3 (Section 5.3), the images are divided into 40 sets (40 SVMs). Recalling that we use the constraint mentioned in (5) to get Graver bases and feasible solutions:

$$\sum_{i=1}^{N} \lambda_i y_i = 0. \tag{13}$$

TABLE 1 Confusion matrix values and time taken for the following methods, respectively: LibSVM (Classical state-of-the-art implementation of SVM), Gurobi1 (Gurobi using all images at once), Gurobi2 (Gurobi with images split into 40 sets), SimAnn-Dn (Simulated Annealing using D-Wave neal), SimAnn-Di (Simulated Annealing using the Digital Annealer from IITM), QuantumAnn (Quantum Annealing with D-Wave), Simulated Annealing using D-Wave neal with GAMA (50 Graver + 50 feasible solutions), Simulated Annealing using D-Wave neal with GAMA (100 Graver + 100 feasible solutions), Simulated Annealing using D-Wave neal with GAMA (200 Graver + 200 feasible solutions), Simulated Annealing using the Digital annealer from IITM with GAMA (499 Graver + 499 feasible solutions), and Quantum Annealing with GAMA run on D-Wave quantum annealer (127 feasible solutions + 127 Graver elements).

Method	True +ve	False +ve	True −ve	False −ve	Time taken
LibSVM	917	19	248	83	3 min 30 s
Gurobi1	712	11	256	288	30 min
Gurobi2	860	111	156	140	2.44 s
SimAnn-Dn	927	22	245	73	6 min 29 s
SimAnn-Di	884	20	247	116	20 s
QuantumAnn	924	46	221	76	12 s
GAMA1	862	28	239	138	10 s (anneal) + 7 s (aug)
GAMA2	900	36	231	100	10 s (anneal) + 36 s (aug)
GAMA3	924	33	234	76	10 s (anneal) + 153 s (aug)
GAMA-Di	885	67	200	115	256 s (anneal) + 1,196 s (aug)
GAMA-Q	875	9	258	125	0.3 s (anneal) + 92 s (aug)

In the table, "aug" represents augmenting time. Quantum annealer time represents only quantum processor time. We are reporting the best of three runs for all annealing methods.

The constraint matrix (QUBO matrix framed from the above equation) remains the same for all SVMs as the Y vector (labels vector) remains the same for all 40 SVMs (each SVM has 25 normal and then 25 opacity images). As the right-hand part of constraints is zero, kernel elements and the feasible solutions are also the same. This special structure implies that a single execution of the annealer is sufficient to address the optimization requirements for all 40 SVMs. Thus, the Graver bases and feasible solutions are obtained once and used for augmentation in all SVMs.

A total of 500 feasible solutions (also kernel elements) were obtained by simulated annealing using the D-Wave neal package (from dwave-ocean-sdk). For simulated annealing using the digital annealer of IITM, we have taken the QUBO of constraint mentioned above in (5) and converted it to an Ising formulation. The annealer performs one round of annealing at a time as mentioned in the method 3(a). We took the initial temperature to be 6.4K, the final temperature to be 0.001K, and iterations at every step to be 20. The entire annealing is performed for 500 times. Here, 500 feasible solutions (also kernel elements) are obtained. When conformal filtration is performed, we obtained 499 partial Graver bases.

Detailed experimentation of this method is performed using D-Wave neal simulated annealing. We experimented with three different sets of Graver bases and feasible solutions. The following cases are considered for augmentation:

1. 50 Graver elements + 50 feasible solutions
2. 100 Graver elements + 100 feasible solutions
3. 200 Graver elements + 200 feasible solutions

We obtained 40 sets of Lagrange multipliers corresponding to 40 SVMs for each of the three cases above. The majority of voting bagging is used to combine 40 SVMs, and the final output is tested on the test data set according to equation (11). Using the digital annealer from IITM, we have utilized all 499 partial Graver bases and feasible solutions and performed the augmentation.

5.4.2 Method 4(b): GAMA using D-Wave quantum annealing

The GAMA with quantum annealing process follows a methodology akin to that of GAMA involving simulated annealing. The number of feasible solutions was 127 (as compared with 500 in the earlier method). Notably, out of 500 calls to D-Wave, only 127 gave the minimum energy solution. All 127 feasible solutions and corresponding (partial) Graver elements (computed *via* conformal filtration, which happened to also be 127, likely due to the fact that the kernel elements are short to begin with) were included in the augmentation process.

6 Results and analysis

The results of various methods are compared through confusion matrix representation and associated metrics as we mentioned below. A confusion matrix is a tabular representation used to assess the performance of classification models. It provides a comprehensive overview of how well the predictions of the model align with actual outcomes for different classes or categories. The matrix is constructed by comparing predicted class labels with true class labels for data points. It represents a breakdown of the predictions into four categories: True Positives (TP) represent correctly predicted positive instances, True Negatives (TN) represent correctly predicted negative instances, False Positives (FP) represent instances that are incorrectly predicted as positive when they are actually negative, and False Negatives represent instances that are incorrectly predicted as negative when they are

TABLE 2 Accuracy, precision, recall, and F1 score for all methods.

Method	Accuracy	Precision	Recall	F1 score
LibSVM	91.9	97.9	91.7	94.6
Gurobi1	76.4	98.4	71.2	82.6
Gurobi2	79.8	88.2	86	87
SimAnn-Dn	92.5	97.6	92.7	95
SimAnn-Di	89.2	97.7	88.4	92.8
QuantumAnn	90.3	95.2	92.4	93.7
GAMA1	86.8	96.8	86.2	91.2
GAMA2	89.2	96.1	90	92.9
GAMA3	91.3	96.5	92.4	94.4
GAMA-Di	85.6	92.9	88.5	90.6
GAMA-Q	89.4	98.9	87.5	92.8

We have highlighted the maximum values in each column in red for easy comparison.

actually positive. The confusion matrix helps in evaluating metrics such as accuracy, precision, recall, and F1-score, which help with a deeper understanding of the performance of the model across various classes.

We evaluate various methods on four metrics as follows:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}, \tag{14}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}, \tag{15}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}, \tag{16}$$

$$\text{F1 score} = \frac{2\,\text{TP}}{2\,\text{TP+FP+FN}}. \tag{17}$$

For all the methods, the results are noted from the confusion matrix, which is shown in Table 1 (Recalling that positive means opacity and negative is normal). For quantum annealing, the annealing time including queue time and post-processing for 40 SVMs is 3 h 16 min. In the table, we have removed all these and only provided annealing time. The metrics of comparison for all the methods are presented in Table 2.

6.1 Comparison of methods

Since the running time for each method is different, we cannot draw direct comparisons based on the values of the four metrics. However, Tables 1, 2 provide insight into some key points. All the metrics from Table 2 are plotted in the graph in Figure 4 for visual convenience. We use LibSVM as the classical solver to compare our SVM implementations. As shown in Table 2, the results from other methods, especially SimAnn-Dn, compare favorably against those from LibSVM.

- Gurobi, when given data divided into 40 SVMs, takes the least time (2.44 s), but the performance is weak. When all images are input at once and trained for 30 min, there is no significant improvement in the performance.
- Simulated annealing performed using D-Wave neal takes approximately 6.5 min to run, and the results obtained are good. The best accuracy (92.5%) and F1 score (95%) are achieved with simulated annealing.
- In the case of GAMA, the performance improves as we increase the number of Graver elements taken for augmentation. The augmentation time taken also increases accordingly (it reaches a threshold value of performance as in Supplementary Figures 2, 4, See Appendix). Indeed, using 200 feasible solutions and 200 Graver elements appears sufficient to reach good performance relatively quickly.
- GAMA when implemented using quantum annealing takes approximately 8.5 min (including queue time) and provides accuracy similar to that of SVM using quantum annealing [Method 3(b)]. Here, we can observe a massive speed-up as method 3(b) takes more than 3 h to run. Thus, despite limited connectivity, GAMA provides a significant time improvement for quantum annealing, without compromising on the metrics.
- Quantum annealers often have a lower precision for encoding QUBO coefficients. However, we found that this did not affect the results because the QUBO matrix elements ranged between 0 and 2 or between 0 and 4 when we used GAMA.

Among our approaches, for a given time budget (of training), the best methods are as follows:

1. **5 min**: GAMA 3 (200 Graver elements + 200 feasible solutions).
2. **10 min**: Simulated annealing [method 3(a)] and GAMA 3 (200 Graver elements + 200 feasible solutions).
3. **20 min**: Simulated annealing [method 3(a)] and GAMA 3 (200 Graver elements + 200 feasible solutions).

Not much improvement is observed by increasing training time.

6.2 Bagging and probability distribution

Majority voting bagging (Kim et al., 2002), the method used to combine SVMs, also improve the performance of the combined SVM. The accuracy of annealing methods [method 3(a) and method 3(b)] without bagging and with bagging is compared in plots (Figures 5, 6).

We can observe that the accuracy improved to 92.5% (Red line in Figure 5) in the case of simulated annealing using D-Wave neal and to 90.3% (Red line in Figure 6) in the case of D-Wave quantum annealing using majority voting bagging.

Many iterations of annealing are taken to find the Lagrange multipliers that best minimize the objective function value. It is instructive to know how often we might get the parameters that give the minimum objective function value. From Figures 5, 6, we also observe that some of the individual SVMs also give sufficiently good results. Thus, there maybe an opportunity to reduce computational time (by only solving a few SVMs rather than all 40) and obtain good results.

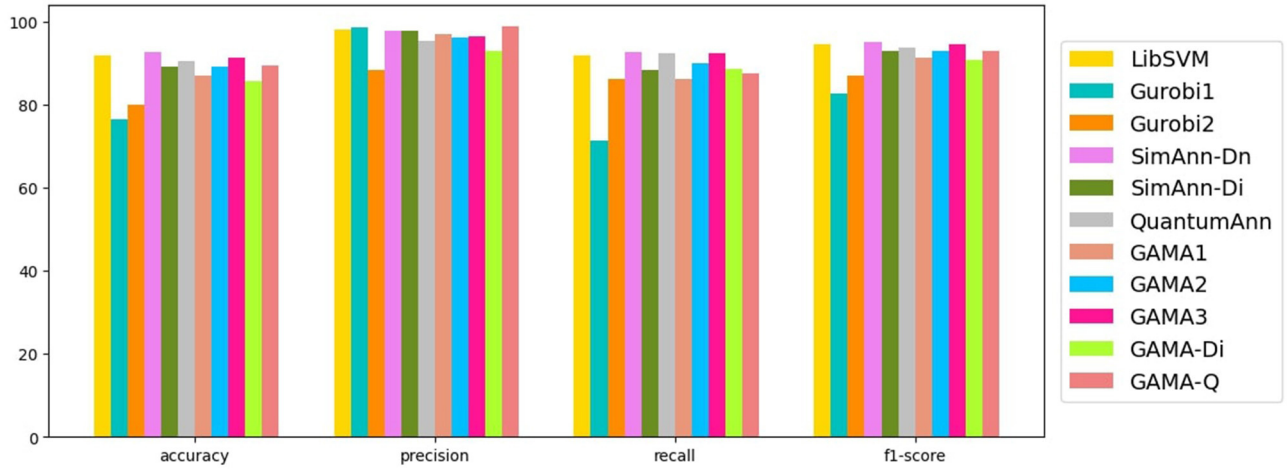


FIGURE 4
Comparative analysis of accuracy, precision, recall, and F1 score for all methods.

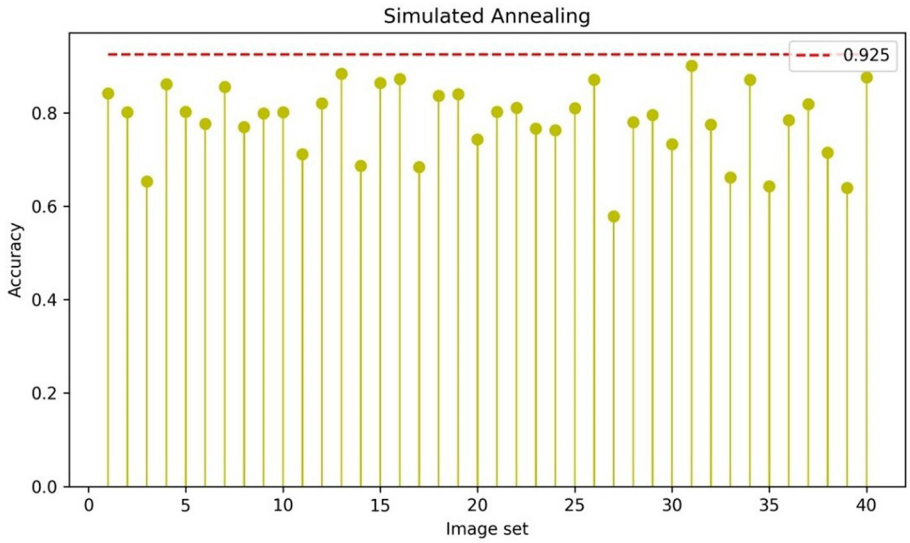


FIGURE 5
The yellow lines represent the accuracy metric for all the 40 SVMs we divided the data into. The red line shows the maximum accuracy achieved using weighted average bagging as 92.5%. All the SVMs are solved with simulated annealing using D-Wave neal.

To understand the probability of obtaining the best solution, we plot the probability distribution for best-performing SVMs (for simulated annealing using D-Wave neal and quantum annealing, respectively). Figure 7 shows the probability distribution for all obtained solutions over 10,000 iterations of simulated annealing for SVM number 31, which gave us the best individual SVM accuracy. We can observe that although our desired low-energy solution occurred with low probability, the median solutions also give good accuracy. Figure 8 shows the probability distribution for all obtained solutions over 8,000 iterations of D-Wave for SVM number 27, which gave us the best individual SVM accuracy. The distribution is similar to that of simulated annealing but did not reach the quality of solutions of simulated annealing.

7 Concluding remarks

In this study, we explored binary classification through classical, quantum, and hybrid methods, using X-ray imaging data for pneumonia, and used LibSVM as our benchmark. To have a balanced data set for SVM, we selected 1,000 images, each, with and without pneumonia as our input data set. We separated the data into 40 sets. We formulated the SVM as a QUBO and solved the QUBOs using simulated annealing and Gurobi and quantum annealing. Additionally, we studied GAMA heuristic, where the (different) QUBOs were solved using simulated annealing and quantum annealing. Each of our data sets yielded an SVM. We used bagging to combine the 40 SVMs, which improved the overall accuracy.

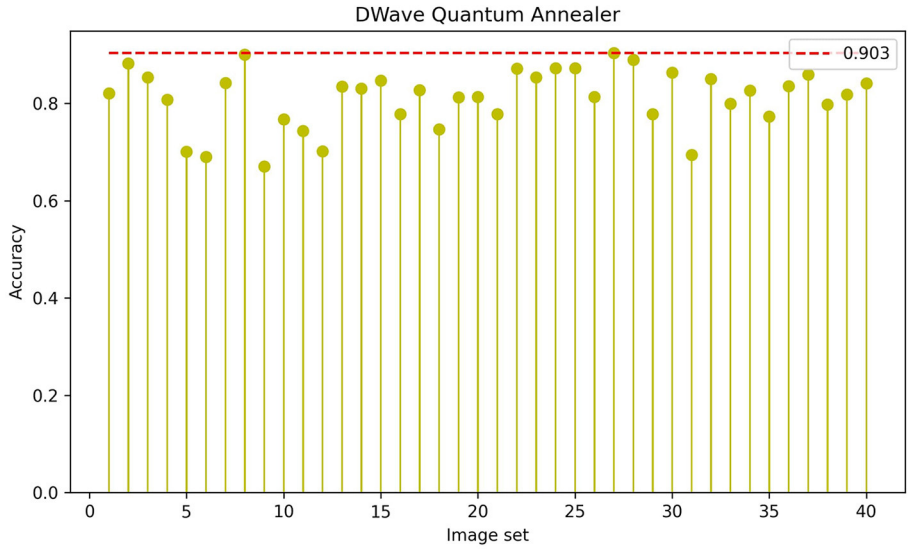


FIGURE 6
The yellow lines represent the accuracy metric for all the 40 SVMs we divided the data into. The red line shows the maximum accuracy achieved using weighted average bagging as 90.3%. All the SVMs are solved using quantum annealing.

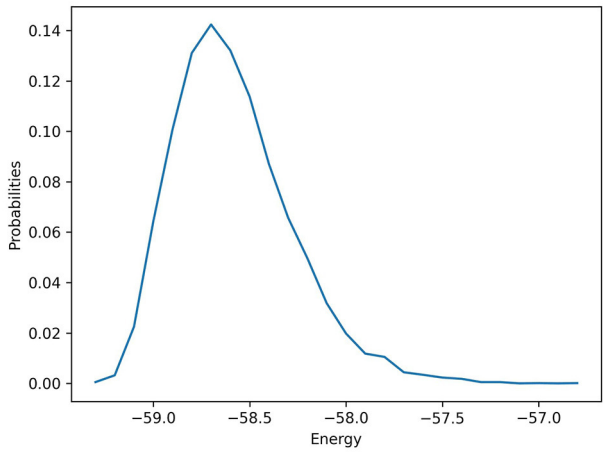


FIGURE 7
Probability distribution of simulated annealing solutions for SVM number 31. The best solution has energy approximately −59.

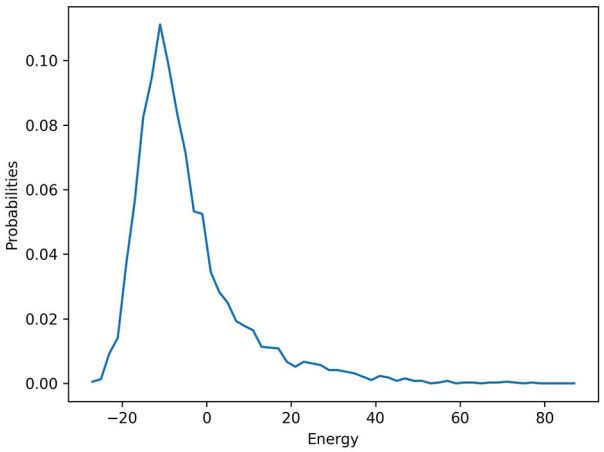


FIGURE 8
Probability distribution of D-Wave quantum annealing solutions for SVM number 27. Notably, the best solution has energy of approximately −20, not as good as that found in simulated annealing.

For binary classification of X-ray images, SVM can be an alternative to CNN, especially when considering pathways to implementations on a quantum annealer. The classical solver, LibSVM, shows a 92% accuracy in classification. However, Simulated Annealing using D-Wave neal (SimAnn-Dn) has comparable or better performance. GAMA provides a speed-up over quantum annealing with the similar performance on metrics. Quantum annealing is not competitive in terms of time taken but provides solutions of quality that are near the best obtained. We anticipate an enhancement in performance when quantum annealers with more qubits and better connectivity become accessible. It is important to acknowledge that improvements in classical hardware and software are also anticipated concurrently. This suggests that periodic comparisons should be encouraged. We hope that our study adds to the literature on the benchmarking of quantum, classical, and hybrid approaches to solve a variety of important combinatorial optimization problems arising from practical applications (Metriq, 2023).

Data availability statement

Publicly available datasets were analyzed in this study. The code is found at: https://github.com/Sai-sakunthala/Pneumonia-Detection-by-Binary-Classification.

Author contributions

SG: Conceptualization, Data curation, Formal analysis, Methodology, Software, Writing – original draft. APa: Conceptualization, Data curation, Formal analysis, Methodology, Software, Writing – review & editing. APr: Funding acquisition, Supervision, Validation, Writing – review & editing. ST: Funding acquisition, Supervision, Validation, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. The authors thank the Mphasis F1 Foundation and RAGS Foundation for supporting this study.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2023.1286657/full#supplementary-material

References

Alghassi, H., Dridi, R., and Tayur, S. (2019a). Graver bases via quantum annealing with application to non-linear integer programs. *arXiv [Preprint]. arxiv:1902.04215*. doi: 10.48550/arXiv.1902.04215

Alghassi, H., Dridi, R., and Tayur, S. (2019b). GAMA: a novel algorithm for non-convex integer programs. *arXiv [Preprint]. arxiv:1907.10930*. doi: 10.48550/arXiv.1907.10930

Bhatia, H. S., and Phillipson, F. (2021). "Performance analysis of support vector machine implementations on the D-Wave quantum annealer," in *Computational Science – ICCS 2021*, eds M. Paszynski, D. Kranzlmüller, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot (Cham: Springer International Publishing), 84–97. doi: 10.1007/978-3-030-77980-1_7

Breviglieri, P. (2019). *Pneumonia X-Ray Images*. Available online at: https://www.kaggle.com/datasets/pcbreviglieri/pneumonia-xray-images (accessed August, 2023).

Chang, C.-C., and Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2, 27:1–27:27. doi: 10.1145/1961189.1961199

Darici, M. B., Dokur, Z., and Olmez, T. (2020). Pneumonia detection and classification using deep learning on chest X-ray images. *Int. J. Intell. Syst. Appl. Eng.* 8, 177–183. doi: 10.18201/ijisae.2020466310

Date, P., Arthur, D., and Pusey-Nazzaro, L. (2021). QUBO formulations for training machine learning models. *Sci. Rep.* 11, 10029. doi: 10.1038/s41598-021-89461-4

Dattani, N., Szalay, S., and Chancellor, N. (2019). Pegasus: the second connectivity graph for large-scale quantum annealing hardware. *arXiv [Preprint]. arxiv:1901.07636*. doi: 10.48550/arXiv.1901.07636

De Loera, J., Hemmecke, R., Onn, S., Rothblum, U., and Weismantel, R. (2009). Convex integer maximization via graver bases. *J. Pure Appl. Algebra* 213, 1569–1577. doi: 10.1016/j.jpaa.2008.11.033

Delilbasic, A., Cavallaro, G., Willsch, M., Melgani, F., Riedel, M., and Michielsen, K. (2021). "Quantum support vector machine algorithms for remote sensing data classification," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2608–2611. doi: 10.1109/IGARSS47720.2021.9554802

Graver, J. E. (1975). On the foundations of linear and integer linear programming I. *Math. Program.* 9, 207–226.

Hemmecke, R., Onn, S., and Weismantel, R. (2011). A polynomial oracle-time algorithm for convex integer minimization. *Math. Program.* 126, 97–117. doi: 10.1007/s10107-009-0276-7

Ibrahim, A. U., Ozsoz, M., Serte, S., Al-Turjman, F., and Yakoi, P. S. (2021). Pneumonia classification using deep learning from chest X-ray images during COVID-19. *Cogn. Comput.* 1–13. doi: 10.1007/s12559-020-09787-5

Kim, H.-C., Pang, S., Je, H.-M., Kim, D., and Bang, S.-Y. (2002). "Support vector machine ensemble with bagging," in *Pattern Recognition with Support Vector Machines* (Berlin; Heidelberg: Springer), 397–408. doi: 10.1007/3-540-45665-1_31

Kundu, R., Das, R., Geem, Z. W., Han, G.-T., and Sarkar, R. (2021). Pneumonia detection in chest X-ray images using an ensemble of deep learning models. *PLoS ONE* 16, e0256630. doi: 10.1371/journal.pone.0256630

Lee, J., Onn, S., Romanchuk, L., and Weismantel, R. (2010). The quadratic graver cone, quadratic integer minimization, and extensions. *Math. Program.* 136, 301–323. doi: 10.1007/s10107-012-0605-0

Madhubala, B., Sarathambekai, S., Vairam, T., Sathya Seelan, K., Sri Sathya, R., and Swathy, A. R. (2021). "Pre-trained convolutional neural network model based pneumonia classification from chest X-ray images," in *Proceedings of the International Conference on Smart Data Intelligence (ICSMDI 2021)*. doi: 10.2139/ssrn.3852043

Metriq (2023). *Community-Driven Quantum Benchmarks*. Available online at: http://metriq.info (accessed August, 2023).

Murota, K., Saito, H., and Weismantel, R. (2004). Optimality criterion for a class of nonlinear integer programs. *Oper. Res. Lett.* 32, 468–472. doi: 10.1016/j.orl.2003.11.007

Nagashree, S., and Mahanand, B. S. (2023). "Pneumonia chest X-ray classification using support vector machine," in *Proceedings of International Conference on Data Science and Applications*, eds M. Saraswat, C. Chowdhury, C. Kumar Mandal, and A. H. Gandomi (Singapore: Springer Nature), 417–425. doi: 10.1007/978-981-19-6634-7_29

Nakrani, N. P., Malnika, J., Bajaj, S., Prajapati, H., and Jariwala, V. (2020). "Pneumonia identification using chest X-ray images with deep learning," in *ICT Systems and Sustainability*, eds M. Tuba, S. Akashe, and A. Joshi (Singapore: Springer Singapore), 105–112. doi: 10.1007/978-981-15-0936-0_9

Pottier, L. (1996). "The Euclidean algorithm in dimension n," in *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation, ISSAC '96* (New York, NY: Association for Computing Machinery), 40–42. doi: 10.1145/236869.236894

Sirish Kaushik, V., Nayyar, A., Kataria, G., and Jain, R. (2020). "Pneumonia detection using convolutional neural networks (CNNs)," in *Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019)*, eds P. K. Singh, W. Pawłowski, S. Tanwar, N. Kumar, J. J. P.

C. Rodrigues, and M. S. Obaidat (Singapore: Springer Singapore), 471–483. doi: 10.1007/978-981-15-3369-3_36

Wang, H., Wang, W., Liu, Y., and Alidaee, B. (2022). Integrating machine learning algorithms with quantum annealing solvers for online fraud detection. *IEEE Access* 10, 75908–75917. doi: 10.1109/ACCESS.2022.3190897

Willsch, D., Willsch, M., De Raedt, H., and Michielsen, K. (2020). Support vector machines on the D-Wave quantum annealer. *Comput. Phys. Commun.* 248, 107006. doi: 10.1016/j.cpc.2019.107006

World Health Organization (2022). *Pneumonia in Children*. Available online at: https://www.who.int/news-room/fact-sheets/detail/pneumonia (accessed November, 2022).

Youssef, T. A., Aissam, B., Khalid, D., Imane, B., and Miloud, J. E. (2020). "Classification of chest pneumonia from X-ray images using new architecture based on ResNet," in *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, 1–5. doi: 10.1109/ICECOCS50124.2020.9314567

frontiers | Frontiers in Computer Science

TYPE Mini Review
PUBLISHED 05 January 2024
DOI 10.3389/fcomp.2023.1286860

OPEN ACCESS

EDITED BY
Nicholas Chancellor,
Durham University, United Kingdom

REVIEWED BY
Zhihao Lan,
University College London, United Kingdom
Jesse Berwald,
Quantum Computing Inc., Leesburg, VA, United States

*CORRESPONDENCE
Sridhar Tayur
✉ stayur@andrew.cmu.edu

RECEIVED 31 August 2023
ACCEPTED 25 October 2023
PUBLISHED 05 January 2024

CITATION
Tayur S and Tenneti A (2024) Quantum annealing research at CMU: algorithms, hardware, applications.
Front. Comput. Sci. 5:1286860.
doi: 10.3389/fcomp.2023.1286860

Quantum annealing research at CMU: algorithms, hardware, applications

Sridhar Tayur* and Ananth Tenneti

Quantum Technologies Group, Carnegie Mellon University, Pittsburgh, PA, United States

In this mini-review, we introduce and summarize research from the Quantum Technologies Group (QTG) at Carnegie Mellon University related to computational experience with quantum annealing, performed in collaboration with several other institutions including IIT-Madras and NASA (QuAIL). We present a novel hybrid quantum-classical heuristic algorithm (GAMA, Graver Augmented Multi-seed Algorithm) for non-linear, integer optimization, and illustrate it on an application (in cancer genomics). We then present an algebraic geometry-based algorithm for embedding a problem onto a hardware that is not fully connected, along with a companion Integer Programming (IP) approach. Next, we discuss the performance of two photonic devices - the Temporal Multiplexed Ising Machine (TMIM) and the Spatial Photonic Ising Machine (SPIM) - on Max-Cut and Number Partitioning instances. We close with an outline of the current work.

KEYWORDS

quantum annealing, Combinatorial Optimization, Photonic Ising Machines, Graver basis, cancer genomics

1 Introduction

Quantum annealing has emerged as a promising approach because a variety of Combinatorial Optimization (CO) problems that arise in practical situations (Smelyanskiy et al., 2012; Tanahashi et al., 2019; Hauke et al., 2020) can be formulated as a Quadratic Unconstrained Binary Optimization (QUBO) problem, which maps naturally to an Ising model, and solved on specially constructed quantum and semi-classical hardware (Wang et al., 2013; Lucas, 2014; McMahon et al., 2016; Glover et al., 2018; Harris et al., 2018; King et al., 2018; Chou et al., 2019; Wang and Roychowdhury, 2019; Mohseni et al., 2022). A lucid introduction to quantum annealing can be found in McGeoch (2014).

At Carnegie Mellon University's Quantum Technologies Group (QTG), we have been working on several initiatives[1] related to computational aspects of quantum annealing (Figure 1).

1. While unconstrained optimization problems expressed as QUBO can be directly passed to an annealer solver, it is also of practical interest to develop scalable decomposition methods that solve general non-linear constrained optimization problems. We describe a novel quantum-classical algorithm, Graver Augmented Multiseed Algorithm (GAMA) (Alghassi et al., 2019b,c) for solving such optimization problems, building on previous work on the use of algebraic geometry for Integer Optimization with a linear objective

1 QTG is also engaged in theoretical research on understanding speed up in adiabatic quantum computing (Dridi et al., 2018b, 2019a), and other connections between algebraic geometry and Ising models (Dridi et al., 2019b), topics not covered here.

		Application
Algorithm for Non-Linear Integer Optimization	Graver Augmented Multiseed Algorithm (GAMA)	Cancer Genomics Binary Classification
Algorithms for Embedding	Algebraic Geometry Integer Programming	D-Wave D-Wave
Photonic Ising Machines (Hardware)	Time Multiplexed (TMCIM) Spatial (SPIM)	Max-Cut Number Partitioning

FIGURE 1
Some computational quantum annealing initiatives at Quantum Technology Group (QTG).

function (Tayur et al., 1995). GAMA is motivated by test sets, and (a) uses partial Graver bases (Graver, 1975) instead of the complete Graver basis and (b) many feasible solutions as starting points (rather than just one) for augmentation. Both the partial Graver basis and a number of feasible solutions are obtained from the constraint equation expressed as QUBOs (that is solved by an annealer). A particular advantage of this algorithm is that it separates the constraints from the objective function, allowing us to tackle situations where the computation of objective function value may need an oracle call (such as a simulation).

2. Many devices such as D-Wave have limited coupling connectivity between qubits. For a dense problem graph, it is therefore necessary to develop a mapping - minor embedding—to the sparse hardware graph. We have developed two methods, based on algebraic geometry and Integer Programming (Dridi et al., 2018a; Bernal et al., 2020).
3. Building fully connected Ising hardware is another exciting area of current research. We have re-constructed (with some refinements) two Photonic Ising Machines (PIM), building on the time-multiplexed coherent Ising machine (TMCIM) (Böhm et al., 2019) and the spatial multiplexed Ising machine (SPIM) (Pierangeli et al., 2020). We have studied the performance of the annealers on Max-Cut and Number Partitioning Problem with D-Wave (McGeoch, 2014) and Gurobi (LLC Gurobi Optimization, 2023).

The rest of the review is organized as follows. In Section 2, we describe GAMA. In Section 3, we illustrate the application of the GAMA on identifying altered pathways in cancer genomics as a proof-of-principle and recovering known results. An algebraic geometry-based embedding algorithm and its Integer Programming reformulation are outlined in Section 4 and compared to the default heuristic that is used by D-Wave. The performance of two Photonic Ising Machines is discussed in Section 5. We conclude in Section 6.

2 Graver augmented multiseed algorithm (GAMA)

We begin with three definitions, taken verbatim from Alghassi et al. (2019c).

Definition 1. A set $S \in \mathbb{Z}^n$ is a Test Set or an optimality certificate if for every non-optimal, feasible solution, x_0, there exists $t \in S$ and $\lambda \in Z_+$ such that $f(x_0 + \lambda t) < f(x_0)$. The vector, t is called the augmenting direction.

The following partial order is defined on $\mathbb{R}^n$.

Definition 2. Given $x, y \in \mathbb{R}^n$, we define x is conformal to y, written as $x \sqsubseteq y$, if $x_i y_i \geq 0$ (x and y lie in the same orthant), and $|x_i| \leq |y_i|, \forall\, i \in \{1..n\}$. A sum $u = \sum_i v_i$ is called conformal if $v_i \sqsubseteq u, \forall i$.

For a matrix $A \in \mathbb{Z}^{m\times n}$, define the lattice

$$L^*(A) = \{x | Ax = 0, x \in \mathbb{Z}^n, A \in \mathbb{Z}^{m\times n}\} \backslash \{0\}. \tag{1}$$

Definition 3. The Graver basis, $\mathcal{G}(A) \subset \mathbb{Z}^n$, of an integer matrix A is defined as the finite set of $\sqsubseteq$ minimal elements in $L^*(A)$.

The Graver basis (Graver, 1975) of an integer matrix, $A \in \mathbb{Z}^{m\times n}$ is known to be a test set for integer linear programs. Graver basis is also a test set for certain non-linear objective functions including Separable convex minimization (Murota et al., 2004), Convex integer maximization (De Loera et al., 2009), Norm p minimization (Hemmecke et al., 2011), Quadratic (Murota et al., 2004; Lee et al., 2010) and Polynomial minimization (Lee et al., 2010). It has also been shown that for these problem classes, the number of augmentation steps needed is polynomial (De Loera et al., 2009; Hemmecke et al., 2011). Graver basis can be computed (only for small size problems) using classical methods such as the algorithms developed by Pottier (1996) and Sturmfels and Thomas (1997).

At QTG, we are exploring (a) the effectiveness of computing partial Graver basis using annealers by solving a QUBO (for kernel elements) and (b) instead of relying on just one feasible solution as the seed for augmentation, using multiple feasible solutions (that are also obtained via annealing, by solving a second QUBO), as parallel starting points. The GAMA heuristic (Alghassi et al., 2019b,c) thus aims to find good solutions to constrained non-linear optimization problems of the form in Equation 2, using multiple seeds as starting points for augmentation, with partial Graver basis elements as the augmenting directions:

$$(IP)_{A,b,l,u,f} = \begin{cases} min\, f(x) & \\ Ax = b & l \leq x \leq n \quad x, l, u \in \mathbb{Z}^n \\ A \in \mathbb{Z}^{m\times n} & b \in \mathbb{Z}^n \end{cases} \tag{2}$$

where $f : \mathbb{R}^n -> \mathbb{R}$ is a real valued function.

2.1 QUBO for kernel calculation

In order to find a sample of the kernel elements for the constraint matrix A, we solve the Quadratic Unconstrained Integer Optimization (QUIO), given by

$$\begin{aligned} min \quad & x^T Q_I x, \quad Q_I = A^T A, \quad x \in \mathbb{Z}^n \\ & x^T = [x_1, x_2 ... x_n], x_i \in \mathbb{Z}. \end{aligned} \tag{3}$$

Since the inputs to the annealer are binary variables, we create a binary encoding of the integer variable. Writing

$$x = L + EX, \tag{4}$$

with L as the lower bound vector and E as the encoding matrix, the QUIO is equivalent to the QUBO

$$\begin{aligned} min \quad & X^T Q_B X, \quad Q_B = E^T Q_I E + diag(2L^T Q_I E), \\ & X \in \{0,1\}^{nk}, Q_I = A^T A. \end{aligned} \tag{5}$$

The above QUBO is solved by an annealer to obtain kernel elements. A partial Graver basis can be obtained from the kernel elements in a classical post-processing step by $\sqsubseteq$-minimal filtering (Alghassi et al., 2019b).

2.2 QUBO for feasible solutions

Similar to the kernel sampling, the $Ax = b$ constraint can be expressed in the QUIO form as

$$\begin{aligned} min \quad & x^T Q_I x - 2b^T Ax \\ & Q_I = A^T A, x \in \mathbb{Z}^n. \end{aligned} \tag{6}$$

After binary encoding, we get the QUBO given by

$$\begin{aligned} min \quad & X^T Q_B X, \quad Q_B = E^T Q_I E + 2diag[(L^T Q_I - b^T A)E] \\ & X \in \{0,1\}^{nk}, Q_I = A^T A. \end{aligned} \tag{7}$$

The above QUBO can be solved by an annealer to obtain a sample of feasible solutions.

3 An application of GAMA: cancer genomics

As a proof-of-principle testing of GAMA, we describe an application (Alghassi et al., 2019a) to identify cancer pathways **de novo** (Vogelstein and Kinzler, 2004; Haber and Settleman, 2007; Ciriello et al., 2012; Vandin et al., 2012a,b; Zhao et al., 2012) from mutation co-occurrence and mutual exclusivity (Leiserson et al., 2013; Weinberg and Weinberg, 2013).

Data from The Cancer Genome Atlas (TCGA) are now available for a variety of cancers, providing information about which genes are mutated for which patient for any given cancer. With this, we can create a matrix. The rows of the matrix are patients, the columns are the genes, and the elements of the matrix (row i, column j) are zero or one (a binary matrix), where "one" in (row i, column j) means that gene j is mutated for patient i.

However, not all mutations matter. The mutations that do not matter are called passengers. Those mutations that matter are called drivers. We want to isolate drivers from passengers (Most mutations are passengers). Furthermore, the same cancer can manifest itself due to different driver mutations, because different mutated driver genes can impact different cellular signaling and regulatory pathways. This mutational heterogeneity complicates efforts to identify drivers solely by their frequency of occurrence.

A pathway is a collection of genes. To find k pathways means finding k different collections of genes. Each collection of genes can be of a different size. To make the discovery of these pathways computationally manageable, we also make two commonly accepted simplifications:

Simplification 1: A pathway has at most one mutated driver gene. This is because driver genes are quite rare. Thus, two different pathways will not likely share a common driver gene. This is called (mutual) exclusivity.

Simplification 2: A pathway should apply to many patients. This is called coverage. The important thing to note is that even though two patients share a pathway, they can have a different mutated gene from that shared pathway as an explanatory reason for their cancer.

Another complexity that we need to handle is that real data are noisy because of measurement errors and passenger mutations. This means that we cannot impose exclusivity as a hard constraint. Instead, we allow for some overlap or "approximate exclusivity" and this is a parameter in our formulation. Similar considerations force a modification of Simplification 2 as well, in the sense that we now can only reasonably hope that most patients have at least one mutation in a pathway. Recall that mutual exclusivity problems even without the modification above are NP-hard (Karp, 1972).

3.1 Multiple-pathway QUBO formulation for GAMA

Alghassi et al. (2019a) developed a novel formulation tailored for GAMA to discover the cancer pathways. Consider a hypergraph $H_g = (V_g, E_p)$ with incidence matrix B, where each gene (g_i) is represented by a vertex $v_i \in V_g, i = 1, 2, ..., n$ and the mutation list of each patient P_i is represented by a hyperedge $e_i \in E_p, i = 1, 2, ..., m$.

The incidence matrix is mapped to its primal graph (G). This is a graph with the same vertices as that of the hypergraph and with edges between all pairs of vertices contained within the same hyperedge. The primal graph can be expressed in terms of the (positive) Laplacian matrix:

$$L^+(G) = BB^T = D(G) + A(G). \tag{8}$$

The weighted adjacency matrix $A = [a(i,j)]^{n \times n}$ is symmetric and has zero as the diagonal elements. The number of patients that have gene pairs (g_i, g_j) mutated is given by $a(i,j)$. The number of patients with gene g_i mutated is given by the element, d_i in the

degree matrix, $D = diag\{d_1, d_2, ..., d_n\}$, where d_i is the degree of the vertex v_i in the primal graph.

A k-pathway QUBO formulation simultaneously finds k pathways in a single optimization run. The solution vector is represented by the binary vector $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T, i = 1, 2, ..., k$ where, each element x_{ij} indicates if a vertex v_j belongs to the i^{th} pathway. Let $\mathbf{X} = [x_1, x_2, ..., x_k]^T$. The QUBO formulation is given by the following (where $L = (D - A)$ is the negative Laplacian matrix).

$$\begin{aligned} min\ \mathbf{X}^T(\mathbf{Q}_{main} + \alpha\mathbf{Q}_{orth})\mathbf{X} \\ \mathbf{Q}_{main} = -I_k \otimes L \\ \mathbf{Q}_{orth} = (J_k - I_k) \otimes I_n \end{aligned} \tag{9}$$

Note that I_k and I_n are $k \times k$ and $n \times n$ identity matrices. J_k is the $k \times k$ matrix with all entries equal to 1.

Rewriting the system of equations(9) as

$$\begin{aligned} min\ \mathbf{X}^T\mathbf{Q}\mathbf{X} \\ (\mathbf{1}_k^T \otimes I_n)\mathbf{X} \leq \mathbf{1}_n \\ \mathbf{Q} = -I_k \otimes L \end{aligned} \tag{10}$$

brings it in the form suitable for GAMA (Alghassi et al., 2019b).

This is a non-linear (quadratic) non-convex integer problem and of a Quadratic Semi Assignment Problem (QSAP) form. We can alternatively extract Graver basis and generate feasible solutions systematically, in this case, Alghassi et al. (2019b) instead of solving QUBOs on a quantum annealer.

3.2 Numerical results

GAMA algorithm is used to solve the k-pathway problem using the mutation data of 33 genes for Acute Myeloid Leukemia (AML) for 200 patients (Network, 2013). By construction, the number of binary variables required is lower than available methods (?). For $k = 3$, the pathways discovered by GAMA are consistent with those reported by the TCGA authors. For $k = 6$, three additional pathways are discovered by GAMA albeit with lower coverage.

4 Embedding algorithms

If an annealer hardware is not fully connected (e.g. the D-Wave system), it is necessary to map the logical graph, Y associated with the optimization problem into the processor graph, X (Choi, 2011, Boothby et al., 2016). We describe embedding algorithms[2] based on algebraic geometry (Dridi et al., 2018a).

Definition 4. Let X be a hardware graph. A minor-embedding of the the graph, Y is a map, $\phi : Vertices(Y) -> connectedSubtrees(X)$ such that, $\forall(y_1, y_2) \in Edges(Y)$, there exists at least one edge connecting the subtrees, $\phi(y_1)$ and $\phi(y_2)$.

2 Note that X can be any graph in general, not just the hardware graph, which is the focus here.

Given an embedding of a logical graph, Y into a physical graph, X, the Y minor is a subgraph of X given by

$$\phi(Y) = \cup_{y \in Vertices(Y)} \phi(y) \tag{11}$$

This is the input graph to the quantum processor. The information regarding the mapping of each logical qubit is stored in a hash map,

$$id \times \phi : Vertices(Y) \times Vertices(Y) -> Vertices(Y) \times Subtrees(X) \tag{12}$$

which can be used to unembed the desired solution returned by the processor.

4.1 Algebraic geometry method

The set of embeddings can be viewed as an algebraic variety, which is the set of zeros of a system of polynomial equations (Cox et al., 2007). Given an embedding the mapping, $\pi : Vertices(X) -> Vertices(Y) \cup \{0\}$, where the pre-image (fiber) $\pi^{-1}(y) = \phi(y)$, $\forall y \in Vertices(Y)$ has the form:

$$\begin{aligned} \pi(x_i) = \sum_j \alpha_{ij} y_j \\ \text{with } \sum_j \alpha_{ij} = \beta_i,\ \alpha_{ij}(\alpha_{ij} - 1) = 0 \\ \alpha_{ij_1}\alpha_{ij_2} = 0,\ \text{for } j_1 \neq j_2 \end{aligned} \tag{13}$$

where $\beta_i \in \{0, 1\}$ is equal to 1, if the physical qubit x_i is used, and 0 otherwise. The conditions on the embedding ϕ in Definition 4 along with a limit on the number of usable physical qubits can be translated into a system of polynomial constraints on α_{ij} and β_i. This system defines an algebraic ideal $\mathcal{I}$, and the embeddings can be obtained using the Groebner basis of $\mathcal{I}$.

4.2 Integer programming (IP) method

An IP formulation of the embedding algorithm (Bernal et al., 2020) is developed by expressing the polynomial conditions in Dridi et al. (2018a) as linear constraints involving integer variables. This formulation includes constraints for *Minimum and Maximum size*. Embeddings are obtained by optimizing the *Embedding size* within the feasible region. A decomposition approach, iterating between a qubit assignment master problem and a fiber condition checking subproblem is also developed.

Bernal et al. (2020) tested these methods using random graphs that vary in structure, size, and density. The results are compared with the D-Wave default heuristic, `minorminer` (Cai et al., 2014). The IP-based approaches are found to be slower whenever the heuristic can find an embedding. However, it is possible to obtain infeasibility proofs and bounds on solution quality with the IP methods, but not from the heuristics. The decomposition approach outperforms the monolithic IP approach.

5 Hardware

Two fully connected Coherent Ising Machines (CIM) - the Temporal Multiplexed Coherent Ising Machine (TMCIM) (based on Böhm et al., 2019, 2021) and the Spatial Photonic Ising Machine (SPIM) (an enhancement of that of Pierangeli et al., 2019)—were built by collaborators at IIT-Madras (Prabhakar et al., 2023).

5.1 Temporal Multiplexed Ising Machine

The TMCIM was tested on the Max-Cut problem (Karp, 1972) and the results on various instances are compared with Gurobi run on an Intel Core i3 processor and also with a D-Wave machine. The graph instances for the problem are generated using **rudy** (Rendl et al., 2010). See Prabhakar et al. (2023) for details.

First, at a fixed graph size (100 nodes), and varying density, TMCIM performed better than Gurobi up to a graph density of 40%. However, above 50%, the performance of TMCIM degraded. Next, the results with a fixed graph density of 40% and varying size of the graph from 100 to 1,000 were obtained. For larger graphs, the performance of TMCIM was found to be considerably lower than Gurobi.

Second, the results are compared with the D-Wave Advantage 1.1 (DWA) annealer with the graph size varying from 20 to 100 nodes and the graph density fixed at 10%. For all the graph sizes, TMCIM is able to always give a Max-Cut value which is at least 96% of the value obtained using Gurobi (see Figure 6 in Prabhakar et al., 2023). A solution accuracy of 99% can be attained up to 30 nodes. For DWA, the solution accuracy degrades beyond 20 nodes. This can be attributed to the limited connectivity of its Pegasus graph.

5.2 Spatial Photonic Ising Machine

The SPIM was tested on a Number Partitioning Problem (NPP) with instance sizes varying from 16 to 16,384 variables. The performance of the SPIM was compared with that of Gurobi and DWA. See Tables 3, 4 in Prabhakar et al. (2023). For DWA, the number of variables that can be embedded is limited to 11×11 fully connected graph and is not competitive. For problem sizes up to 1024 variables, Gurobi performs better than SPIM. However, Gurobi is unable to find solutions as the problem size gets larger while SPIM can handle up to 16,384 variables.

6 Conclusion

We have developed GAMA (Graver Augmented Multi-seed Algorithm), a novel hybrid quantum-classical algorithm for non-linear constrained integer optimization. As an application, we have explored a new formulation for the discovery of de-novo cancer pathways. This tailored formulation is found to require fewer binary variables when compared with existing methods, and the pathways detected have been found to be consistent with previously published results.

For minor embedding that is usually required in Ising hardware that does not have an all-to-all connectivity, we have developed algebraic geometry and IP-based algorithms. The IP algorithm is found to perform well for highly structured source graphs when compared with the currently employed heuristic, `minorminer` and the Groebner basis method. While slower overall when compared with the heuristic, the algorithm can detect instance infeasibility and obtain bounds on solution quality.

We have built two photonic Ising machines, TMCIM and SPIM. We have studied their performance on Max-Cut and NPP problems, respectively, by comparing them with D-Wave and Gurobi. For the Max-Cut, TMCIM gave better results than Gurobi at smaller graph sizes ($<$ 100 nodes) and lower densities ($<$ 40%), while its accuracy is lower for larger problems. However, the performance is better than D-Wave, which can be attributed to better connectivity. SPIM can solve NPP problems up to 16384 spins, which is larger than the problem sizes solved by D-Wave and Gurobi. Gurobi's performance is better at smaller sizes, but cannot exceed more than 1024 spins.

We conclude by noting some current work in quantum annealing. We are testing GAMA[3] against state-of-the-art classical approaches for an application in disaster preparation, in collaboration with researchers at Koc University, as part of an initiative of the Turkish Ministry of Transportation and Infrastructure, focused on probable earthquakes in Istanbul. As noted earlier, the performance of the annealers depends crucially on connectivity in the hardware. We are in the process of building another fully connected annealer, based on Floquet Theory, collaborating with researchers at Cornell University and Raytheon BBN Technologies, that is implemented using superconducting circuits (Onodera et al., 2020), adding to a growing set of devices with all-to-all connectivity being developed on other technologies (such as trapped ions or cold Rydberg atoms, such as QuEra processor). Nevertheless, we expect the size of complete connectivity in any hardware in the foreseeable future to be limited. It is therefore necessary to develop additional decomposition techniques for efficiently partitioning (and then recombining) large-scale optimization problems, an area of active algorithmic research at QTG.

Author contributions

ST: Conceptualization, Supervision, Writing—review & editing. AT: Writing—original draft.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. ST and AT were supported by a DARPA Grant (through BBN Raytheon Technologies) on an AFRL Contract entitled GLIMPSE. BBN

3 A parallel article in this issue explores GAMA for Binary Classification of X-ray images via Support Vector Machine (SVM) to detect pneumonia (Guddanti et al., 2023).

Raytheon Technologies was not involved in the study design, collection, analysis, interpretation of data, the writing of this article, or the decision to submit it for publication.

Acknowledgments

Dr. Hedayat Alghassi (IBM) and Dr. Raouf Dridi (QCI) were part of QTG and co-developed content described in Sections 2–4. Dr. David Bernal (Purdue) and researchers at NASA QuAIL collaborated on research in Section 4.2. Faculty (Dr. A. Prabhakar and Dr. N. Chandrachoodan) and students (P. Shah, U. Gautham, V. Natarajan, and V. Ramesh) at IIT-Madras collaborated on the photonic devices discussed in Section 5.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Alghassi, H., Dridi, R., Robertson, A. G., and Tayur, S. (2019a). Quantum and Quantum-inspired methods for de novo discovery of altered cancer pathways. *bioRxiv*. doi: 10.1101/845719

Alghassi, H., Dridi, R., and Tayur, S. (2019b). GAMA: a novel algorithm for non-convex integer programs. *arXiv [Preprint]*. doi: 10.48550/arXiv.1907.10930

Alghassi, H., Dridi, R., and Tayur, S. (2019c). Graver bases via quantum annealing with application to non-linear integer programs. *arXiv [Preprint]*. doi: 10.48550/arXiv.1902.04215

Bernal, D. E., Booth, K. E., Dridi, R., Alghassi, H., Tayur, S., and Venturelli, D. (2020). "Integer programming techniques for minor-embedding in quantum annealers," in *Integration of Constraint Programming, Artificial Intelligence, and Operations Research: 17th International Conference, CPAIOR 2020*. Vienna, Austria: Springer, 112–129.

Böhm, F., Vaerenbergh, T. V., Verschaffelt, G., and Van der Sande, G. (2021). Order-of-magnitude differences in computational performance of analog Ising machines induced by the choice of nonlinearity. *Communications Physics* 4:149. doi: 10.1038/s42005-021-00655-8

Böhm, F., Verschaffelt, G., and Van der Sande, G. (2019). A poor man coherent Ising machine based on opto-electronic feedback systems for solving optimization problems. *Nat. Commun.* 10, 3538. doi: 10.1038/s41467-019-11484-3

Boothby, T., King, A. D., and Roy, A. (2016). Fast clique minor generation in Chimera qubit connectivity graphs. *Quant. Inform. Proc.* 15, 495–508. doi: 10.1007/s11128-015-1150-6

Cai, J., Macready, W. G., and Roy, A. (2014). A practical heuristic for finding graph minors. *arXiv [Preprint]*. doi: 10.48550/arXiv.1406.2741

Choi, V. (2011). Minor-embedding in adiabatic quantum computation: II minor-universal graph design. *Quant. Inform. Proc.* 10, 343–353. doi: 10.1007/s11128-010-0200-3

Chou, J. B., Bramhavar, S., Ghosh, S., and Herzog, W. (2019). Analog coupled oscillator based weighted Ising machine. *Sci. Rep.* 9, 5. doi: 10.1038/s41598-019-49699-5

Ciriello, G., Cerami, E., Sander, C., and Schultz, N. (2012). Mutual exclusivity analysis identifies oncogenic network modules. *Genome Res.* 22, 398–406. doi: 10.1101/gr.125567.111

Cox, D. A., Little, J., and O'Shea, D. (2007). *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Berlin, Heidelberg: Springer-Verlag.

De Loera, J., Hemmecke, R., Onn, S., Rothblum, U., and Weismantel, R. (2009). Convex integer maximization via Graver bases. *J. Pure Appl. Algeb.* 213, 1569–1577. doi: 10.1016/j.jpaa.2008.11.033

Dridi, R., Alghassi, H., and Tayur, S. (2018b). Homological description of the quantum adiabatic evolution with a view toward quantum computations. *arXiv [Preprint]. arXiv*:1811.00675.

Dridi, R., Alghassi, H., and Tayur, S. (2019a). Enhancing the efficiency of adiabatic quantum computations. *arXiv [Preprint]*. doi: 10.48550/arXiv.1903.01486

Dridi, R., Alghassi, H., and Tayur, S. (2019b). Minimizing polynomial functions on quantum computers. *arXiv [Preprint]*. doi: 10.48550/arXiv.1903.08270

Dridi, R., Alghassi, H., and Tayur, S. (2018a). A novel algebraic geometry compiling framework for adiabatic quantum computations. *arXiv [Preprint]*. doi: 10.48550/arXiv.1810.01440

Glover, F. W., Kochenberger, G. A., and Du, Y. (2018). Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models. *4OR* 17, 335–371. doi: 10.1007/s10288-019-00424-y

Graver, J. E. (1975). On the foundations of linear and integer linear programming I. *Math. Program.* 9, 207–226. doi: 10.1007/BF01681344

Guddanti, S. S., Padhye, A., Prabhakar, A., and Tayur, S. (2023). *Pneumonia Detection by Binary Classification: Classical, Quantum and Hybrid Approaches for Support Vector Machine (SVM)*. doi: 10.3389/fcomp.2023.1286657

Haber, D. A., and Settleman, J. (2007). Cancer: drivers and passengers. *Nature* 446, 145–146. doi: 10.1038/446145a

Harris, R., Sato, Y., Berkley, A. J., Reis, M., Altomare, F., Amin, M. H., et al. (2018). Phase transitions in a programmable quantum spin glass simulator. *Science* 361, 162–165. doi: 10.1126/science.aat2025

Hauke, P., Katzgraber, H. G., Lechner, W., Nishimori, H., and Oliver, W. D. (2020). Perspectives of quantum annealing: methods and implementations. *Rep. Prog. Phys.* 83, 054401. doi: 10.1088/1361-6633/ab85b8

Hemmecke, R., Onn, S., and Weismantel, R. (2011). A polynomial oracle-time algorithm for convex integer minimization. *Math.Program.* 126, 97–117. doi: 10.1007/s10107-009-0276-7

Karp, R. M. (1972). *Reducibility among Combinatorial Problems*. Boston, MA: Springer US, 85-103.

King, A. D., Carrasquilla, J., Raymond, J., Ozfidan, I., Andriyash, E., Berkley, A., et al. (2018). Observation of topological phenomena in a programmable lattice of 1,800 qubits. *Nature* 560, 456–460. doi: 10.1038/s41586-018-0410-x

Lee, J., Onn, S., Romanchuk, L., and Weismantel, R. (2010). The quadratic Graver cone, quadratic integer minimization, and extensions. *Math. Program.* 136, 301–323. doi: 10.1007/s10107-012-0605-0

Leiserson, M. D. M., Blokh, D., Sharan, R., and Raphael, B. J. (2013). Simultaneous identification of multiple driver pathways in cancer. *PLoS Comput. Biol.* 9, 5. doi: 10.1371/journal.pcbi.1003054

LLC Gurobi Optimization (2023). *Gurobi Optimizer Reference Manual*. Available online at: https://www.gurobi.com

Lucas, A. (2014). Ising formulations of many NP problems. *Front. Phys.* 2, 5. doi: 10.3389/fphy.2014.00005

McGeoch, C. C. (2014). *Adiabatic Quantum Compuutation and Quantum Annealing*. Kenfield, CA: Morgan & Claypool.

McMahon, P. L., Marandi, A., Haribara, Y., Hamerly, R., Langrock, C., Tamate, S., et al. (2016). A fully programmable 100-spin coherent Ising machine with all-to-all connections. *Science* 354, 614–617. doi: 10.1126/science.aah5178

Mohseni, N., McMahon, P. L., and Byrnes, T. (2022). Ising machines as hardware solvers of combinatorial optimization problems. *Nat. Rev. Phys.* 4, 363–379. doi: 10.1038/s42254-022-00440-8

Murota, K., Saito, H., and Weismantel, R. (2004). Optimality criterion for a class of nonlinear integer programs. *Oper. Res. Lett.* 32, 468–472. doi: 10.1016/j.orl.2003.11.007

Network, C. G. A. R. (2013). Genomic and epigenomic landscapes of adult de novo acute myeloid leukemia. *N. Engl. J. Med.* 368, 2059–2074. doi: 10.1056/NEJMoa1301689

Onodera, T., Ng, E., and McMahon, P. L. (2020). A quantum annealer with fully programmable all-to-all coupling via Floquet engineering. *NPJ Quant. Inform*. 6, 48. doi: 10.1038/s41534-020-0279-z

Pierangeli, D., Marcucci, G., Brunner, D., and Conti, C. (2020). Noise-enhanced spatial-photonic Ising machine. *Nanophotonics* 9, 4109–4116. doi: 10.1515/nanoph-2020-0119

Pierangeli, D., Marcucci, G., and Conti, C. (2019). Large-scale photonic Ising machine by spatial light modulation. *Phys. Rev. Lett*. 122, 213902. doi: 10.1103/PhysRevLett.122.213902

Pottier, L. (1996). "The Euclidean Algorithm in Dimension *n*," In *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation, ISSAC '96*. New York, NY, USA: Association for Computing Machinery, 40–42.

Prabhakar, A., Shah, P., Gautham, U., Natarajan, V., V, R., N, C., et al. (2023). Optimization with photonic wave-based annealers. *Philos. Trans. Math. Phys. Eng*. 381, 18. doi: 10.1098/rsta.2021.0409

Rendl, F., Rinaldi, G., and Wiegele, A. (2010). Solving Max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Math. Program*. 121, 307–335. doi: 10.1007/s10107-008-0235-8

Smelyanskiy, V. N., Rieffel, E. G., Knysh, S. I., Williams, C. P., Johnson, M. W., Thom, M. C., et al. (2012). A near-term quantum computing approach for hard computational problems in space exploration. *arXiv [Preprint]*. doi: 10.48550/arXiv.1204.2821

Sturmfels, B., and Thomas, R. R. (1997). Variation of cost functions in integer programming. *Math. Program*. 77, 357–387. doi: 10.1007/BF0261 4622

Tanahashi, K., Takayanagi, S., Motohashi, T., and Tanaka, S. (2019). Application of ising machines and a software development for ising machines. *J. Phys. Soc. Japan* 88, 061010. doi: 10.7566/JPSJ.88.061010

Tayur, S. R., Thomas, R. R., and Natraj, N. R. (1995). An algebraic geometry algorithm for scheduling in presence of setups and correlated demands. *Math. Program*. 69, 369–401. doi: 10.1007/BF01585566

Vandin, F., Upfal, E., and Raphael, B. J. (2012b). De novo discovery of mutated driver pathways in cancer. *Genome Res*. 22, 375–385. doi: 10.1101/gr.120477.111

Vandin, F., Clay, P., Upfal, E., and Raphael, B. J. (2012a). "Discovery of mutated subnetworks associated with clinical data in cancer," in *Pacific Symposium on Biocomputing* (Stanford University), 55–66.

Vogelstein, B., and Kinzler, K. W. (2004). Cancer genes and the pathways they control. *Nat. Med*. 10, 789–799. doi: 10.1038/nm1087

Wang, T., and Roychowdhury, J. (2019). Oim: Oscillator-based Ising machines for solving combinatorial optimisation problems. In *Unconventional Computation and Natural Computation: 18th International Conference, UCNC 2019*. Berlin, Heidelberg: Springer-Verlag, 232–256.

Wang, Z., Marandi, A., Wen, K., Byer, R. L., and Yamamoto, Y. (2013). Coherent Ising machine based on degenerate optical parametric oscillators. *Phys. Rev. A*. 88, 063853. doi: 10.1103/PhysRevA.88.063853

Weinberg, R. A., and Weinberg, R. A. (2013). *The Biology of Cancer*. New York: Garland Science.

Zhao, J., Zhang, S., Wu, L.-Y., and Zhang, X.-S. (2012). Efficient methods for identifying mutated driver pathways in cancer. *Bioinformatics (Oxford, England)* 28, 2940–2947. doi: 10.1093/bioinformatics/bts564

frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 05 June 2024
DOI 10.3389/fcomp.2024.1286057

OPEN ACCESS

EDITED BY
David Esteban Bernal Neira,
Purdue University, United States

REVIEWED BY
Luis Zuluaga,
Lehigh University, United States
Marcos César de Oliveira,
State University of Campinas, Brazil

*CORRESPONDENCE
Stéphane Louise
✉ Stephane.LOUISE@cea.fr
Kristel Michielsen
✉ k.michielsen@fz-juelich.de

†These authors share first authorship

RECEIVED 30 August 2023
ACCEPTED 14 May 2024
PUBLISHED 05 June 2024

CITATION
Vert D, Willsch M, Yenilen B, Sirdey R, Louise S and Michielsen K (2024) Benchmarking quantum annealing with maximum cardinality matching problems.
Front. Comput. Sci. 6:1286057.
doi: 10.3389/fcomp.2024.1286057

Benchmarking quantum annealing with maximum cardinality matching problems

Daniel Vert[1,2†], Madita Willsch[3,4†], Berat Yenilen[3,5], Renaud Sirdey[1,2], Stéphane Louise[1,2*] and Kristel Michielsen[3,4,5*]

[1]Université Paris-Saclay Commission for Atomic Energy (CEA), Laboratory of Integration of Systems and Technologies (LIST), Palaiseau, France, [2]AIDAS, Palaiseau, France, [3]Jülich Supercomputing Centre, Forschungszentrum Jülich, Jülich, Germany, [4]University in German Rheinisch-Westfälische Technische Hochschule Aachen (RWTH), Jülich, Germany, [5]RWTH Aachen University, Aachen, Germany

We benchmark Quantum Annealing (QA) vs. Simulated Annealing (SA) with a focus on the impact of the embedding of problems onto the different topologies of the D-Wave quantum annealers. The series of problems we study are especially designed instances of the maximum cardinality matching problem that are easy to solve classically but difficult for SA and, as found experimentally, not easy for QA either. In addition to using several D-Wave processors, we simulate the QA process by numerically solving the time-dependent Schrödinger equation. We find that the embedded problems can be significantly more difficult than the unembedded problems, and some parameters, such as the chain strength, can be very impactful for finding the optimal solution. Thus, finding a good embedding and optimal parameter values can improve the results considerably. Interestingly, we find that although SA succeeds for the unembedded problems, the SA results obtained for the embedded version scale quite poorly in comparison with what we can achieve on the D-Wave quantum annealers.

KEYWORDS

quantum annealing, simulated annealing, benchmarking, maximum cardinality matching problem, minor embedding

1 Introduction

In theory, quantum computing has the potential to yield significant, potentially even exponential, speed-up compared with the best known algorithms of traditional computing. Whether quantum computing can meet these high expectations is not yet clear. The current technological state of the art only allows for rather modest results: Even though some computing advantage has been shown on some corner cases (Arute et al., 2019; King et al., 2021), no computing advantage that is relevant for practical applications has been shown so far, and although successful application of quantum error correction has been reported (Ryan-Anderson et al., 2022; Takeda et al., 2022; Acharya et al., 2023), the threshold for fault-tolerant quantum computing has not yet been reached.

However, especially when focusing on the field of optimization, there exist quantum processors that do not rely on quantum gates but on Quantum Annealing (QA)—the natural time evolution of a "programmable" quantum system—to find a solution of an optimization problem. The Canadian company D-Wave Systems Inc. builds and commercializes such quantum processors comprising over 5,000 qubits. As these types of quantum computers act as optimizing black boxes, an important aspect from a user perspective is to characterize the quality of the outcome. Furthermore, such benchmarking

activities are of high interest for commercial and scientific prospects. There are several aspects of benchmarking:

(i) One aspect is the investigation of whether or not quantum effects are at play in the annealing process of the D-Wave processors. Studies with this aim were conducted by comparing D-Wave data with (simulated) QA and classical models often including, among others, Simulated Annealing (SA). Such studies either showed that previously considered classical models cannot reproduce the D-Wave results, for instance for freeze-out time vs. temperature (Johnson et al., 2011), for success probability distributions over many random spin glass instances (Boixo et al., 2014), and for the non-uniform probability of degenerate ground states of specially crafted problem instances where the probability of one ground state is suppressed in the quantum case and enhanced for SA (Boixo et al., 2013; Albash et al., 2015) or introduced another classical model such as noisy Boltzmann distributions (Chancellor et al., 2022), classical spin dynamics (Smolin and Smith, 2013), and spin-vector Monte Carlo (SVMC, also called SSSV model—from Shin Smith Smolin Vazirani) (Shin et al., 2014), which shows agreement with the studied D-Wave data.

(ii) A second aspect is the comparison between different generations of processors to judge possible improvements. Previous studies compared D-Wave 2000Q and Advantage system (Calaza et al., 2021; McLeod and Sasdelli, 2022; Willsch et al., 2022a), D-Wave 2000Q, Advantage system and Advantage2 prototype (Pelofske, 2023), and D-Wave Two, D-Wave 2X, D-Wave 2000Q, and Advantage system (Pokharel et al., 2021).

(iii) Another aspect is the search for quantum speedup (Rønnow et al., 2014) and investigations of the performance of quantum processors in comparison to classical algorithms. Studies were performed for academic instances such as random spin glasses (Rønnow et al., 2014), specially crafted problems with or without planted solutions (Hen et al., 2015; King et al., 2015; Albash and Lidar, 2018; Vert et al., 2020; McLeod and Sasdelli, 2022), a variety of problems with different level of difficulty (Jünger et al., 2021; McGeoch and Farre, 2023) and problems with industrial application such as the multi-car paint shop problem (Yarkoni et al., 2021), job shop scheduling problem (Carugno et al., 2022), and Earth-observation satellite mission planning problem (Stollenwerk et al., 2021). Studies benchmarking QA against classical algorithms comprise annealing-like algorithms such as SA (Rønnow et al., 2014; Hen et al., 2015; King et al., 2015; Albash and Lidar, 2018; Vert et al., 2020; Yarkoni et al., 2021; Carugno et al., 2022; McLeod and Sasdelli, 2022; Ceselli and Premoli, 2023; McGeoch and Farre, 2023), parallel tempering (McGeoch and Farre, 2023), simulated QA and SVMC (Hen et al., 2015; Albash and Lidar, 2018), and heuristic algorithms such as Tabu search (McGeoch and Wang, 2013; Yarkoni et al., 2021; Carugno et al., 2022), Hamze-de Freitas-Selby algorithm (Hen et al., 2015; King et al., 2015), or greedy algorithms (Yarkoni et al., 2021; Carugno et al., 2022; McGeoch and Farre, 2023), as well as exact solvers (McGeoch and Wang, 2013; Jünger et al., 2021; Stollenwerk et al., 2021; Ceselli and Premoli, 2023).

Benchmarking QA by comparing it to SA is a common approach since these two meta-heuristics share some similarities, SA being inspired by statistical physics and QA relying on quantum processes to achieve an optimization. In essence, SA (slowly) converges toward a Boltzmann-like distribution with a high probability of sampling low-cost solutions, whereas QA attempts to converge toward a quantum state which contains with high probability the low-energy states of an Hamiltonian. SA works well for a variety of problems, including NP-hard problems, to obtain a solution of reasonably good quality without spending too many computational resources.

In this study, we extend previous study on earlier generations of the D-Wave quantum processors in which QA was compared with SA on a specially crafted series of problems that are known to be asymptotically difficult for SA: the G_n series of the Maximum Cardinality Matching (MCM) problem. In Vert et al. (2020) and Vert et al. (2021), this problem was studied on the D-Wave 2X processor and showed how the sparse connectivity of this particular machine adds to the difficulty of solving this series of problem.

In McLeod and Sasdelli (2022), the same problem was studied, and it was pointed out that the G_n series may be exponentially difficult for the QA meta-heuristic as well, as preliminary results indicated that the spectral gap could decrease quickly in this series of problem. Indeed, the spectral gap is an important element in deciding if a given problem requires a long annealing time to reach the solution with a high probability.

In this study, we extend the study of the G_n series further:

- We added the results obtained from simulating the ideal QA process on conventional computers to assess how problem difficulty is affected by the embedding procedure even under ideal conditions.
- We provide some order of magnitude estimation to check that intrinsic precision limitations of the D-Wave processor are not the main cause of error up to the largest problem which was possible to map onto these processors.
- We compare the results including improved embeddings between several generations of the D-Wave quantum processing units (QPUs), which have different levels of connectivity, to investigate improvements between the different generations and assess the performance gain due to the higher connectivity.
- We show that the embedding is a sensitive parameter for the quality of the results obtained from the QPUs by studying the performance of SA using the same embedded instances on the D-Wave processors.

Using several approaches (ideal QA, QA on different generations of processors, and SA), we present a extensive study to demonstrate that the minor embedding required to map a problem instance onto the quantum processors can increase the difficulty of the problem significantly.

The outline of the study is as follows: First, in Section 2, we introduce the theoretical background and the applied methods relevant for the study. Second, we show and discuss the results of several experiments in Section 3 before concluding our study in Section 4.

2 Theoretical background and methods

In this section, we introduce basic concepts to understand the important ideas of this study and outline why a particular series of problems aimed at theoretically probing the worst-case complexity of SA which is relevant to test some aspects of QA. Therefore, in this section, we briefly introduce the key concepts of SA, the MCM problem, and QA with a focus on how it is implemented and what are the concrete limitations of its implementation on D-Wave quantum processors.

2.1 Simulated annealing

SA is a probabilistic meta-heuristic algorithm that is commonly used for solving optimization problems (Kirkpatrick et al., 1983). The name "annealing" comes from the annealing process in metallurgy, and the idea is strongly inspired by statistical physics.

In SA, an initial solution is randomly generated, and the algorithm gradually explores the search space by making small changes to the solution. The algorithm can even accept these changes in detrimental cases with a probability that decreases over time, i.e., with a parameter corresponding to temperature, similar to a cooling process. This allows the algorithm to escape local optima and search for better solutions.

SA has been used to solve a wide range of optimization problems, including the Traveling Salesman Problem (TSP), scheduling problems, and other NP-hard problems. It is particularly useful for problems where the search space is large and the objective function is noisy or difficult to evaluate.

One of the key advantages of SA is that it can find a good solution even if the objective function has multiple local optima. However, it can be slow to converge, and it may require a large number of iterations to find a good solution. Additionally, the performance of the algorithm can be sensitive to its parameters, such as the cooling schedule and the acceptance probability function.

As SA has been around for so long, there is no need to further introduce the general method but rather to specify the key free parameter choices for reproducibility. In our case, we have used a standard cooling schedule of the form $T_{k+1} = 0.95T_k$ starting at $T_0 = |c_0|$ (c_0 is the high cost of the initial random solution) and stopping when $T < 10^{-3}$. The key parameter of our implementation, however, is the number of iterations of the Metropolis algorithm running for each k at constant temperature which we set to n and n^2 (where n denotes the number of variables in the QUBO). For n iterations per plateau of temperature, the algorithm is very fast but the Metropolis algorithm has less iterations to reach its stationary distribution, and hence, the algorithm is expected to provide lower quality results. On the other end of the spectrum, n^2 iterations per plateau means that one can expect high-quality results, but the computation time is then much more important.

2.2 Quantum annealing principles and D-wave processors

Like SA, QA is a meta-heuristic, but instead of being inspired by statistical physics, QA lays its base on quantum physics. While first intended as a variation of SA where thermal fluctuations are replaced by quantum fluctuations (Finnila et al., 1994; Kadowaki and Nishimori, 1998), in case of a closed system, QA can also be understood by the adiabatic theorem of quantum mechanics (Farhi et al., 2000; Albash and Lidar, 2018). More accurately, a specific utilization of the adiabatic theorem applies: If a quantum system is prepared in an initial state that is the ground state of a Hamiltonian, and if the Hamiltonian is slowly and continuously changed over time, the system will remain in the instantaneous ground state of the Hamiltonian throughout the evolution. The time of the total process from the initial (or "driver") Hamiltonian to the final one is known as the *annealing time* t_A.

Let $\mathcal{H}_0$ denote the initial (driver) Hamiltonian and $\mathcal{H}_f$ the final one. For $A(s)$ and $B(s)$, respectively decreasing and increasing functions of $s = t/t_A$ with support in the interval $[0, 1]$ so that $A(0) \gg B(0)$ and $A(1) \ll B(1)$,

$$\mathcal{H}(t) = A(t/t_A)\mathcal{H}_0 + B(t/t_A)\mathcal{H}_f$$

interpolates between $\mathcal{H}_0$ and $\mathcal{H}_f$. A usual illustration is given by using a linear ramp:

$$\mathcal{H}(t) = \left(1 - \frac{t}{t_A}\right)\mathcal{H}_0 + \frac{t}{t_A}\mathcal{H}_f$$

In QA, the system is initialized in the ground state of a simple Hamiltonian $\mathcal{H}_0$, and the Hamiltonian is slowly changed to $\mathcal{H}_f$ that encodes the cost function of interest. In theory, the adiabatic theorem ensures that the system remains in the ground state throughout this process, allowing the cost function to be effectively minimized.

In the case of D-Wave quantum annealers, $\mathcal{H}_0$ and $\mathcal{H}_f$ are given as follows:

$$\mathcal{H}_0 = -\sum_{i=1}^{N} \sigma_i^x, \tag{1}$$

$$\mathcal{H}_f = \sum_{i=1}^{N} h_i \sigma_i^z + \sum_{(i,j)} J_{i,j} \sigma_i^z \sigma_j^z, \tag{2}$$

where σ_i^x and σ_i^z are the Pauli x- and z operators, N is the number of qubits, and the summation index (i, j) indicates that the sum is over pairs of qubits that are connected on the hardware. The initial quantum state $|\varphi_0\rangle = \bigotimes_i |+\rangle$ is the uniform superposition of all basis states in the z-basis or equivalently the state with all spins aligned with the x-axis. The final solution is a vector of spins $\mathbf{s}$ with $s_i \in \{-1, 1\}$ aligned with the z-axis.

The Ising Hamiltonian Equation (2) is formally and bijectively a Quadratic Unconstrained Binary Optimization (QUBO) problem formulation. Therefore, finding the energy of the fundamental state of $\mathcal{H}_f$ is equivalent to solving a QUBO problem, which is in the general case an NP-hard problem. Since NP-complete problems are polynomially reducible to one another (Garey and Johnson, 1979), many NP-hard problems such as the TSP or many logistics

and other important optimization problems can be expressed as a QUBO formulation. However, this is easier said than done as finding quadratic reformulations allowing faster resolution for some NP-hard problems is still an active domain of research in the Operations Research community (see Anthony et al., 2017 for a recent reference). A few important examples are further presented in Lucas (2014).

A QUBO problem is defined as follows:

$$\min_{\mathbf{x}\in\{0,1\}^N} (\mathcal{C}(\mathbf{x}) = \mathbf{x}^t Q \mathbf{x}),$$

where $\mathcal{C}(\mathbf{x})$ is the cost function to be minimized, $\mathbf{x}$ is a binary vector with $x_i \in \{0, 1\}$, and Q is a matrix in $\mathcal{M}_{N\times N}(\mathbb{R})$. It is worth noting that Q is often expressed as an upper triangular matrix, although it is not mandatory. Note that constrained quadratic binary problems can also be transformed to QUBO if constraints are incorporated into the cost function as soft constraints (Lucas, 2014).

2.2.1 The embedding problem on D-Wave machines

Coupling coefficients of the Ising problem can only be set to values different from zero if the physical connection between the qubits exists. The connectivity between qubits on D-Wave quantum processors is given by a sparse graph which is different for the three generations of devices that we benchmark in our present study. The architecture of the connectivity graph in the DW_2000Q Quantum Processing Unit (QPU) is called "Chimera." The average number of connections per qubit is equal to 6, and the number of qubits of DW_2000Q processors is slightly more than 2,000. The topology on the current D-Wave Advantage series is called "Pegasus," and it has an average of 15 connections per qubit. Advantage processors have more than 5,000 qubits (the exact number varies between processors). The latest generation topology, which will be available in the future Advantage2 QPU and is already available in a prototype processor, is called "Zephyr" and has an average of 20 connections per qubit. While the Advantage2 will have more than 7,000 qubits, the prototype only comprises approximately 500 qubits.

As the number of connections per qubit is low compared with the number of available qubits in any D-Wave QPU (e.g., 15 connections for over 5,000 possible destination qubits), the graph defined by the connections of a given Ising or QUBO problem is rarely isomorphic to (a subgraph of) the hardware topology. To circumvent the connectivity problem of the QPU, it is often required to combine several physical qubits to form a single logical qubit with an effectively higher connectivity to represent a given logical variable of the problem (Choi, 2008, 2010). These sets of physical qubits, also called a qubit chain, have to act like a single qubit to produce valid outputs. To achieve this, a strong ferromagnetic coupling is applied between them. Such a procedure in which multiple qubits of the QPU are used to constitute a logical variable of the target problem is called minor embedding. The D-Wave Ocean SDK (D-Wave Systems Inc, 2023a) provides a heuristic algorithm to generate a minor embedding for a given Ising (or QUBO) problem. The coupling strength of the ferromagnetic coupling between the physical qubits of a qubit chain is called chain strength, and this hyperparameter can be set by the user.

The value of the chain strength has to be chosen carefully (Choi, 2008; Raymond et al., 2020). If the chain strength is chosen too weak, low-energy solutions of the embedded problem might favor solutions where qubit chains are broken, i.e., physical qubits that represent the same logical variable have different values. In such a case, post-processing has to be applied to obtain a valid logical variable from the physical qubits, but the resulting bitstring does not necessarily have to be a low-energy solution of the original problem. If the chain strength is chosen too strong, all qubit chains act as one logical variable but their actual value might be random. This is due to the limited range and precision of the h_i and J_{ij} values. The problem has to be rescaled too much so that the h_i and J_{ij} values become too small (i.e., below the precision limit) so that all valid solutions get too close and, thus, difficult to separate, leading to random outcomes. Moreover, in principle, the chain strength should be optimized to find the sweet spot (Grant and Humble, 2022). The default procedure "uniform torque compensation" (D-Wave Systems Inc, 2023b) provided in the Ocean SDK may (see Chen et al., 2021) or may not (see Carugno et al., 2022) work well enough.

Here, we use the relative chain strength r, i.e., we scan the chain strength by setting it to $s_c = rm$ in relation to the maximum value $m = \max(\max_i |h_i|, \max_{ij} |J_{ij}|)$, occurring in the particular problem instance.

2.2.2 Searching for better embeddings

Finding an optimal embedding generally is computationally intractable in the worst-case: When both the application graph *and* the hardware topology are part of its input, the problem is NP-hard [even when the set of topologies is limited to subgraphs of the Chimera or Pegasus graphs (Lobe and Lutz, 2021)]; when the hardware topology is fixed, the problem becomes polynomial but the large constants hidden in the big-O do not lead to practical algorithms (Roberston and Seymour, 1995). Therefore, generating an optimal embedding would annihilate any possible quantum advantage a QPU would provide. Nonetheless, quickly finding a good-enough embedding is required to try and solve any given problem on a D-Wave quantum annealer. Finding a good embedding can be advantageous because it reduces the complexity of the problem to solve on the QPU and may reduce the number of utilized qubits (Gilbert and Rodriguez, 2023) while taking potentially more time on the classical computer to be generated.

From our experiments, we find that the embeddings generated by the default heuristic provided in D-Wave's Ocean library can be quite far from optimal. Nonetheless, it has the advantage of ease of use so that it does not overly complicate the procedure of solving a problem on the QPU.

We try and probe what kind of gain can be achieved by looking at better embedding results. We note that we do not aim for a highly optimized embedding here but for a reasonably good one to assess the performance improvement compared with a randomly picked embedding of potentially poor or average quality. The methodology we utilize here is to use a large number of different seeds for the Ocean's embedding procedure to generate many different embeddings (between 100 for the larger instances and up to 500 for the small ones) and select the best outcome of this process. For

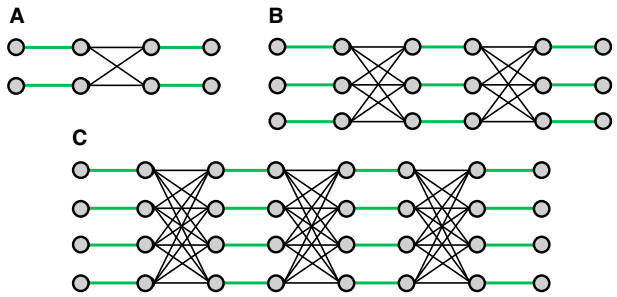


FIGURE 1
An illustration of the **(A)** G_1, **(B)** G_2, **(C)** G_3 maximum cardinality matching problems. The maximum matching is reached when all edges are selected in the sparse parts of the graph (in green).

all except the smallest problem size, we utilized the most obvious measure to estimate the quality of the embedding, namely, the number of qubits required by the embedding. Another option would be to also try to deduce the lengths of qubit chains. Trying to avoid particularly noisy parts of the QPU could be an additional consideration. For the smallest problem size, we could not apply either of the first two measures as all embeddings require eight physical qubits (on Pegasus and Zephyr topologies), which equal the number of the logical variables (i.e., no minor embedding is required). We show and discuss the results obtained from the QPU with these improved embeddings in Section 3.2.2 below. Of course, this modus operandi would not be considered for any consistent quantum advantage as such because of the amount of required pre-processing. However, used for benchmarking purposes, it can show the influence of the quality of the embedding.

2.3 The maximum cardinality matching problem

Given an undirected graph (V, E) where V is a set of vertices linked by a set of edges $E \subset V \times V$, the MCM problem is a combinatorial optimization problem aimed at finding the highest number of edges $|\tilde{E}|, \tilde{E} \subseteq E$, such that each vertex V is linked by at most one edge $e \in \tilde{E}$. While being a combinatorial problem, this problem is polynomial.

To demonstrate explicitly the slow convergence of SA in some cases, in 1988, Sasaki and Hajek (1988) devised a series of special variants of this class of problem called the G_n series. This particular series of problem instances is trivial to solve, but it is demonstrated to be exponentially hard to solve by utilizing SA.

The simplest problem of the series is the G_1 problem with only eight edges and one densely connected subgraph. Each increment in the series adds one new line of vertices and edges and one new densely connected subgraph. Therefore, G_2 adds 10 vertices and 19 edges resulting in 18 vertices and 27 edges. Only nine of these edges constitute the optimal solution. To summarize, for each instance of the G_n series, there are $2(n+1)^2$ vertices and $(n+1)^3$ edges but only $(n+1)^2$ edges in the optimal subgraph. Thus, the probability of selecting an adequate edge in the random selection operated by SA vanishes quickly. This provides a hint as to why this series of problem instances is exponentially difficult to solve for SA. Figure 1 provides an illustration of the G_1, G_2, and G_3 problems.

2.3.1 QUBO formulation

The MCM problem is a problem with constraints, and therefore, the MCM problem and the G_n series have to be transformed to a QUBO problem that can be executed on D-Wave machines. There are three steps required for this transformation. The first step is to associate relevant aspects of the original problem with binary variables, the second step is to find the cost function and constraints separately, and the third step is to add the constraints as soft constraints by incorporating them as penalties in the cost function.

The first two steps are usually closely related and also performed when the problem is solved classically with linear programming. The obvious choice for the binary variables, as we aim to maximize the number of edges in the graph, is to associate the binary variable x_e with the fact that a given edge $e \in E$ is selected for the matching ($x_e = 1$) or not ($x_e = 0$). Then, the cost function to be maximized would be $\mathcal{C}_0(\mathbf{x}) = \sum_{e \in E} x_e$, where E, as already defined, is the set of the edges of the G_n graph. The constraints are such that, for each vertex v, at most one of its associated edges is selected: $\sum_{e \in \Gamma(v)} x_e \leq 1$, where $\Gamma(v) \subseteq E$ denotes the set of edges associated to vertex v. Under the additional assumption that any maximum matching is a perfect one (which is the case for the G_n family of graphs), the penalty terms added to the cost function are then $\mathcal{C}_1 = \lambda \sum_{v \in V} \left(1 - \sum_{e \in \Gamma(v)} x_e\right)^2$ with $\lambda > 0$ and which are zero if and only if all vertices are associated with one and only one selected edge.

By constructing the total cost function $\mathcal{C}_t(\mathbf{x}) = -\mathcal{C}_0(\mathbf{x}) + \mathcal{C}_1(\mathbf{x})$ so that the cost is minimized when the solution is found, the following is obtained for the coefficients of the Q matrix:

$$q_{ee'} = \begin{cases} -1 - 2\lambda & \text{if } e = e' \\ 2\lambda & \text{if } e \neq e' \wedge \exists v \in V : (e \in \Gamma(v) \wedge e' \in \Gamma(v)) \\ 0 & \text{else}. \end{cases}$$

By choosing $\lambda = |E|$ as $|E| \geq \sum_{e \in E} x_e$, we ensure that the configuration with the minimal cost satisfies the soft constraints.

As an illustration, Figure 1 shows the graph for G_1, and the associated Q_{G_1} matrix is given by:

$$Q_{G_1} = \begin{bmatrix} -17 & 0 & 16 & 16 & 0 & 0 & 0 & 0 \\ & -17 & 0 & 0 & 16 & 16 & 0 & 0 \\ & & -17 & 16 & 16 & 0 & 16 & 0 \\ & & & -17 & 0 & 16 & 0 & 16 \\ & & & & -17 & 16 & 16 & 0 \\ & & & & & -17 & 0 & 16 \\ & & & & & & -17 & 0 \\ & & & & & & & -17 \end{bmatrix}.$$

It is easy to verify that the cost of the optimal solution is $\mathcal{C}_{G_n}(\mathbf{x}_{opt}) = -(n+1)^2 - 2(n+1)^5$, where $-2n(n+1)^5$ is an offset introduced by neglecting the constant term of $\mathcal{C}_1$ which is irrelevant to the optimization.

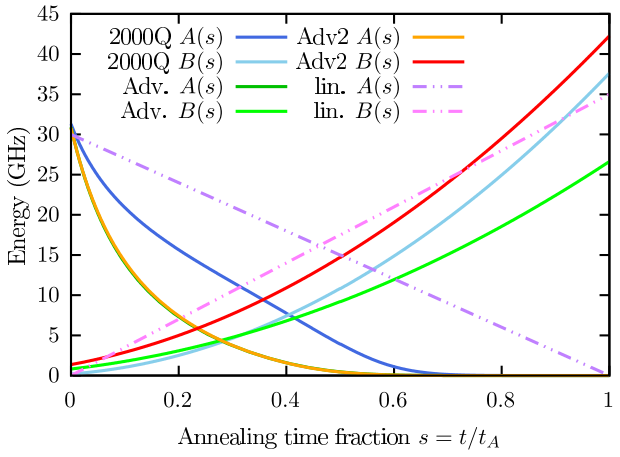


FIGURE 2
Annealing schedules of the DW_2000Q_6 (light and dark blue, dash-dotted), Advantage_system4.1 (light and dark green, solid), and Advantage2_prototype1.1 (red and orange, dashed) QPUs as well as a linear annealing schedule (pink and purple, dot-dash-dotted).

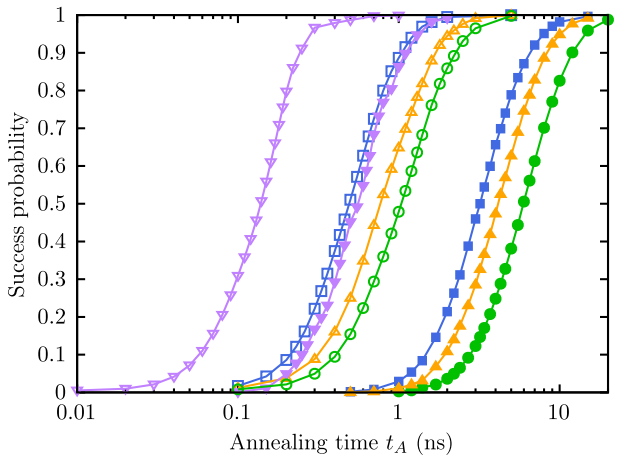


FIGURE 3
Success probability for the G_1 graph (open markers) and the G_2 graph (filled markers) as a function of annealing time for the DW_2000Q_6 (blue squares), Advantage_system4.1 (green circles), Advantage2_prototype1.1 (orange up-triangles), and linear (purple down-triangles) annealing schedules. No embedding is used for these simulation results.

2.4 Principles of ideal quantum annealing simulation

We simulate the (ideal) QA process by solving the time-dependent Schrödinger equation ($\hbar = 1$)

$$i\frac{\partial}{\partial t}|\psi(t)\rangle = \mathcal{H}(t)|\psi(t)\rangle$$

numerically, where $|\psi(t)\rangle$ denotes the state vector. For this, we use the Suzuki-Trotter product-formula algorithm (Trotter, 1959; Suzuki, 1976, 1985; De Raedt, 1987; Huyghebaert and De Raedt, 1990). This method yields a full state vector simulation of an ideal, closed system, i.e., we have direct access to the theoretical success probability, and we do not need to acquire an estimate from sampling. For the decomposition of the Hamiltonian, we use $\mathcal{H}(t) = A(t/t_A)\mathcal{H}_0 + B(t/t_A)\mathcal{H}_f$, with $A(s = t/t_A)$ and $B(s = t/t_A)$ either linear annealing schedules or the annealing functions of the particular D-Wave processor that we simulate. All annealing schedules are shown in Figure 2, and $\mathcal{H}_0$ and $\mathcal{H}_f$ are given in Equations (1, 2), respectively.

Since the memory requirement to store the full state vector grows exponentially, specifically with 2^N with N number of qubits, we have to use supercomputers with distributed memory to simulate systems with more than 30 qubits. The communication via the Message Passing Interface (MPI) follows the same scheme that is used for the gate-based quantum computer simulator JUQCS (De Raedt et al., 2007, 2019; Willsch et al., 2022b). The code utilizes OpenACC and CUDA-aware MPI to be run on GPUs. To speed up the simulation, we also run the 27-qubit cases on 4 GPUs (on one node).

The ideal QA simulation can be used to compare aspects of the D-Wave processors which are inaccessible on the real devices. For instance, we can compare the different annealing schedules of the processors without the need to apply minor embedding since in the simulation, we have all-to-all connectivity between the qubits.

3 Results and discussion

We present and discuss our results from QA simulation and QA on three generations of D-Wave processors and compare them with the results from SA. Note that most of the raw data utilized here and program to generate them are available in Benchmarking QA with MCM and problems (2023).

3.1 (Ideal) quantum annealing simulation

3.1.1 Influence of the annealing schedule

The shape of an annealing schedule influences the performance of a QA device and optimizing the annealing schedule is an active field of research (Farhi et al., 2002; Morita, 2007; Zeng et al., 2016; Brady et al., 2021; Mehta et al., 2021; Susa and Nishimori, 2021; Venuti et al., 2021; Chen et al., 2022; Hegde et al., 2022, 2023). The D-Wave annealing schedules are partly dictated by the hardware as the functions A and B cannot be chosen completely independently (Harris et al., 2010). We illustrate the influence of the annealing schedule on the difficulty of the problem exemplarily for the G_1 and G_2 graphs.

Figure 3 shows the success probability as a function of annealing time for four different annealing schedules. Three annealing schedules are taken from the DW_2000Q_6, Advantage_system_4.1, and Advantage2_prototype1.1 QPUs, the fourth is a linear one, and they are shown in Figure 2.

Since the results are obtained from (ideal) simulation of the QA process, no embedding is required regardless of the annealing schedule. Thus, differences in success probability are only due to the different annealing schedules. We find that using the linear annealing schedule requires a much shorter annealing time to reach the same success probabilities than using the D-Wave annealing

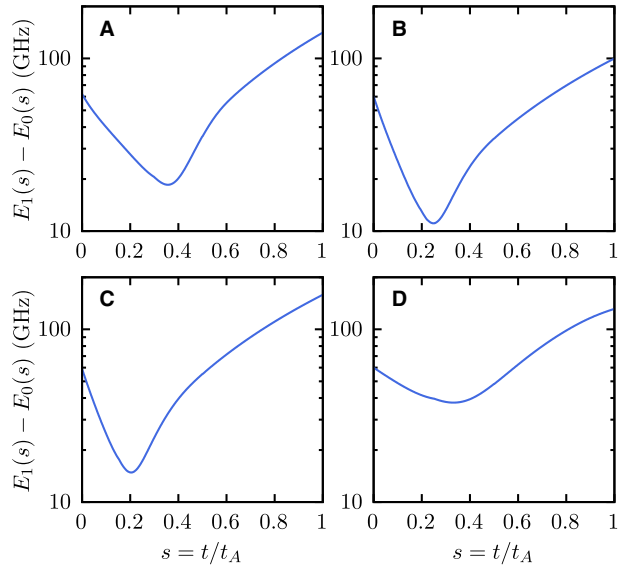


FIGURE 4
Energy $E_1(s) - E_0(s)$ between the ground state and the first excited state for the G_1 graph as a function of the anneal fraction s. **(A)** The DW_2000Q_6 annealing schedule is used. **(B)** The Advantage_system4.1 annealing schedule is used. **(C)** The Advantage2_prototype1.1 annealing schedule is used. **(D)** The linear annealing schedule is used.

schedules, which also show different performances: For the G_1 and G_2 graphs, using the DW_2000Q_6 annealing schedule performs better than using the Advantage2_prototype1.1 annealing schedule which performs better than the Advantage_system_4.1 annealing schedule. More subtle is the observation that by going from the G_1 to the G_2 problem, the required annealing time to reach a fixed success probability increases more for the DW_2000Q_6 annealing schedule than for the others.

The influence of the annealing schedule onto the difficulty of the problem can also be observed in the energy spectrum: The energy gap between the ground state and the first excited state can differ quite significantly between different annealing schedules. This is presented for the G_1 problem in Figure 4. The different panels show the energy gap during the annealing process for the DW_2000Q_6, Advantage_system4.1, Advantage2_prototype1.1, and linear annealing schedules for the same problem instance. Obviously, the energy gap between the ground state and the excited state is much smaller for the Advantage_system4.1 annealing schedule, which means the G_1 problem is then harder to solve, resulting in the longer annealing times required to achieve the same success probabilities as with the other annealing schedules. This might be different for other problems but especially for small and/or sparse problems, it was observed that DW_2000Q processors could achieve a better performance than Advantage processors (Calaza et al., 2021; Willsch et al., 2022a).

3.1.2 Influence of the embedding

To study the influence of the embedding on the performance, we consider three different embeddings for the G_1 problem. To exclude the influence of the annealing schedule, we only use the DW_2000Q_6 annealing schedule for all embeddings. We perform a scan of the relative chain strength and the annealing time.

The results are shown in Figure 5. On the left, we show the results of a scan of the relative chain strength for three different embeddings which require a different number of chains of different lengths. The embeddings onto their respective topologies are shown in Supplementary material. We normalize the problem to keep the values for h in the range $[-2, 2]$ and the values for J in the range $[-1, 1]$ as is the standard on DW_2000Q QPUs. Depending on the embedding, the maximal relative chain strength $r_{\text{max}}^{\text{scale}}$ which does not require further (compared with the unembedded problem) rescaling of the h and J parameters can be different. The minimal relative chain strength $r_{\text{min}}^{\text{sol}}$ for which the solution state is also the ground state depends on the embedding too. The relation between these two chain strengths is different for all three cases that we consider. Thus, the curves exhibit different features. For the embedding onto the Chimera topology with 4 chains of length 2, which we label by (i), $r_{\text{min}}^{\text{sol}} > r_{\text{max}}^{\text{scale}}$, the maximum success probability is reached for relative chain strengths larger than $r_{\text{min}}^{\text{sol}}$ and $r_{\text{max}}^{\text{scale}}$. For the embedding onto the modified Chimera topology with 2 chains of length 2, which we label by (ii), the solution state is always the ground state and the success probability is maximal for relative chain strengths smaller than $r_{\text{max}}^{\text{scale}}$. For the embedding onto the square grid with 2 chains of length 3, which we label by (iii) $r_{\text{max}}^{\text{scale}} > r_{\text{min}}^{\text{sol}}$, the maximum lies between $r_{\text{max}}^{\text{scale}}$ and $r_{\text{min}}^{\text{sol}}$. For embeddings (i) and (iii), there is a minimal relative chain strength one has to choose to achieve success probabilities significantly larger than zero. Due to the rescaling of the h and J parameters, one cannot choose an arbitrarily large relative chain strength as the success probability also drops for large values.

The right column in Figure 5 shows the success probability as a function of annealing time for different relative chain strengths. We consider for the success probability only the solution state without broken chains. This corresponds to discarding all samples with broken chains. Here, we include chain break fixes by majority vote (labeled by "mv" in the legend), i.e., we assign to the logical variable the value of the majority of the qubits in a chain. If the numbers in a chain is equal, we multiply the probability of this particular state by 0.5 for each such chain. For the first and second embeddings, the improvement by including chain break fixes is marginal. For embedding (iii), however, the chain break fix by majority vote can improve the performance a lot so that the success probability after majority vote gets close to the success probability without embedding. This might be surprising at first as the chain length is largest for this case (3 compared with 2 for the other 2 embeddings). However, if only one qubit is flipped in a chain of length 3, majority vote fixes it with 100%, while for the chains of length 2, the probability to fix the chain correctly is only 50%.

3.2 The G_n series on D-Wave quantum processors

The first instances of the G_n series were run on several generations of the D-Wave quantum annealer. Here, we present our results obtained with the default embeddings and the improved

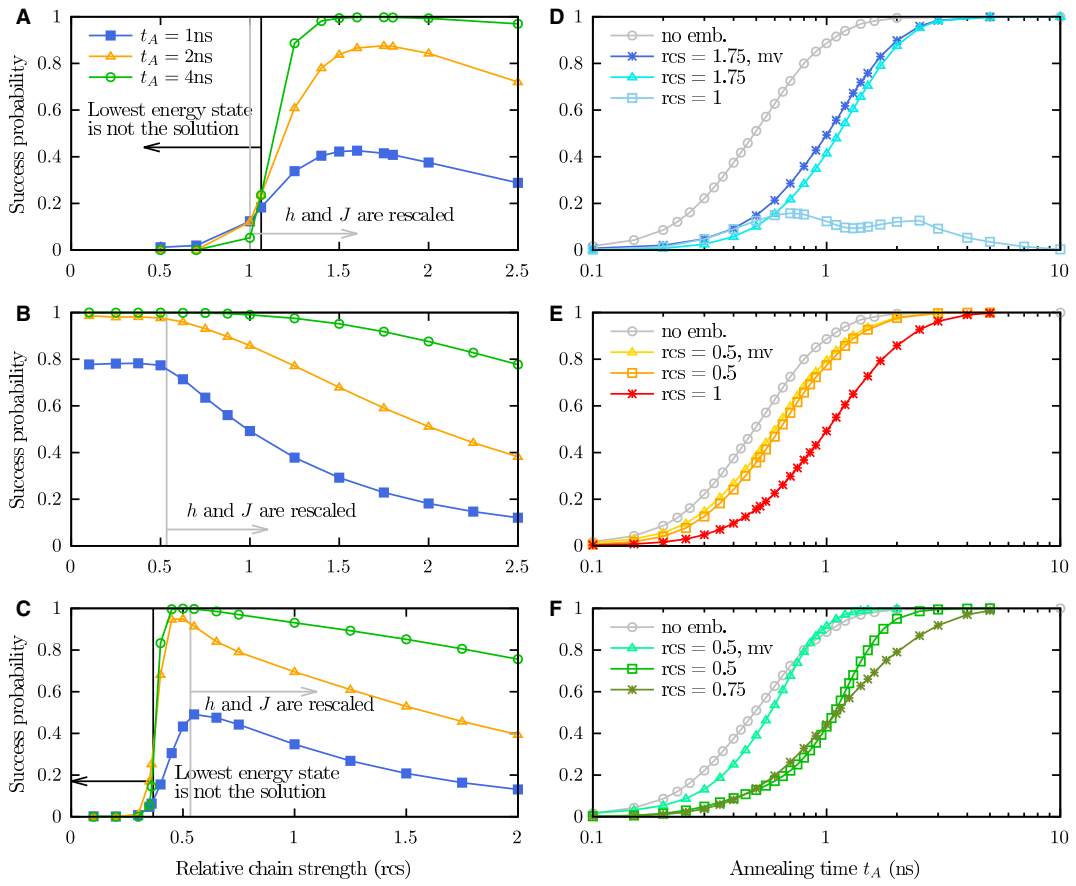


FIGURE 5
Success probability for the G_1 graph **(A–C)** as a function of the relative chain strength for different annealing times indicated in the legend and **(D–F)** as a function of annealing time for different chain strengths indicated in the legend. The DW_2000Q_6 annealing schedule is used, and the rows show different embeddings. Only the solution state is counted as success; the results with broken chains are not considered unless indicated by "mv" which means chains are fixed by majority vote. In case of the chains of length 2, the chance to fix a chain is 50%. The success probability for the unembedded case is shown in gray for comparison. **(A, D)**: Embedding onto the Chimera topology with 4 chains of length 2. **(B, E)**: Embedding onto a modified Chimera topology with additional couplers yielding 2 chains of length 2. **(C, F)**: Embedding onto a square grid with 2 chains of length 3.

ones. For reference, Table 1 shows the number of variables and the optimal energy for the graphs G_1 to G_7.

TABLE 1 Number of variables and optimal energies for G_1 up to G_7.

	# var	Opt
G_1	8	−68
G_2	27	−495
G_3	64	−2064
G_4	125	−6275
G_5	216	−15588
G_6	343	−33663
G_7	512	−65600

3.2.1 Results with default embedding procedures

D-Wave's Ocean SDK (D-Wave Systems Inc, 2023a) provides a heuristic procedure to generate minor embeddings of the problem graph onto the hardware's topology. We use 5–10 different seeds for this heuristic embedding procedure to avoid obtaining an unluckily bad embedding. Since the chain strength can have a strong influence on the success rate, we scan the relative chain strength and the annealing time as well to achieve high success rates on the different processors. The chosen relative chain strength and annealing time values are shown in the Supplementary material.

The results are shown in Table 2 for the processors Advantage_system4.1, Advantage_system5.2, Advantage_system6.1, DW_2000Q_6, and Advantage2_prototype1.1. We obtain the optimal solution up to G_3 on the Advantage processors. For DW_2000Q_6, we obtained the solution for G_1 and G_2. The required number of physical qubits is much higher on this processor due to the sparser connectivity of the Chimera topology, and since this processor has approximately 2,000 qubits, we are able to find an embedding only up to G_4 while we can embed problems until G_7 onto the Pegasus topology of the Advantage_system processors with approximately 5,000 qubits. The Advantage2_prototype1.1 with its Zephyr topology has an even higher connectivity than the Advantage_system processors, but we can only find an embedding up to G_4 due to the limited number of qubits which is approximately 500.

TABLE 2 Results obtained on various D-Wave processors for the G_n series.

	#qbs	#Opt	Best	Worst	Mean	Median	Stdev
G_1	8	1,000	−68	−68	−68	−68	0
	8	1,000	−68	−68	−68	−68	0
	8	993	−68	−52	−67	−68	1.3
	8	1000	−68	−68	−68	−68	0
	13	**998**	**−68**	**−51**	**−68**	**−68**	**0.7**
G_2	42	840	−495	−335	−486	−495	21.3
	41	951	−495	−388	−492	−495	12.5
	43	863	−495	−387	−487	−495	19.4
	37	912	−495	−388	−490	−495	15.4
	84	**849**	**−495**	**−387**	**−488**	**−495**	**19.1**
G_3	144	9	−2,064	−1,171	−1,736	−1,683	140.9
	144	15	−2,064	−1,298	−1,802	−1,808	125.8
	150	27	−2,064	−1,423	−1,773	−1,808	127.7
	121	110	−2,064	−1,551	−1,884	−1,936	104.6
	315	**0**	**−1,937**	**−1,295**	**−1,608**	**−1,554**	**98.5**
G_4	427	0	−5,527	−3,526	−4,766	−4,776	308.5
	424	0	−5,526	−3,775	−4,947	−5,024	279.2
	396	0	−5,527	−3,278	−4,514	−4,527	358.7
	322	0	−5,775	−4,030	−5,042	−5,027	227.5
	917	**0**	**−5277**	**−3774**	**−4785**	**−4775**	**232.0**
G_5	923	0	−12,994	−8,676	−11,122	−11,266	733.4
	952	0	−13,428	−9,114	−11,779	−11,702	615.9
	987	0	−12,997	−7,817	−11,396	−11,271	686.9
G_6	2,018	0	−27,487	−18,574	−23,302	−23,374	1,463.9
	1,853	0	−27,489	−19,943	−24,008	−24,058	1,309.0
	1,880	0	−27,488	−17,883	−23,136	−23,373	1,528.0
G_7	3,573	0	−51,266	−31,819	−41,678	−42,048	2,807.0
	3,562	0	−52,284	−36,921	−44,852	−45,113	2,413.0
	3,410	0	−50,239	−34,883	−43,029	−43,072	2,827.0

For each graph, rows show the results on Advantage_System4.1, Advantage_System5.2, and Advantage_System6.1 (until G_7) and Advantage2_prototype1.1 and DW_2000Q_6 (until G_4, resp. gray background and gray background with bold font). The columns show the graph, the number of physical qubits (after minor embedding), the number of optimal solutions found, the best, worst, mean and median energies, and the standard deviation of the energies obtained. The sample size is 1,000, and the used annealing times and relative chain strengths are given in the Supplementary material.

Taking into account the results on the older generation D-Wave 2X processor which can be found in the study mentioned in the reference (Vert et al., 2021), we find significant improvement from D-Wave 2X (5.1%) to D-Wave 2000Q (84.9%) for the G_2 graph. For the G_1 graph, all processors perform quite well, and there is no significant difference in success rates between the different generations. Similarly, there is not much difference in the success rates between the D-Wave 2000Q and newer generations for the G_2 graph, although the embedding onto the DW_2000Q_6 requires approximately twice as many physical qubits as embeddings onto the newer processors. For the G_3 graph, neither D-Wave 2X nor D-Wave 2000Q returned the optimal solution in our case, but the "best" energy returned by the D-Wave 2000Q is lower than the D-Wave 2X. We note that in the study mentioned in the reference (McLeod and Sasdelli, 2022), D-Wave 2000Q was able to return the optimal solution at least once. Advantage_system and Advantage2 prototype processors were able to sample the optimal solution several times, and with improved embedding (see below), Advantage2 prototype even reached a success rate of approximately 50%. For G_4, none of the tested processors returned the optimal solution, and surprisingly, D-Wave 2X and the Advantage_system processors all achieved (almost) the same "best" energy (unless an improved embedding was used on Advantage_system4.1). McLeod and Sasdelli (2022) also found the same "best" energy on the

DW_2000Q processor, and a slightly lower "best" energy on Advantage_system4.1 is very close to the one which we obtained with the Advantage2 prototype processor. For graphs G_5 and higher, only embeddings onto the Advantage_system processors were possible, and the results are compatible with each other. We note that the values of the energies and the standard deviation are stretched due to the choice of $\lambda = |E| = (n+1)^3$. Division by λ (as is also done to fit the values into the parameter ranges of the QPUs) would give more compact values and especially standard deviations, which are more comparable among the different instances. However, we decided not to divide by λ to keep the easy comparison to the previous studies in Vert et al. (2021); McLeod and Sasdelli (2022).

3.2.2 Results with improved embeddings

Table 3 shows the results of the Advantage_system4.1 QPU from G_2 to G_7 for a random embedding and the improved embedding (cf. Section 2.2.2). Table 2 shows the results of the Advantage2_prototype1.1. The annealing time was set to 500 μs, and the chain strength values are shown in Supplementary material.

The Advantage2 prototype is a preview of what will be available for the future Advantage2 full system. Its main limitation, when compared with the planned release QPU, is the number of available qubits which is only approximately 500 compared with the over 7,000 that are planned for the release-ready QPU. This limitation means we can only test the chip up to G_4.

We find that usually the solution quality is comparable; however, there are cases (for instance G_3) where the improved embedding also leads to a significant improvement in the solution quality (increased probability of finding the solution). The number of utilized qubits for an improved embedding is typically approximately 7% to 17% lower than the random one.

At first glance of Table 4, we can observe the number of qubits required for the random embedding comes close to the number of qubits required for the improved embedding of the Advantage_system4.1. As can be expected, the results become much better than what can be achieved on systems 2X and 2000Q.

The fact that the Pegasus architecture provides more connections means that any heuristic for generating a better embedding has more options and therefore the chance of finding a good embedding is much higher. This can decrease the complexity of finding a better heuristic for this particular QPU, hence greatly improving the overall quality of the solutions.

3.3 Comparison of SA, ideal QA, and QA on D-Wave processors

As the G_n series was especially crafted to probe the SA meta-heuristic, it is also a potential good candidate to probe the QA-based meta-heuristics. In this study, we did so with an ideal QA process simulated from the Schrödinger equation on a standard supercomputer, and we also completed a full set of experiments on several generations of D-Wave's quantum annealers. The latter requires to take into account several parameters that can have significant impacts on the results such as relative chain strength, annealing time, and the particular embedding.

As stated in the study mentioned in Vert et al. (2021), finding the optimal solution without the embedding for the G_n series up to G_7 is not difficult for SA with a relatively low number of iterations. Nonetheless, taking the embedding into account, D-Wave QPUs compare very favorably to SA. When using the same embedding of the Chimera topology for the cases up to G_4 also for SA, SA yields similar results as the 2X QPU only when utilizing $\alpha \mathrm{card}(E)^2$ annealing steps (α being chosen around 1,000) per plateau of temperature, which is considered a costly but usually accurate parametrization of SA. Table 5 shows the results of SA on the embeddings onto the Advantage_system4.1 QPU. As can be observed, the D-Wave results remain mostly comparable to the same high-quality SA heuristic with $\alpha \mathrm{card}(E)^2$ annealing steps per plateau of temperature. Nonetheless, it is worth noticing that obtaining these SA results with G_7 costed more than 5 days of computing time on a AMD 7,702 P, 2 GHz with 64 cores (C++ optimized SA code running in parallel with several instances per core to obtain a statistical significance). In this regard, we can conclude that even by taking the pre-processing into account, the Advantage_system4.1 QPU compares well with SA on the embedded versions of the G_n series.

Table 6 shows the same results with the embedding of the Advantage2 prototype. In this case, we can too strong observe that the Advantage2 prototype outperforms any reasonably parameterized SA metaheuristic on embedded problems both in quality and in processing time (which may also be transferable to better energy performance).

We compare the results from ideal QA obtained by simulation to the results obtained on the Advantage2_prototype1.1 processor. In Figure 6, we show the success probability obtained by ideal QA for the G_2 problem instance.

Three cases are shown: The unembedded instance with the maximum h-range on the Advantage processors ($h_{\max} = 4$) indicated by orange asterisks, the unembedded instance with maximum h-range on D-Wave 2,000Q processors ($h_{\max} = 2$) indicated by blue squares, and the embedded instance requiring 33 qubits using a relative chain strength of 0.35 with the maximum h-range of the Advantage processors indicated by red triangles. The latter two are actually quite close, suggesting that the embedding (with reasonably good chain strength, cf. Figure 7) has similar effects than a rescaling of the problem parameters h_i and J_{ij}. This is not totally surprising since a (required minimum) chain strength may force the problem parameters to be rescaled (cf. Figure 5A).

Figure 8 shows the success probability obtained by ideal QA for the embedded case in comparison to the results obtained on the Advantage2_prototype1.1 with the same embedding. Obviously, the time scales are very different. While in the ideal case approximately 20 ns is sufficient to reach a success probability of approximately 1, the shortest time possible on the D-Wave processor is 1 μs where the success rate only reaches approximately 0.9. For an annealing time of $\approx$ 35 μs, the success rate exceeds 0.98. From this observation, we conclude that for this problem instance, the D-Wave processor works in the quasistatic regime (Amin, 2015) as the ideal simulation shows that the coherent regime would have to be at approximately 1–20 ns. We note that the time to simulate the ideal QA process requires much more time (on the

TABLE 3 Results for the G_n series obtained on the Advantage_system4.1 processor for a random embedding (G_n^{rand}) and the improved embedding (G_n^{imp}).

	#qbs	#opt	best	worst	av. best	av. worst	av. mean	av. std
G_2^{rand}	44	981.3 ± 10.2	−495	−388	−495.0 ± 0.0	−434.1 ± 16.9	−494.0 ± 0.6	7.0 ± 2.0
G_2^{imp}	38	975.1 ± 6.0	−495	−387	−495.0 ± 0.0	−428.9 ± 21.6	−493.7 ± 0.3	8.4 ± 1.0
G_3^{rand}	142	61.8 ± 23.0	−2,064	−1,172	−2,064.0 ± 0.0	−1,422.8 ± 53.7	−1,825.6 ± 19.0	122.8 ± 3.6
G_3^{imp}	129	167.5 ± 27.9	−2,064	−1,298	−2,064.0 ± 0.0	−1,547.2 ± 61.8	−1,906.0 ± 10.4	101.0 ± 2.5
G_4^{rand}	438	0.0 ± 0.0	−5,776	−3,533	−5,546.7 ± 67.3	−3,992.9 ± 172.0	−4,965.1 ± 25.7	229.2 ± 6.6
G_4^{imp}	360	0.0 ± 0.0	−6,025	−3,776	−5,811.2 ± 86.2	−3,983.2 ± 169.9	−5,076.1 ± 32.4	279.3 ± 7.9
G_5^{rand}	935	0.0 ± 0.0	−13,861	−8,250	−13,257.3 ± 227.5	−9,379.8 ± 309.2	−11,668.2 ± 74.5	620.0 ± 16.9
G_5^{imp}	870	0.0 ± 0.0	−13,862	−9,113	−13,490.8 ± 149.1	−10,019.1 ± 289.1	−12,217.1 ± 53.8	555.5 ± 16.5
G_6^{rand}	1,971	0.0 ± 0.0	−28,865	−17,208	−28,191.0 ± 422.0	−19,825.0 ± 652.5	−24,574.9±146.8	13,08.6 ± 37.2
G_6^{imp}	1,801	0.0 ± 0.0	−28,865	−19,266	−28,739.1 ± 262.6	−20,949.8 ± 630.9	−25,596.2±174.1	1,209.7 ± 43.1
G_7^{rand}	3,635	0.0 ± 0.0	−55,361	−32,847	−53,907.7 ± 649.7	−38,367.4±1,661.8	−47,159.0±657.4	2397.5 ± 109.8
G_7^{imp}	3,293	0.0 ± 0.0	−55,360	−35,910	−53,478.4 ± 747.6	−38,018.6±1,145.5	−46,677.0±384.6	2366.4 ± 59.6

The columns show the graph with the corresponding embedding, the number of (physical) qubits, the average number of optimal solutions found, the best, worst, average best, average worst, and average mean energies, and the average standard deviation of the energies obtained. The sample size is 1,000, averages are over 50 repetitions and given with standard deviation. Best and worst energies are over all samples of all repetitions.

TABLE 4 Results for the G_n series obtained on the Advantage2_prototype1.1 processor for a random embedding (G_n^{rand}) and the improved embedding (G_n^{imp}).

	#qbs	#opt	best	worst	av. best	av. worst	av. mean	av. std
G_2^{rand}	37	996.0 ± 3.3	−495	−440	−495.0 ± 0.0	−442.6 ± 7.5	−494.8 ± 0.2	3.1 ± 1.3
G_2^{opt}	33	999.9 ± 0.2	−495	−442	−495.0 ± 0.0	−491.8 ± 12.6	−495.0 ± 0.0	0.1 ± 0.4
G_3^{rand}	126	278.4 ± 110.1	−2,064	−1,555	−2,064.0 ± 0.0	−1,669.4 ± 45.5	−1,952.2 ± 20.0	80.9 ± 6.9
G_3^{opt}	112	724.6 ± 103.9	−2,064	−1,554	−2,064.0 ± 0.0	−1,750.3 ± 77.2	−2,027.5 ± 14.0	58.7 ± 7.1
G_4^{rand}	333	0.0 ± 0.0	−5,777	−4,278	−5,687.1 ± 119.3	−4,474.5 ± 102.8	−5,185.6 ± 35.1	188.2 ± 11.8
G_4^{opt}	307	0.0 ± 0.0	−6,026	−4,529	−5,881.5 ± 122.8	−4,773.4 ± 92.6	−5,429.7 ± 36.4	171.4 ± 12.5

The columns show the graph with the corresponding embedding, the number of (physical) qubits, the average number of optimal solutions found, the best, worst, average best, average worst, and average mean energies, and the average standard deviation of the energies obtained. The sample size is 1,000, averages are over 50 repetitions and given with standard deviation. Best and worst energies are over all samples of all repetitions.

TABLE 5 SA results using the embedding graphs onto the Pegasus topology of the Advantage_system4.1 QPU.

	#iter	#Opt	Best	Worst	Mean	Median	Stdev
G_1	n	1,000	−68	−68	−68	−68	0
	n^2	1,000	−68	−68	−68	−68	0
G_2	n	187	−495	−334	−435	−442	38.8
	n^2	937	−495	−389	−491	−495	13.2
G_3	n	0	−1,682	−787	−1,336	−1,300	148.8
	n^2	0	−1,937	−1,299	−1,626	−1,680	107.2
G_4	n	0	−5,601	−4,026	−4,849	−4,844	225.0
	n^2	0	−6,054	−5,048	−5,585	−5,570	185.6
G_5	n	0	−13,367	−9,355	−11,395	−11,404	633.8
	n^2	0	−14,949	−13,125	−13,813	−13,775	286.9
G_6	n	0	−27,019	−18,867	−23,091	−23,131	1,370.4
	n^2	0	−33,245	−30,524	−32,112	−32,119	470.4
G_7	n	0	−37,073	−11,152	−24,083	−24,187	4,137.3
	n^2	0	−58,859	−52,187	−56,192	−56,204	999.3

The columns show the graph, the number of iterations, the number of optimal solutions found, the best, worst, mean and median energies, and the standard deviation of the energies obtained. The sample size is 1,000.

TABLE 6 SA results using the embedding graphs onto the Zephyr topology of the Advantage2_prototype1.1 QPU.

	#iter	#Opt	Best	Worst	Mean	Median	Stdev
G_1	n	1000	−68	−68	−68	−68	0
	n^2	1,000	−68	−68	−68	−68	0
G_2	n	840	−495	−389	−486	−495	19.6
	n^2	997	−495	−442	−495	−495	2.9
G_3	n	0	−1,937	−1,299	−1,628	−1,681	94.2
	n^2	118	−2,064	−1,681	−1,889	−1,937	98.5
G_4	n	0	−5,528	−3,843	−4,761	−4,778	259.5
	n^2	0	−6,026	−4,979	−5,433	−5,526	190.2

The columns show the graph, the number of iterations, the number of optimal solutions found, the best, worst, mean and median energies, and the standard deviation of the energies obtained. The sample size is 1,000.

order of minutes to hours depending on system size and simulated annealing time).

3.4 Effects of systematic errors in the QPU

When a given problem is assigned to a D-Wave QPU, there is a limited precision and several sources of potential noise. While some of the sources of errors and imprecisions are compensated for by the electronic controllers and the base software, some of these so-called ICEs (Integrated Control Errors) are intrinsic, e.g., the precision of the Digital to Analog (DAC) converters to control the couplers. As a consequence, any user of the QPU can not expect an infinite precision of the given values of the coupling. The D-Wave documentation points at an expected precision for coupling values between 0.1% and 2%.

When dividing by 2λ, the non-zero off-diagonal elements of the QUBO matrix take the value $q_{ij} = 1$ and the diagonal elements $q_{ii} = -1 - 1/2\lambda$, which is between -1 and -2 for $\lambda \geq 0.5$, and thus does not require further rescaling.

Nonetheless, the important element to consider is the associated Ising model as this is what is actually mapped to the QPU after renormalizing the values in the interval $[-1, 1]$ for J_{ij} and $[-2, 2]$ (on D-Wave 2,000Q and older systems) or $[-4, 4]$ for h_i (on Advantage and Advantage2 systems). Going from the QUBO formulation to the Ising model (where we use the convention $x_i = (1 - s_i)/2$, cf. also Section 2.2) yields non-zero $J_{ij} = \lambda/2$ (it is non-zero for all adjacent edges j of a given edge i) and $h_i \in \{(n\lambda - \lambda - 1)/2, n\lambda - 1/2\}$ for the outermost edges and all the others, respectively. The dependence on n of the h_i arises from the number of edges connected to the vertices which also grows with n. Rescaling by $2/n$ yields $h_i \in \{\lambda - (\lambda + 1)/n, 2\lambda - 1/n\}$ and $J_{ij} = \lambda/n$. As we can observe, the coupling matrix J tends toward 0 in an inverse law of n independent of the choice of λ as choosing a λ growing with n enforces a rescaling by $1/\lambda$ due to the h_i.

Nonetheless, the coupling should be an order of magnitude above the noise and biases of ICE up to $n \approx 20$ considering a 0.5% accuracy or better on the coupling. Therefore, this particular point would not constitute a large contributor to the limitation of the D-Wave QPUs in the near term future. It can thus be considered safe to assume that the limitations of D-Wave compared with ideal QA are due to the other limitations of the QPU in our experiments.

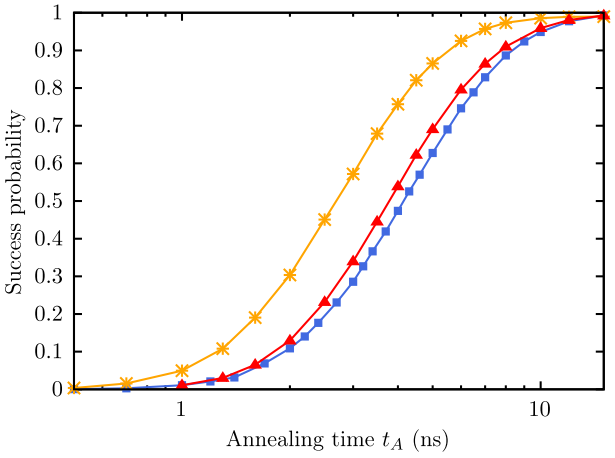


FIGURE 6
Success probability obtained from simulation for the G_2 graph as a function of annealing time for an embedding onto the Zephyr topology (rcs = 0.35, red triangles) with the Advantage2_prototype1.1 annealing schedule in comparison to direct embedding with the allowed range on Advantage processors ($-4 \leq h_i \leq 4$, orange asterisks) and direct embedding with the reduced range available on DW2000Q QPUs ($-2 \leq h_i \leq 2$, blue squares).

4 Conclusion and outlook

In summary, we have benchmarked three generations of D-Wave quantum processors by studying the performance when using different embeddings and by comparing to simulated annealing and ideal quantum annealing (i.e., solving the Schrödinger equation numerically) using a particular series (the G_n series) of maximum cardinality matching problems. From ideal quantum annealing with all-to-all connectivity, we find that for the problems without minor embedding (at least for the smaller instances), the annealing schedule of the DW_2000Q performs better, but this improvement is compensated by the decrease in

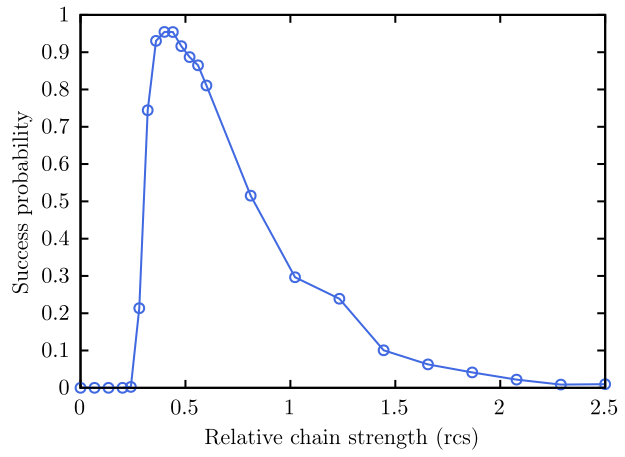


FIGURE 7
Relative chain strength scan for G_2 with a 33-qubit embedding on Advantage2_prototype1.1. with annealing time 1.0 μs and sample size 10,000.

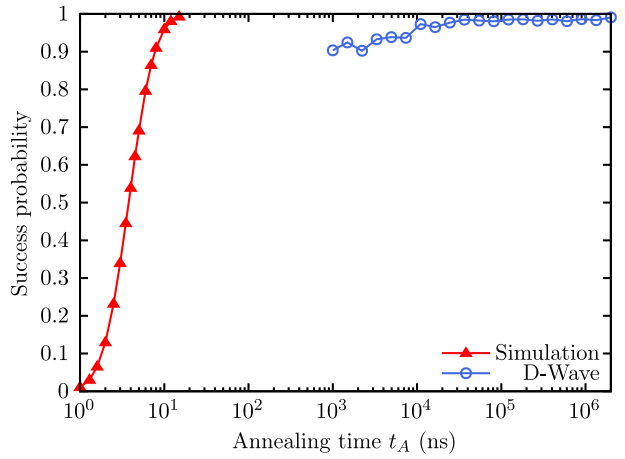


FIGURE 8
Success probability for the G_2 graph as a function of annealing time for a 33-qubit embedding onto the Zephyr topology obtained on the D-Wave quantum processor Advantage2_prototype1.1 (blue circles) and by simulation using the same annealing schedule (red triangles). The relative chain strength was set to 0.35. The sample size on the D-Wave processor is 10,000.

performance due to the required embedding onto the Chimera topology. Results for the G_1 and G_2 graphs are comparable between DW_2000Q_6 and Advantage processors, which is reasonable given the investigation using ideal QA. Although embeddings onto the Pegasus topology are possible with less than 2,000 qubits ($\approx$ the size of the DW_2000Q processors) up to G_6, we could not find an embedding onto DW_2000Q_6 for G_5. Moreover, not only the size of the processor but also the connectivity between the qubits are important to map larger problems onto the QPU's hardware graph.

For the Advantage_system4.1 and Advantage2_prototype1.1 processors, we explicitly compared the performance with different embeddings (a random one and the one with the least number of qubits found).

For most of the embeddings, we found little variation in the performance (either in the success rate if the optimal solution was found or in the minimal energy obtained). Only in a few cases, we found a significant difference suggesting that the utilized embedding might have been an unluckily bad one. Thus, trying more than a single embbedding can be useful, but trying to find an embedding with a particularly low number of qubits does not seem to be worth the effort.

The comparison to SA underlines the drop in performance when minor embedding has to be applied. While the unembedded problems are easy to solve with SA, especially the larger ones (G_3 or larger) become increasingly difficult to solve with SA if the embedding is taken into account. The performance when utilizing the embedding onto the Zephyr topology is worse than what we obtained from the Advantage2_prototype1.1 processor even for n^2 iterations per plateau of temperature. When using the embedding onto the Pegasus topology, the performance of SA is comparable (sometimes better sometimes worse) to the performance of the Advantage_system4.1 processor. However, the actual computing time required to obtain the SA results was significantly larger than what was used on the QPUs.

In conclusion, we find that the main bottleneck is the minor embedding required to solve an arbitrary problem on a QPU. The annealing schedule does have an influence but at least in the current study, it was found that it could only compensate for the larger embeddings required for DW_2000Q compared with Advantage_system in the cases of the smaller instances.

Possible future directions to continue the present study could include the benchmark of other embedding algorithms (such as Lucas, 2019; Zbinden et al., 2020) or a more in-depth investigation of annealing schedule variations such as pausing and quenching or reverse annealing in the spirit of Ikeda et al. (2019); Marshall et al. (2019); Venturelli and Kondratyev (2019); Gonzalez Izquierdo et al. (2021); Grant et al. (2021); Carugno et al. (2022). Another interesting path to include in these studies would be to consider the general QUBO formulation studied by McLeod and Sasdelli (2022) and different values for λ in the soft constraint to investigate whether all three algorithms (SA, ideal QA, and QA on the D-Wave) show the same behavior in performance change.

Data availability statement

The program code to generate the problem instances and to perform the experiments on the QPUs is publicly available in the accompanying repository (https://jugit.fz-juelich.de/qip/benchmarking-qa-with-mcm-problems). Additional information is given in the Supplementary material. Further inquiries can be directed to the corresponding authors.

Author contributions

DV: Investigation, Software, Writing – original draft. MW: Investigation, Software, Writing – original draft. BY: Investigation, Software, Writing – review & editing. RS: Writing – review & editing. SL: Software, Supervision, Writing – original draft. KM: Supervision, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. MW acknowledges support from the project Jülich UNified Infrastructure for Quantum computing (JUNIQ) that has received funding from the German Federal Ministry of Education and Research (BMBF) and the Ministry of Culture and Science of the State of North Rhine-Westphalia. BY acknowledges funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy — Cluster of Excellence Matter and Light for Quantum Computing (ML4Q) EXC 2004/1 — 390534769. The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC). The authors gratefully acknowledge the Jülich Supercomputing Centre (https://www.fzjuelich.de/ias/jsc) for funding this project by providing computing time on the D-Wave Advantage™ System JUPSI through the Jülich UNified Infrastructure for Quantum computing (JUNIQ). The work presented in this paper has been also supported by AIDAS - AI, Data Analytics and Scalable Simulation - which is a Joint Virtual Laboratory gathering the Forschungszentrum Jülich (FZJ) and the French Alternative Energies and Atomic Energy Commission (CEA). Open Access publication funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) — 491111487.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2024.1286057/full#supplementary-material

References

Acharya, R., Aleiner, I., Allen, R., Andersen, T. I., Ansmann, M., Arute, F., et al. (2023). Suppressing quantum errors by scaling a surface code logical qubit. *Nature* 614, 676–681. doi: 10.1038/s41586-022-05434-1

Albash, T., and Lidar, D. A. (2018). Demonstration of a scaling advantage for a quantum annealer over simulated annealing. *Phys. Rev. X* 8:031016. doi: 10.1103/PhysRevX.8.031016

Albash, T., Vinci, W., Mishra, A., Warburton, P. A., and Lidar, D. A. (2015). Consistency tests of classical and quantum models for a quantum annealer. *Phys. Rev. A* 91:042314. doi: 10.1103/PhysRevA.91.042314

Amin, M. H. (2015). Searching for quantum speedup in quasistatic quantum annealers. *Phys. Rev. A* 92:052323. doi: 10.1103/PhysRevA.92.052323

Anthony, M., Boros, E., Crama, Y., and Gruber, A. (2017). Quadratic reformulations of nonlinear binary optimization problems. *Mathem. Progr.* 162, 115–144. doi: 10.1007/s10107-016-1032-4

Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., et al. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*. 574, 505–510. doi: 10.1038/s41586-019-1666-5

Benchmarking QA with MCM and problems (2023). *Accompanying repository*. Available online at: https://jugit.fz-juelich.de/qip/benchmarking-qa-with-mcm-problems

Boixo, S., Albash, T., Spedalieri, F. M., Chancellor, N., and Lidar, D. A. (2013). Experimental signature of programmable quantum annealing. *Nat. Commun.* 4:2067. doi: 10.1038/ncomms3067

Boixo, S., Rønnow, T. F., Isakov, S. V., Wang, Z., Wecker, D., Lidar, D. A., et al. (2014). Evidence for quantum annealing with more than one hundred qubits. *Nat. Phys.* 10:218. doi: 10.1038/nphys2900

Brady, L. T., Baldwin, C. L., Bapat, A., Kharkov, Y., and Gorshkov, A. V. (2021). Optimal protocols in quantum annealing and quantum approximate optimization algorithm problems. *Phys. Rev. Lett.* 126:070505. doi: 10.1103/PhysRevLett.126.070505

Calaza, C. D. G., Willsch, D., and Michielsen, K. (2021). Garden optimization problems for benchmarking quantum annealers. *Quant. Inf. Proc.* 20:305. doi: 10.1007/s11128-021-03226-6

Carugno, C., Dacrema, M. F., and Cremonesi, P. (2022). Evaluating the job shop scheduling problem on a D-Wave quantum annealer. *Sci. Rep.* 12:6539. doi: 10.1038/s41598-022-10169-0

Ceselli, A., and Premoli, M. (2023). On good encodings for quantum annealer and digital optimization solvers. *Sci. Rep.* 13:5628. doi: 10.1038/s41598-023-32232-0

Chancellor, N., Crowley, P. J. D., Durić, T., Vinci, W., Amin, M. H., Green, A. G., et al. (2022). Error measurements for a quantum annealer using the one-dimensional Ising model with twisted boundaries. *NPJ Quant. Inf.* 8:73. doi: 10.1038/s41534-022-00580-w

Chen, J., Stollenwerk, T., and Chancellor, N. (2021). Performance of domain-wall encoding for quantum annealing. *IEEE Trans. Quant. Eng.* 2, 1–14. doi: 10.1109/TQE.2021.3094280

Chen, Y.-Q., Chen, Y., Lee, C.-K., Zhang, S., and Hsieh, C.-Y. (2022). Optimizing quantum annealing schedules with monte carlo tree search enhanced with neural networks. *Nat. Mach. Intell.* 4, 269–278. doi: 10.1038/s42256-022-00446-y

Choi, V. (2008). Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. *Quant. Inf. Proc.* 7, 193–209. doi: 10.1007/s11128-008-0082-9

Choi, V. (2010). Minor-embedding in adiabatic quantum computation: II. Minor-universal graph design. *Quant. Inf. Proc.*10, 343–353. doi: 10.1007/s11128-010-0200-3

De Raedt, H. (1987). Product formula algorithms for solving the time dependent Schrödinger equation. *Comp. Phys. Rep.* 7:1. doi: 10.1016/0167-7977(87)90002-5

De Raedt, H., Jin, F., Willsch, D., Willsch, M., Yoshioka, N., Ito, N., et al. (2019). Massively parallel quantum computer simulator, eleven years later. *Comput. Phys. Commun.* 237, 47–61. doi: 10.1016/j.cpc.2018.11.005

De Raedt, K., Michielsen, K., De Raedt, H., Trieu, B., Arnold, G., Richter, M., et al. (2007). Massively parallel quantum computer simulator. *Comput. Phys. Commun.* 176, 121–136. doi: 10.1016/j.cpc.2006.08.007

D-Wave Systems Inc (2023a). *Ocean SDK*. Available online at: https://github.com/dwavesystems/dwave-ocean-sdk (accessed July 17, 2023).

D-Wave Systems Inc (2023b). *Dwave-system Reference Documentation*. Available online at: https://docs.ocean.dwavesys.com/projects/system/en/latest/reference/generated/dwave.embedding.chain_strength.uniform_torque_compensation.html (accessed August 22, 2023).

Farhi, E., Goldstone, J., and Gutmann, S. (2002). Quantum adiabatic evolution algorithms with different paths. *arXiv preprint quant-ph/0208135*.

Farhi, E., Goldstone, J., Gutmann, S., and Sipser, M. (2000). Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*.

Finnila, A., Gomez, M., Sebenik, C., Stenson, C., and Doll, J. (1994). Quantum annealing: a new method for minimizing multidimensional functions. *Chem. Phys. Lett*. 219, 343–348. doi: 10.1016/0009-2614(94)00117-0

Garey, M. R., and Johnson, D. S. (1979). Computers and intractability: a guide to the theory of Np-completeness. *Siam Rev*. 24:90. doi: 10.1137/1024022

Gilbert, V., and Rodriguez, J. (2023). "Discussions about high-quality embedding on quantum annealers," in *Emerging Optimization Methods: From Metaheuristics to Quantum Approaches*.

Gonzalez Izquierdo, Z., Grabbe, S., Hadfield, S., Marshall, J., Wang, Z., and Rieffel, E. (2021). Ferromagnetically shifting the power of pausing. *Phys. Rev. Appl*. 15:044013. doi: 10.1103/PhysRevApplied.15.044013

Grant, E., and Humble, T. S. (2022). Benchmarking embedded chain breaking in quantum annealing*. *Quant. Sci. Technol*. 7:025029. doi: 10.1088/2058-9565/ac26d2

Grant, E., Humble, T. S., and Stump, B. (2021). Benchmarking quantum annealing controls with portfolio optimization. *Phys. Rev. Appl*. 15:014012. doi: 10.1103/PhysRevApplied.15.014012

Harris, R., Johnson, M. W., Lanting, T., Berkley, A. J., Johansson, J., Bunyk, P., et al. (2010). Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor. *Phys. Rev. B* 82:024511. doi: 10.1103/PhysRevB.82.024511

Hegde, P. R., Passarelli, G., Cantele, G., and Lucignano, P. (2023). Deep learning optimal quantum annealing schedules for random Ising models. *New J. Phys*. 25:073013. doi: 10.1088/1367-2630/ace547

Hegde, P. R., Passarelli, G., Scocco, A., and Lucignano, P. (2022). Genetic optimization of quantum annealing. *Phys. Rev. A* 105:012612. doi: 10.1103/PhysRevA.105.012612

Hen, I., Job, J., Albash, T., Rønnow, T. F., Troyer, M., and Lidar, D. A. (2015). Probing for quantum speedup in spin-glass problems with planted solutions. *Phys. Rev. A* 92:042325. doi: 10.1103/PhysRevA.92.042325

Huyghebaert, J., and De Raedt, H. (1990). Product formula methods for time-dependent Schrödinger problems. *J. Phys. A* 23, 5777–5793. doi: 10.1088/0305-4470/23/24/019

Ikeda, K., Nakamura, Y., and Humble, T. S. (2019). Application of quantum annealing to nurse scheduling problem. *Sci. Rep*. 9:12837. doi: 10.1038/s41598-019-49172-3

Johnson, M. W., Amin, M. H., Gildert, S., Lanting, T., Hamze, F., Dickson, N., et al. (2011). Quantum annealing with manufactured spins. *Nature* 473, 194–198. doi: 10.1038/nature10012

Jünger, M., Lobe, E., Mutzel, P., Reinelt, G., Rendl, F., Rinaldi, G., et al. (2021). Quantum annealing versus digital computing. *ACM J. Exper. Algor*. 26, 1–30. doi: 10.1145/3459606

Kadowaki, T., and Nishimori, H. (1998). Quantum annealing in the transverse ising model. *Phys. Rev. E* 58, 5355–5363. doi: 10.1103/PhysRevE.58.5355

King, A. D., Raymond, J., Lanting, T., Isakov, S. V., Mohseni, M., Poulin-Lamarre, G., et al. (2021). Scaling advantage over path-integral monte carlo in quantum simulation of geometrically frustrated magnets. *Nat. Commun*. 12:1113. doi: 10.1038/s41467-021-20901-5

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

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science* 220, 671–680. doi: 10.1126/science.220.4598.671

Lobe, E., and Lutz, A. (2021). Minor embedding in broken Chimera and Pegasus graphs is NP-complete. *arXiv preprint arXiv:2110.08325*.

Lucas, A. (2014). Ising formulations of many NP problems. *Front. Phys*. 2:74887. doi: 10.3389/fphy.2014.00005

Lucas, A. (2019). Hard combinatorial problems and minor embeddings on lattice graphs. *Quant. Inf. Proc*. 18:702. doi: 10.1007/s11128-019-2323-5

Marshall, J., Venturelli, D., Hen, I., and Rieffel, E. G. (2019). Power of pausing: advancing understanding of thermalization in experimental quantum annealers. *Phys. Rev. Appl*. 11:044083. doi: 10.1103/PhysRevApplied.11.044083

McGeoch, C. C., and Farre, P. (2023). "Milestones on the quantum utility highway," in *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)* (IEEE), 393–399. doi: 10.1109/SEC54971.2022.00058

McGeoch, C. C., and Wang, C. (2013). "Experimental evaluation of an adiabatic quantum system for combinatorial optimization," in *Proceedings of the ACM International Conference on Computing Frontiers* (ACM). doi: 10.1145/2482767.2482797

McLeod, C. R., and Sasdelli, M. (2022). "Benchmarking D-Wave quantum annealers: spectral gap scaling of maximum cardinality matching problems," in *Computational Science-ICCS 2022*, eds. D. Groen, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot (Cham: Springer International Publishing), 150–163. doi: 10.1007/978-3-031-08760-8_13

Mehta, V., Jin, F., De Raedt, H., and Michielsen, K. (2021). Quantum annealing with trigger Hamiltonians: application to 2-satisfiability and nonstoquastic problems. *Phys. Rev. A* 104:032421. doi: 10.1103/PhysRevA.104.032421

Morita, S. (2007). Faster annealing schedules for quantum annealing. *J. Phys. Soc. Japan* 76:104001. doi: 10.1143/JPSJ.76.104001

Pelofske, E. (2023). Comparing three generations of D-Wave quantum annealers for minor embedded combinatorial optimization problems. *arXiv preprint arXiv:2301.03009*.

Pokharel, B., Izquierdo, Z. G., Lott, P. A., Strbac, E., Osiewalski, K., Papathanasiou, E., et al. (2021). Inter-generational comparison of quantum annealers in solving hard scheduling problems. *Quant. Inf. Proc*. 22:364. doi: 10.1007/s11128-023-04077-z

Raymond, J., Ndiaye, N., Rayaprolu, G., and King, A. D. (2020). "Improving performance of logical qubits by parameter tuning and topology compensation," in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 295–305. doi: 10.1109/QCE49297.2020.00044

Roberston, N., and Seymour, P. D. (1995). Graph minors. xiii. The disjoint paths problem. *J. Combin. Theory Series B* 63, 65–110. doi: 10.1006/jctb.1995.1006

Rønnow, T. F., Wang, Z., Job, J., Boixo, S., Isakov, S. V., Wecker, D., et al. (2014). Defining and detecting quantum speedup. *Science* 345, 420–424. doi: 10.1126/science.1252319

Ryan-Anderson, C., Brown, N. C., Allman, M. S., Arkin, B., Asa-Attuah, G., Baldwin, C., et al. (2022). Implementing fault-tolerant entangling gates on the five-qubit code and the color code. *arXiv preprint arXiv:2208.01863*.

Sasaki, G. H., and Hajek, B. (1988). The time complexity of maximum matching by simulated annealing. *J. ACM* 35, 387–403. doi: 10.1145/42282.46160

Shin, S. W., Smith, G., Smolin, J. A., and Vazirani, U. (2014). How "quantum" is the D-Wave machine? *arXiv preprint arXiv:1401.7087*.

Smolin, J. A., and Smith, G. (2013). Classical signature of quantum annealing. *Front. Phys*. 2:52. doi: 10.3389/fphy.2014.00052

Stollenwerk, T., Michaud, V., Lobe, E., Picard, M., Basermann, A., and Botter, T. (2021). Agile earth observation satellite scheduling with a quantum annealer. *IEEE Trans. Aerosp. Electr. Syst*. 57, 3520–3528. doi: 10.1109/TAES.2021.3088490

Susa, Y., and Nishimori, H. (2021). Variational optimization of the quantum annealing schedule for the Lechner-Hauke-Zoller scheme. *Phys. Rev. A* 103:022619. doi: 10.1103/PhysRevA.103.022619

Suzuki, M. (1976). Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. *Commun. Math. Phys*. 51, 83–190. doi: 10.1007/BF01609348

Suzuki, M. (1985). Decomposition formulas of exponential operators and Lie exponentials with some applications to quantum mechanics and statistical physics. *J. Math. Phys*. 26, 601–612. doi: 10.1063/1.526596

Takeda, K., Noiri, A., Nakajima, T., Kobayashi, T., and Tarucha, S. (2022). Quantum error correction with silicon spin qubits. *Nature* 608, 682–686. doi: 10.1038/s41586-022-04986-6

Trotter, H. F. (1959). On the product of semi-groups of operators. *Proc. Amer. Math. Soc*. 10, 545–551. doi: 10.1090/S0002-9939-1959-0108732-6

Venturelli, D., and Kondratyev, A. (2019). Reverse quantum annealing approach to portfolio optimization problems. *Quant. Mach. Intell*. 1, 17–30. doi: 10.1007/s42484-019-00001-w

Venuti, L. C., D'Alessandro, D., and Lidar, D. A. (2021). Optimal control for quantum optimization of closed and open systems. *Phys. Rev. Appl*. 16:054023. doi: 10.1103/PhysRevApplied.16.054023

Vert, D., Sirdey, R., and Louise, S. (2020). "Revisiting old combinatorial beasts in the quantum age: quantum annealing versus maximal matching," in *Computational Science-ICCS 2020*, eds. V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, et al. (Cham: Springer International Publishing), 473–487. doi: 10.1007/978-3-030-50433-5_37

Vert, D., Sirdey, R., and Louise, S. (2021). Benchmarking quantum annealing against "hard" instances of the bipartite matching problem. *SN Comput. Sci*. 2:106. doi: 10.1007/s42979-021-00483-1

Willsch, D., Willsch, M., Gonzalez Calaza, C. D., Jin, F., De Raedt, H., Svensson, M., et al. (2022a). Benchmarking Advantage and D-Wave 2000Q quantum annealers with exact cover problems. *Quant. Inf. Proc*. 21:141. doi: 10.1007/s11128-022-03476-y

Willsch, D., Willsch, M., Jin, F., Michielsen, K., and De Raedt, H. (2022b). GPU-accelerated simulations of quantum annealing and the quantum approximate optimization algorithm. *Comput. Phys. Commun*. 278:108411. doi: 10.1016/j.cpc.2022.108411

Yarkoni, S., Alekseyenko, A., Streif, M., Von Dollen, D., Neukart, F., and Bäck, T. (2021). "Multi-car paint shop optimization with quantum annealing," in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 35–41. doi: 10.1109/QCE52317.2021.00019

Zbinden, S., Bärtschi, A., Djidjev, H., and Eidenbenz, S. (2020). "Embedding algorithms for quantum annealers with Chimera and Pegasus

connection topologies," in *Lecture Notes in Computer Science* (Cham: Springer International Publishing), 187–206. doi: 10.1007/978-3-030-50743-5_10

Zeng, L., Zhang, J., and Sarovar, M. (2016). Schedule path optimization for adiabatic quantum computing and optimization. *J. Phys. A* 49:165305. doi: 10.1088/1751-8113/49/16/165305

frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 10 June 2024
DOI 10.3389/fcomp.2024.1356983

OPEN ACCESS

EDITED BY
Nicholas Chancellor,
Durham University, United Kingdom

REVIEWED BY
Teng Bian,
Facebook, United States
Pascal Halffmann,
Fraunhofer Institute for Industrial Mathematics (FHG), Germany

*CORRESPONDENCE
Arthur Witt
✉ arthur.witt@ieee.org
Jangho Kim
✉ j.kim@fz-juelich.de

RECEIVED 16 December 2023
ACCEPTED 21 May 2024
PUBLISHED 10 June 2024

CITATION
Witt A, Kim J, Körber C and Luu T (2024) ILP-based resource optimization realized by quantum annealing for optical wide-area communication networks—A framework for solving combinatorial problems of a real-world application by quantum annealing. *Front. Comput. Sci.* 6:1356983. doi: 10.3389/fcomp.2024.1356983

ILP-based resource optimization realized by quantum annealing for optical wide-area communication networks—A framework for solving combinatorial problems of a real-world application by quantum annealing

Arthur Witt [1]*, Jangho Kim [2]*, Christopher Körber [3] and Thomas Luu [2]

[1]Institute of Communication Networks and Computer Engineering, University of Stuttgart, Stuttgart, Germany, [2]Institute for Advanced Simulation (IAS-4) & Jülich Aachen Research Alliance High Performance Computing, Forschungszentrum Jülich, Jülich, Germany, [3]Fraunhofer Research Institution for Energy Infrastructures and Geothermal Systems IEG, Fraunhofer IEG, Bochum, Germany

Resource allocation of wide-area internet networks is inherently a combinatorial optimization problem that if solved quickly, could provide near real-time adaptive control of internet-protocol traffic ensuring increased network efficacy and robustness, while minimizing energy requirements coming from power-hungry transceivers. In recent works we demonstrated how such a problem could be cast as a quadratic unconstrained binary optimization (QUBO) problem that can be embedded onto the D-Wave Advantage™ quantum annealer system, demonstrating proof of principle. Our initial studies left open the possibility for improvement of D-Wave solutions via judicious choices of system run parameters. Here we report on our investigations for optimizing these system parameters, and how we incorporate machine learning (ML) techniques to further improve on the quality of solutions. In particular, we use the Hamming distance to investigate correlations between various system-run parameters and solution vectors. We then apply a decision tree neural network (NN) to learn these correlations, with the goal of using the neural network to provide further guesses to solution vectors. We successfully implement this NN in a simple integer linear programming (ILP) example, demonstrating how the NN can fully map out the solution space that was not captured by D-Wave. We find, however, for the 3-node network problem the NN is not able to enhance the quality of space of solutions.

KEYWORDS

discrete optimization, integer linear program, machine learning, quantum annealing, quantum computing, resource allocation, wide-area networks

1 Introduction

Quantum computing is a cutting-edge technology that has gained significant relevance during the last decades. Algorithms for searching and optimization are currently studied intensively on quantum computers as they hold the potential for solving problems with non-polynomial (NP) complexity very efficiently. Nowadays, quantum computers have reached a scale that allows for the solution of non-trivial problems which have real-world applications.

One example is the energy-aware resource allocation of wide-area networks (Chiaraviglio et al., 2009). In these cases, one can consider the resource allocation as an optimization problem and introduce it as a relevant application of quantum computing, and in particular quantum annealing (QA), as it inherently provides a certain failure tolerance with self-healing capability. Further, the problem has NP complexity and requires frequent solutions for just-in-time adaptation of the network. If solutions are generated quickly, say on the order of seconds, a revolution in network operation with increased network efficiency might be possible since current solutions obtained from classical and/or heuristic algorithms require 15 minutes or more for time-to-solution as shown in Feller (2012) and Tornatore et al. (2002). In previous studies, (Witt et al., 2023), we have demonstrated how this resource allocation problem can be formulated, based on an integer linear program (ILP) model, as a quadratic unconstrained binary problem (QUBO) which can then be embedded onto a quantum annealer.

In our initial studies we used the D-Wave Advantage™ system (`JUPSI`) at the Forschungszentrum Jülich to perform the quantum annealing. As part of the solution process, the QUBO problem was embedded onto the quantum qubits prior to performing the quantum annealing. This entailed mapping the problem onto a network of logical qubits, whereby each logical qubit consists of a constellation, or "chain", of physical qubits. This mapping ensures the requisite "connectivity" of the logical qubits as dictated by the QUBO problem. We discovered that the network optimizing approach was greatly limited by this embedding process. For example, the optimization problem of a network with three nodes can be described as a QUBO with ∼100 binary variables (logical qubits). Even though this 3-node problem is quite small, the required amount of physical qubits was in the range of 500 qubits, representing already roughly 10% of the physical qubits available in the D-Wave Advantage™ system. With the current embedding process that we employed at the time, a simulation of a 15-node problem, corresponding to a real-world network, would require a quantum annealer with ∼50,000 physical qubits, which is an order of magnitude larger then current systems. We note that the embedding process is not unique.

Our initial studies also had limited scope in system run parameters, such as annealing time and profile of the annealing process, penalty factor of the QUBO matrix, and chain strength between physical qubits constituting logical qubits. Our choice of run parameters were constrained mainly to system default values, with little exploration on the dependence of quality of solutions on these run parameters. Therefore there is potential room for increasing the efficiency of the quantum annealing process (which would result in better quality and more *feasible* solutions) by judicious choice of optimized run parameters.

In this paper we address some of these issues by studying the process of annealing with the aim to optimize the parameters for the quantum annealing procedure. We introduce solution quality metrics for evaluation purposes. Of particular import is the Hamming distance metric, which rates the distance between the ideal and obtained solution vector in binary space. By using D-Wave solutions in conjunction with the Hamming distance to optimal solution, we empirically determine correlations between various run parameters and the quality of solution. These correlations guide us in determining optimized run parameters for the system in question, with the hope that the same optimized run parameters can be applied to similar, but larger, systems. Furthermore, we apply a decision tree neural network (NN) to learn these correlations, after which we use the NN to "guess" improved solutions. This NN represents a machine learning (ML) approach that we couple with D-Wave generated solutions that aims at providing better quality solutions, and represents an example of a hybrid classical (ML)/quantum (QA) procedure for solving the combinatorial optimization problem.

Our paper is organized as follows. In the following section we give a cursory description of ILPs in general, the used method to solve ILPs on quantum annealer, and two ILPs that we examined in our study. We then introduce in Subsection 2.4 a Hamming distance metric, and demonstrate how it is used to derive correlations between quality of solutions and various system run parameters. Such correlations will be "learned" by our decision tree NN, which we describe in detail in Subsection 2.5. In Section 3 we present our findings. We first concentrate on a simple ILP problem, demonstrating that our hybrid classical/quantum procedure does indeed result in new feasible solutions while at the same time providing guidance on optimized run parameters. We then apply the formalism to the 3-node network problem mentioned above, where here we see limited improvement in solutions, all of which unfortunately are nowhere near the optimal solution. In Section 4 we discuss our findings and recapitulate. We comment on possible future directions of investigation.

2 Materials and methods

2.1 The concept of integer linear programs

The investigations performed in our work fall under the class of discrete optimization problems, meaning variables $\boldsymbol{x}$ to be optimized take on only discrete values. Such problems can be cast succinctly as an integer linear program (ILP), where certain constraints, given as a set of linear (in-)equation, have to be satisfied while minimizing a linear function. An ILP can be defined in its canonical form by (Equations 1–4):

$$\text{objective} \quad \boldsymbol{x}_0 = \underset{\boldsymbol{x}}{\operatorname{argmin}}\{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}\} \tag{1}$$

$$\text{constraints} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \leq \boldsymbol{0} \tag{2}$$

$$\text{variables} \quad \boldsymbol{x} \in \mathbb{N}^n,\ x_i \geq 0 \tag{3}$$

$$\text{constants} \quad \boldsymbol{c} \in \mathbb{R}^n, \boldsymbol{b} \in \mathbb{R}^m, \boldsymbol{A} \in \mathbb{R}^{m\times n}\,. \tag{4}$$

The ILP's objective function can be seen as a loss function and is defined in Equation (1) with a vector of cost terms $\boldsymbol{c}$ weighting

the variable vector $\boldsymbol{x}$. Matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$ parameterize the linear equations that represent the inequality constraints (Equation 2). They can be reshaped to equality constraints, $\boldsymbol{Ax} + \boldsymbol{b} + \boldsymbol{s} = \boldsymbol{0}$, by introducing *slack* variables $\boldsymbol{s} \in \mathbb{R}, s_j \geq 0$. This is a typical step within the classical ILP-solving algorithm *simplex*, see Nash (2000). We use the convention, that $\mathbb{R}$ are real-valued numbers, $\mathbb{N}$ natural numbers inclusive zero, and $\mathbb{B}$ binary numbers.

Such problems are well-known to be non-polynomial (NP)-hard in general. According to Karp (1972) and Adler et al. (2014), linear programs are a rare class of problems in NP that resists the classification as NP-complete or polynomial-solvable problems. Lenstra (1983) argued, that mixed-integer linear programs with fixed number of variables are solvable in polynomial-time. In contrary, Nguyen and Pak (2017) present integer programs that are NP-complete, even for fixed number of variables. Their work further shows, that some integer programs are solvable in polynomial-time. We can conclude, that bounded integer linear programs are NP-complete and are solvable within polynomial time in few cases.

2.2 Solving integer linear programs on quantum annealer

Quantum annealers are well-suited for investigating ILP problems. However, an additional modification to the ILP problem is required prior to embedding the problem on the quantum qubits. Here the constraints are included into the cost function (to be minimized) by introduction of penalty weight p. In so doing, the original ILP problem with constraints is recast into quadratic form without constraints,

$$\left.\begin{aligned} \boldsymbol{x}_0 &= \underset{\boldsymbol{x}}{\operatorname{argmin}} \left\{\boldsymbol{c}^\top \boldsymbol{x}\right\} \\ \boldsymbol{Ax} + \boldsymbol{b} &\leq \boldsymbol{0} \\ \boldsymbol{x} &\in \mathbb{N}^n \geq 0 \end{aligned}\right\} \longleftrightarrow \left\{\begin{aligned} \boldsymbol{q}_0 &= \underset{\boldsymbol{q}}{\operatorname{argmin}} \left\{\boldsymbol{q}^\top \boldsymbol{Q} \boldsymbol{q} + C\right\} \\ \boldsymbol{q} &\in \{0,1\}^k \end{aligned}\right. .$$

Here $\boldsymbol{A}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ are problem specific parameters as introduced in the previous section. It is useful to classify solution vectors $\boldsymbol{x}$ into two categories: **feasible** solutions which fulfill the constraints and **unfeasible** solutions which violate the constraints. While a feasible solution to the ILP is not necessarily an optimal solution, an unfeasible solution hypothetically can have a smaller objective value than the optimal feasible solution.

The problem is mapped to the quadratic unconstrained binary optimization (QUBO) by definition of matrix $\boldsymbol{Q}$ that includes a penalty factor p and a constant C. The inequality can be expressed by an equation and another minimization over a slack variable $\boldsymbol{s}$ incorporated into the bit vector $\boldsymbol{q}$. The QUBO objective function minimizes both the ILP objective function plus another objective function representing the constraints. The penalty term expresses the relative weight between both (ILP objective and constraint) objective functions, see Chang et al. (2020) and Witt et al. (2023) for a more detailed description. Finding the solution set $\boldsymbol{q}_0$ that provides the absolute minimum of $\boldsymbol{q}^\top \boldsymbol{Q} \boldsymbol{q}$ is equivalent to solving the original ILP problem with solution vector $\boldsymbol{x}_0$.

The D-Wave Advantage™ system is adapted to solving the Ising spin system that represents an array of binary spins with interactions between spins $\boldsymbol{\sigma}$ giving by some connectivity matrix $\boldsymbol{J}$ and external magnetic field $\boldsymbol{h}$. Our QUBO matrix can easily be rewritten using $\boldsymbol{J}$ and $\boldsymbol{h}$ without any loss of generality,

$$\begin{aligned} &\boldsymbol{q}^\top \boldsymbol{Q} \boldsymbol{q} + C \Leftrightarrow \boldsymbol{\sigma}^\top \boldsymbol{J} \boldsymbol{\sigma} + \boldsymbol{h}^\top \boldsymbol{\sigma} + g \\ &\text{with} \quad \begin{cases} \boldsymbol{J} &= \frac{1}{4} \boldsymbol{Q}_0 \\ \boldsymbol{h} &= \frac{1}{2} \hat{\boldsymbol{q}} + \frac{1}{2} \boldsymbol{Q}_0 \boldsymbol{1} \\ g &= \frac{1}{4} \boldsymbol{1}^\top \boldsymbol{Q}_0 \boldsymbol{1} + \frac{1}{2} \boldsymbol{1}^\top \hat{\boldsymbol{q}} + C \end{cases} , \end{aligned}$$

with $\boldsymbol{Q}_0 = \boldsymbol{Q} - \operatorname{diag}\{\hat{\boldsymbol{q}}\}$, $\hat{\boldsymbol{q}} = \operatorname{diag}^{-1}\{\boldsymbol{Q}\}$, and g some constant. The problem is now well-suited for the D-Wave machine. In Chang et al. (2020) and Witt et al. (2023), we demonstrated proof of principle that such a problem can be solved on a quantum annealer.

2.3 Investigated integer linear programs

In this work we have investigated multiple ILPs, two of which we define explicitly here. The details of the remaining ILPs we considered are described in our accompanying Supplementary material. The first ILP optimizes the selection of two integer variables under some constraints. It provides a test case where all possible solutions can be studied with the approach of brute force sampling, i.e., it provides a well-suited setup for benchmarking. The second ILP describes a realistic network resource optimization as studied in Witt et al. (2023). As possible solutions are representable as binary vectors with more than 60 variables, a brute force sampling is not applicable within reasonable time for this case.

2.3.1 Trivial ILP

Based on expressions (1)–(4), we can define a particular ILP problem by

$$\boldsymbol{A} = \begin{bmatrix} -1/3 & -1 \\ -3 & -1 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 2 \\ 6 \\ -2 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 1 \\ 3 \end{bmatrix},$$
$$\boldsymbol{x} \in \mathbb{N}^2, \quad \boldsymbol{s} \in \mathbb{N}^3 .$$

A graphical interpretation of this ILP is depicted in Figure 1. We can easily obtain the optimal solution vectors,

$$\boldsymbol{x}_0 = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \text{ or } \begin{bmatrix} 6 \\ 0 \end{bmatrix},$$

and the optimal cost value $\boldsymbol{c}^\mathrm{T} \boldsymbol{x}_0 = 6$ from this graph. This problem is an explicit example of an ILP that can have more than one optimal solution, which in turn can cause misleading results in benchmarking experiments. In general, ILPs can have zero, one or more solutions. We restrict the values in $\boldsymbol{x}$ to $x_i \in \{0,1,2,3\}, \forall i \in \{1,2\}$, such that the ILP is uniquely solvable.

For binary representation of integer values, we will use 2 bits for each variable in $\boldsymbol{x}$ and 3 bits for each element in $\boldsymbol{s}$. Then $\boldsymbol{q}$ is the binary search vector to be optimized. According to Chang

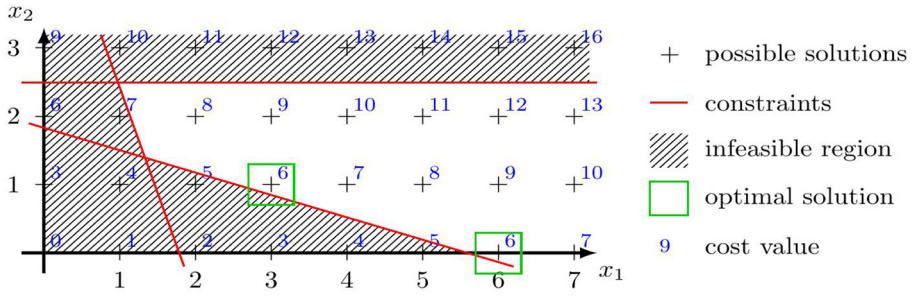


FIGURE 1
Graphical interpretation of the trivial ILP problem. The drawn constraint lines are slightly shifted for visualization purposes without falsifying the feasible region of integer values.

et al. (2020), this mapping with integer mapping matrix $\boldsymbol{Z}$ can be described by

$$\begin{bmatrix} \boldsymbol{x} \\ -- \\ \boldsymbol{s} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ -- \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \underbrace{\left[\begin{array}{cccc|cccccccc} 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 2 & 1 \end{array}\right]}_{\boldsymbol{Z}} \underbrace{\begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{13} \end{bmatrix}}_{\boldsymbol{q}}.$$

In Supplementary Table S1 we enumerate other trivial ILPs that we have investigated. These ILPs encompass a range of optimal solutions, parameters, and dimensions.

2.3.2 Network resource allocation problem

Optical wide-area networks consist of nodes that are linked by optical fiber systems in form of a meshed topology. Nodes $v \in V$ are two layered. They are equipped with electrical IP routers in the upper layer and optical cross connects (OXCs) in the lower layer. Traffic from connected networks that traverses the wide area network is "handed over" at the IP layer. Signal transitions between layers inside the WAN are performed with optical bidirectional transceivers, that are configured for unidirectional use as required. Optical transceivers generate optical signals with a bandwidth of 50 GHz at various center frequencies. A finite number of signals can be combined in a dense wavelength division multiplexing (DWDM) scheme on a particular optical fiber link. This schemes are specified according to ITU-T (2020). Thus, usable frequency bands in the optical region, typically referred as wavelengths, are uniquely defined. Optical cross connects enable wavelength-selective forwarding and redirection of optical signals between connected fibers. Fiber links are realized by a sequence of fiber spans and fiber amplifiers and provide a hardware-wise connection between nodes according to the networks topology. The maximal reach of optical signals depends on the signal configuration (specified by modulation schemes, and used forward error correction, etc.) and the transceiver type itself. As an example, a tunable coherent transceiver[1] achieves an optical reach of 1,000 km at a rate of 100 GBit/s. Typically, optical transmission paths are organized as a sequence of transmission sections c with at least one section to enable a end-to-end data transfer. Transmission sections are abstract links in the lower layer that provide optical transparent transmission on multiple wavelength. Their spanning distance is limited by the optical reach of the driving transceivers. Figure 2 illustrates how transmission paths in wide-area networks can be realized.

Energy-aware traffic engineering can be seen as a major task for economic network operation. Therefore, network resources like transceivers and wavelengths on fiber links have to be allocated to assign the required capacity to a transmission section c. Assuming that the network is operated as single rate system, i.e., all transceivers have the same signal rate, e.g., $\xi = 100$ GBit/s, capacities at a transmission section c can be scaled if multiple transceivers, enumerated by w_c, are activated. Thus, the transmission section's capacity is $w_c\xi$. A unidirectional traffic demand d represents a connectivity request between two network nodes. We assume, that a demand exist for all disjunct node pairs (u, v) with $u \neq v$ and $u, v \in V$. The network has to provide appropriate transmission paths, i.e., routes through the network topology along a sequence of transmission sections, to enable the transport of the demand's traffic with volume h_d. We prepare a network-specific collection T of possible transmission path realization prior the optimization, whereas possible transmission path realizations per demand d are defined as $t_d \in T_d \subset T$, see Witt et al. (2023), Section II-C. The economic resource allocation within WDM networks is a discrete and combinatorial optimization problem.

Witt et al. (2023) devised an integer linear program (ILP) based on Enderle et al. (2020) to address the energy-aware resource allocation problem within wide-area networks and prepared the ILP according to the ILP-to-QUBO mapping formalism as presented in Chang et al. (2020) and delineated in Subsection 2.2. They further studied the solvability of this ILP, prepared in QUBO form, on the D-Wave Advantage™, a state-of-the-art quantum

1 100/200 G Tunable Coherent CFP2-DCO Transceiver: https://www.fs.com/de-en/products/120128.html?attribute=5320&id=297112.

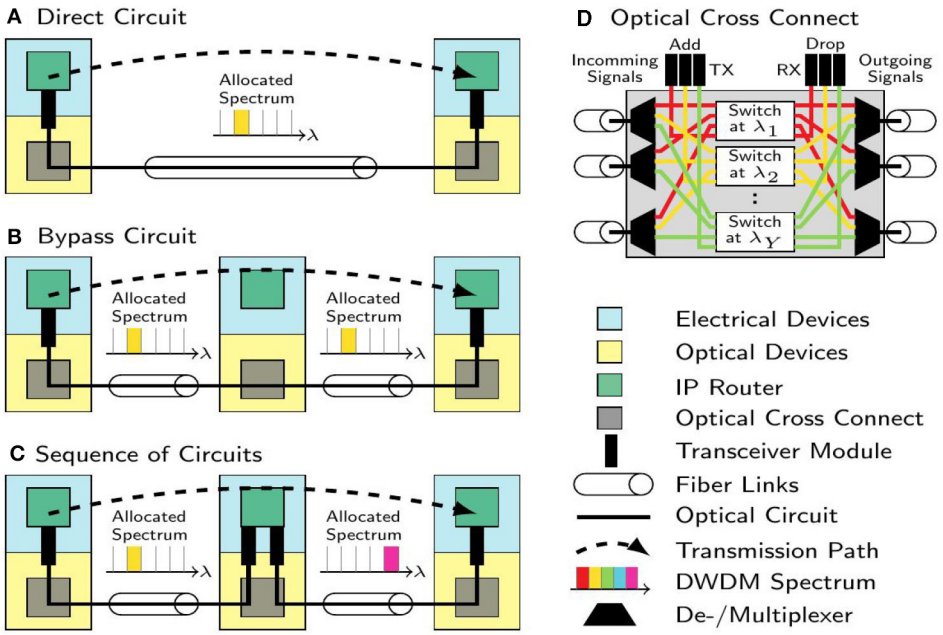


FIGURE 2
(A–C) Various ways of realizing transmission paths in wide-area networks with optical DWDM layer and **(D)** architecture of a OXC as introduced in Witt et al. (2023).

annealer with over 5,000 qubits. Since the current work focuses on improvement methods within the algorithmic part and not on the application itself, we refer to Witt et al. (2023) for a more in-depth explanation and interpretation of the ILP. In the following, we recapitulate the ILP briefly. Parameters and variables are given in Table 1. Traffic volumes h_d per demand d, that are varying over time, are held constant during the optimization and will be updated frequently in a real scenario. The equality constraint (Equation 5) enforces that a demand is routed on exactly one transmission path. Constraint (Equation 6) combines traffic flows per transmission section as selected in Equation (5) and reserves the required capacity in terms of optical channels w_c. Constraint (Equation 7) activates installed transceivers to drive the transmission sections. Minimizing the number of optical channels w_c as defined in objective (Equation 8), reduces the total amount of active transceivers as well. This enables a energy-aware network operation.

Constraints:

$$\sum_{t_d \in T_d} g_{t_d} = 1 \quad \forall d \in D \tag{5}$$

$$-w_c + \sum_{d \in D} \sum_{t_d \in T_d} \rho_{c,t_d} \cdot \frac{h_d}{\xi} \cdot g_{t_d} \leq 0 \quad \forall c \in C \tag{6}$$

$$\sum_{c \in C} w_c \cdot \varphi_{v,c} \leq \eta_v \quad \forall v \in V \tag{7}$$

Objective:

$$\sum_{c \in C} w_c \rightarrow \min \tag{8}$$

The network under test is a fully-connected 3 node network, e.g., the topology has a triangular shape with two short edges of 300 km length and a long edge spanning a 424 km distance. Each network edge is realized by two fiber links to realize bidirectional transmission. Traffic demand values h_d are taken from a normal distribution with mean 75 Gbit/s and standard deviation 20 Gbit/s. As they represent floating numbers, we discretize them with an accuracy of a = 1 (acc. to Witt et al., 2023, Section III-C), i.e., fractions are rounded to "x.0" or "x.5". We set the number of installed transceivers per node to $\eta_v = 15$ and the maximal number of parallel optical signals per transmission path to $\omega_{c,\max} = 3$. The parameter $\omega_{c,\max}$ influences the QUBO's matrix sizes as described in Witt et al. (2023), Section III-C. The parameters ρ_{c,t_d} and $\varphi_{v,c}$ represent the connectivity described by the topology. They are predefined together with the transmission path realization sets T_d. The boolean selector variable g_{t_d}, indicating the selection of a predefined transmission path realization t_d for demand d, and the number of parallel optical signals per transmission path ω_c are determined during the optimization.

2.4 Correlations between solution metrics and system run parameters

With the intent to minimize the objective function, D-Wave provides a distribution of solutions, all of which are not equally important nor of equal quality. The setup of the ILP scenario (penalty term, float variable solution, integer sizes) and the QUBO (sparsity-affecting transformations, embedding, chain strength) parameterize the problem. The annealing procedure (annealing schedule, spin transformations, thermalization/decorrelation

TABLE 1 List of parameters used in the ILP for network optimization.

Parameter		Interpretation
Constants	$\xi \in \mathbb{R}$	Data rate of a single transceiver
	$\eta_v \in \mathbb{N}$	Amount of transceivers installed at node v
	$\rho_{c,t_d} \in \mathbb{B}$	Indicates whether the transmission section c is part of the demand-specific transmission path realization t_d
	$\varphi_{v,c} \in \mathbb{B}$	Indicates whether transmission section c is connected to node v
	$h_d \in \mathbb{R}$	Traffic volume of demand d
Variables	$g_{t_d} \in \mathbb{B}$	Path selector is 1 if a transmission path for demand d is realized by circuit configuration $t_d \in T_d$
	$\omega_c \in \mathbb{N}$	Amount of active transceivers, driving a transmission section c

pauses) can also have significant influence on the obtained distribution of solutions. Studies like Willsch et al. (2022) show that a proper parameter selection in terms of annealing schedule and embedding variants can change the situation significantly. Furthermore, the effect of thermalization in the context of quantum annealing processes can have an impact, as was shown in Dickson et al. (2013). Ideally, the solutions to the problem should not be affected by the choices for these meta parameters. Still, we selected the range of parameters to be tested using our experience garnered from our previous study, see Witt et al. (2023).

However, as we show in later sections, different combinations of parameters significantly affect the likelihood of obtaining feasible solutions. Choices for such meta parameters can be highly correlated. For example, longer (slower) annealing profiles can provide higher probabilities for finding a feasible solution, yet at the cost of generating fewer total number of solutions.

Our first studies, Witt et al. (2023), found that probabilities for finding a minimal feasible solution for the three-node network problem were at the order of 10^{-4}% and below. This presented a non-trivial task to evaluate the quality of the distribution of solutions when only having a sample sizes of $< 10^6$. To address this issue, we formulate statistical measures based on the distribution of samples to quantify the quality of our D-Wave setup. Since the optimal solution is, by definition, a feasible solution, we are interested in the rate in which feasible solutions are produced. We thus consider the feasibility ratio,

$$r_{\text{feasible}} = \frac{N_{\text{feasible}}}{N_{\text{samples}}}, \tag{9}$$

that rates the success of finding N_{feasible} feasible solutions within a solution set with N_{samples} samples.

Another metric of choice for solutions in the binary search space that we use in our research here is the Hamming distance

$$\text{dist}\{\boldsymbol{x}, \boldsymbol{y}\} = \sum_i XOR(x_i, y_i) .$$

This metric gives the number of flipped bits between an ideal solution $\boldsymbol{x}$ obtained by a classical ILP solver like CPLEX or GLPK and a non-ideal solution $\boldsymbol{y}$ obtained by the quantum annealer. For binary solution vectors the Hamming distance is equivalent to the L_2-norm of the difference between $\boldsymbol{x}$ and $\boldsymbol{y}$. This metric provides a sense of "distance" between two solution vectors, essentially telling us how many "bit-flips" are required to bring one solution into another. Ultimately it allows us to perform a direct comparison between particular D-Wave solution vectors and a known desired solution vector.

Finally, we can train a neural network (NN) on these correlations, with the goal that once trained, we can use the NN to make further guesses on optimal solutions vectors. We describe our NN in the following section.

2.5 Machine-learning approach

We employ a decision tree (DT) neural network in our investigations. This NN is a type of supervised machine learning (ML) algorithm that is used typically for regression and classification analysis. It is a model that represents a series of decisions and their possible consequences in the form of a tree-like structure (Breiman et al., 1984). Each node in the tree represents a decision, and each branch represents a possible outcome or path that can be taken based on that decision. In Figure 3 we provide a graphical example of a decision tree and its mapping to a neural network.

A major advantage of decision tree NNs is their ease of use, understandability, and interpretability. This makes their implementation simple and their application efficient. Another advantage comes from their inherent robustness to data outliers. They can even handle missing values in the data. The data itself can be both categorical and numerical in nature.

However, a potential drawback of DTs is that they can easily overfit the data. This ultimately means that, though they may be sufficiently *expressive* to explain the trained data, they fail when extrapolating to new, or unseen data. Thus, the NN is limited in its *generalizability*. This issue can to a certain degree be mitigated by pruning the tree or using other techniques to reduce the complexity of the model. In our studies we did not employ such mitigation techniques, and leave such potential studies for later investigations. We used the `Scikit-learn` python module (Pedregosa et al., 2011) and its functionalities to implement our DT networks.

2.6 Construction of Sherrington-Kirkpatrick graph

The Sherrington-Kirkpatrick (SK) graph encloses the coupling strength and external fields of a Ising Hamiltonian. As mentioned

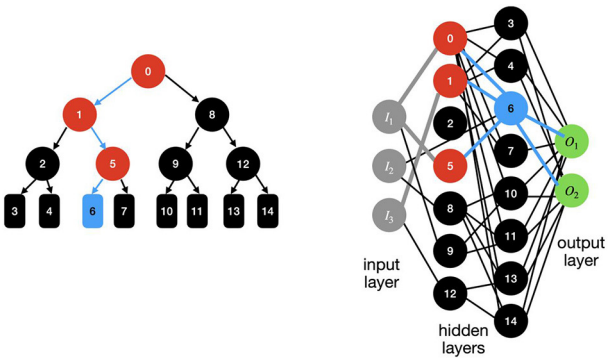


FIGURE 3
A graphical example of decision tree network **(left)** and its mapping to a neural network **(right)**.

in Thai et al. (2022), finding the weighted minimal cut in this graph is equivalent to finding the ground state in the Ising Hamiltonian. Further, the Hamiltonian's energy landscape can be explored by exploration of the SK graph's cut space.

The corresponding SK graph of the Ising Hamiltonian $\mathcal{H}(\boldsymbol{x}) = \boldsymbol{h}^\top \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{J}\boldsymbol{x}$ with n variables x_i can be denoted as $G_H^{SK} = (V, E, w)$ with node set V, undirected edges $(i, j) \in E$ and their weights W. The first n nodes of V correspond to the variables x_i. A further node is added to V to capture the external fields $\boldsymbol{h}$. Set E contains only edges with non-zero weights according to $w_{ij} = J_{ij} + J_{ji}$ for $1 \leq i, j \leq n$ and weights $w_{i,n+1} = h_i$. Then, the weighted adjacency matrix $\boldsymbol{J}'$ of graph G_H^{SK} with $J'_{ij} = J_{ij} + J_{ji}$, $J'_{ji} = 0$ and $J'_{i,n+1} = h_i$ can be used together with $\boldsymbol{y} \in \mathbb{S}^{n+1}$ to define the SK Hamiltonian as

$$\mathcal{H}^{SK}(\boldsymbol{y}) = \boldsymbol{y}^\top \boldsymbol{J}' \boldsymbol{y}.$$

To apply a weighted minimal-cut approach on the SK graph for graph reduction, a cut is defined by a subset $S \subseteq V$, such that $\langle S, V \backslash S \rangle$ contains a set of edges that needs to be cut for separation of S and $V \backslash X$. With $c(S) = \sum_{(u,v) \in \langle S, V \backslash S \rangle} w_{uv}$, the capacity of the cut, a minimal cut is defined as $S^* = mc(G^{SK}) = \underset{S \subseteq V}{\operatorname{argminc}}(S)$ with the minimal capacity of $MC(G^{SK}) = \underset{S \subseteq V}{\operatorname{minc}}(S)$.

3 Results

3.1 Trivial ILP problem

We now provide our findings for our simple ILP problem that we described in Subsubsection 2.3.1. Similar results for the other trivial ILPs we considered are found in the Supplementary material. Note that this problem is sufficiently small that we can determine all possible feasible solutions via brute force, which includes the optimal solution. In this case this corresponds to a total number of $N_{\text{feasible}} = 1,536$ feasible solutions. The whole solution space contains $N_{\text{samples}} = 2^{13} = 8,192$ possible vectors as our binary search vector $\boldsymbol{q}$ has a dimension of 13, see Subsubsection 2.3.1. Thus, the feasibility ratio (Equation 9) for brute force sampling is $r_{\text{feasible}} = 18.75\%$.

3.1.1 Observations and correlations

In Figure 4 we show results for brute force sampling (left side) and a run on the D-Wave Advantage™ (right side) using a penalty of $p = 2$ and default run parameters. In the D-Wave case, just 200 samples are taken, which is a relative small portion (∼2.4%) compared to the complete solution space. We have to remark, that the optimal solution can be found even if the sample set is small. Figure 4 shows the distribution of solutions over energy (upper row) and Hamming distance (middle row) obtained by the mentioned sampling methods and classified by their feasibility demarcated by feasible (green), infeasible (red), and all (blue) solutions. We can observe, that solutions obtained with D-Wave show low energies and only Hamming distances of up to 8. This indicates, that the aimed optimization takes place and only solutions with mostly good qualities are found by D-Wave quantum annealer. But, we still have to sort solutions by feasibility after sampling as minimizing the energy can not entirely sort out infeasible solutions. The lower row of Figure 4 shows the correlations between the solution's energy and their Hamming distance in relation to the best feasible solution. Solutions with small Hamming distances tend to have smaller energy values as observable and indicated by the best fitting curves. We identified, that higher-energy solutions are correlated with increasing Hamming distances to optimal solution as the slope of the fitting curves are non-zero. The energy range for solutions at same Hamming distances is spread widely if the whole search space is considered. Feasible solutions could be found only at the lower energy range. Within the D-Wave sample set, solutions with small Hamming distances are over-proportionally feasible solutions which is indicted by the regression curves.

In the brute force sampling case, we can describe the distribution of solutions upon the Hamming distance (Figure 4 left, middle row) by a cumulative distribution function,

$$CDF(d) = \frac{1}{2^{N_q}} \sum_{d} \binom{N_q}{d} \quad \forall \quad d \in \{0, 1, \ldots, N_q\},$$

with d representing possible Hamming distances for binary search vectors $\boldsymbol{q}$ of length N_q. This relation can be used for benchmarking as it forms a fundamental boundary that only depends on the vector size.

In Figure 5 (upper left panel) we show dependence of the feasibility ratio (9) as a function of anneal time and penalty factor, as well as the average hamming distance (upper right panel) as a function of the same parameters. There is seemingly little correlation between p and the anneal time as long as $p \gtrsim 10$. However, these results suggest that increasing beyond $p \gtrsim 100$ is beneficial since in this region solutions with lower Hamming distance are more likely. It is remarkable, that all feasibility ratios that are shown in Figure 5 are significantly larger than the theoretical value of 18.75% for the case that all possible solutions are considered.

We also encountered a number of individual feasible solution samples obtained in a single run whereby the solution vectors

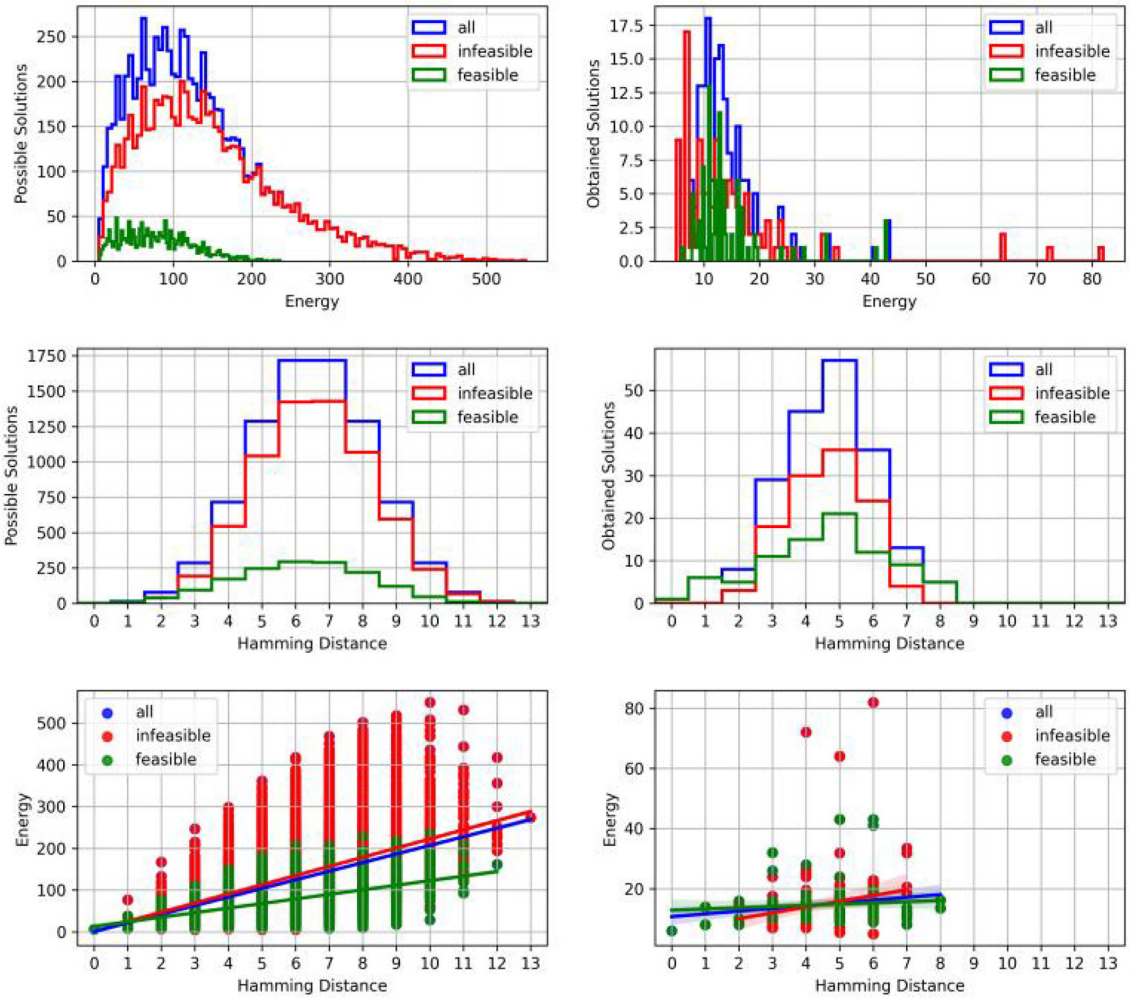


FIGURE 4
(Left) Brute force sampling to investigate the entire solution space. **(Right)** D-Wave sampling with penalty $p = 2$ and a set of 200 samples. **(Upper row)** Histogram of solutions sorted by energy values. As solutions gathered by D-Wave's quantum annealer have only energy values in the lower area compared to the brute force case, x-axis are scaled differently. **(Middle row)** Histogram of solutions over Hamming distance with respect to the best feasible solution vector. **(Lower row)** Scatter plot of solutions with reference to their energy values and the Hamming distance with respect to the best feasible solution vector. (Blue) solution set under investigation. (Red) infeasible solutions. (Green) feasible solutions.

fulfill the ILP's constraints and differ from each other in at least one of its components, but are not necessarily optimal solutions. It can happen that some of these individual feasible solutions can share the same cost value. The parameter dependence of the number of individual feasible solutions is presented in lower left panel of Figure 5. We find that short annealing times generate more individual solutions, however at the expense of reducing the low-energy solutions. So the D-Wave quantum annealer finds more solutions with higher energies if shorter annealing profiles are applied. To no surprise, these correlations suggest that optimizing to longer anneal times will provide lower energy solutions. Similar findings are found for the other trivial ILPs listed in Supplementary material. The relevant figures in this case are Supplementary Figures S1–S3.

We point out that we find no correlations between the parameters `chain_strength` and `annealing_time`, suggesting that further optimization of the `chain_strength` parameter is not possible.

3.1.2 Improvements obtained by machine learning approach

Within a sample set, generated by D-Wave Advantage™, we have 110 independent solutions for our trivial ILP problem when using $p = 2$ and an annealing time of 20 μs. This represents ~10% of the possible feasible solutions. To improve upon this, we train a NN on the correlations described above and then use the NN to generate more solutions.

In particular, we train our NN using the solution vector vs. the energy and feasibility correlations obtained from D-Wave data. With input of energy and feasibility, the decision tree regression predicts a new solution which has the corresponding input energy and feasibility. We note that our NN does not always provide new solutions whose output energy coincides with the same input energy. This is readily seen in the left plot in Figure 6, where the output energy E_{out} is plotted as a function of input energy E_{in}. A one-to-one correspondence would provide a straight line with slope of unity, which is clearly not seen. However, the correlation

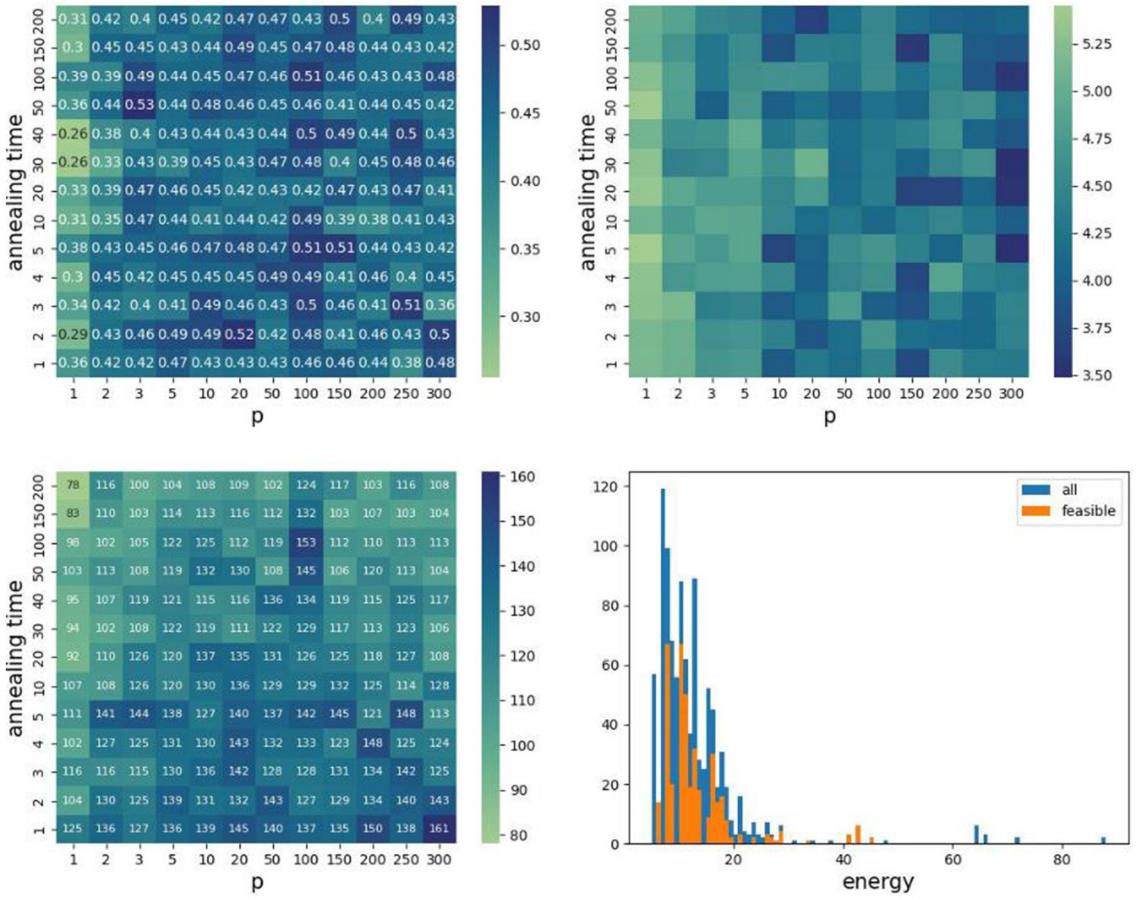


FIGURE 5
Feasibility rate **(Upper left)**, averaged Hamming distance **(Upper right)**, and **(Lower left)** number of individual solutions as a function of anneal time (μs) and penalty factor p. **(Lower right)** Energy distribution at $p = 2$ and annealing time of 20 μs. Values based on samples generated by D-Wave Advantage™ to solve our trivial ILP problem.

between input and output energy as captured by our NN is still positive. We find that the slope of this correlation depends on the p value, whereby larger p values provide a slope closer to unity. Qualitatively similar behavior is found for the ILPs listed in Supplementary material, as can be seen in Supplementary Figure S5 of this document.

We expect the decision tree to recognize the feasibility condition, but predicted solutions of the NN are not always feasible. As mentioned above, the NN predicts solution vectors whose energy ranges have some correlation with the input energies. This feature provides, in principle, an advantage over brute-force sampling since we can target solutions within a specific energy range using our NN, whereas such control via brute force sampling is not possible. However, there isn't a complete one-to-one correspondence between input and output energy since ∼20% of the predicted solution vectors have components that are not binary but contain fractional numbers. In these cases we round the fraction to zero if the fractional number is smaller than 0.2, and to one if larger than 0.8. Between 0.2 and 0.8, we enumerate all possible combinations of 0 and 1, generating in these case new proposed solution vectors. We then perform another feasibility test on these NN solutions to filter out infeasible solutions. The energy distribution of feasible vs. infeasible solutions after this treatment is shown in Figure 6 right.

3.2 Three-node network

We now turn our attention to the 3-node problem, which represents the smallest, non-trivial system of wide-area networks. Here we use CPLEX to obtain the optimal solution vector, from which we make comparisons with D-Wave solution vectors. The distribution of D-Wave solutions as a function of Hamming distance to the optimal solution is given in Figure 7. Note that in this case the optimal solution is not captured by D-Wave. In fact, D-Wave cannot find any feasible solutions within a set of 600,000+ samples. As remark, the entire search space for this case is 2^{63}.

When we investigate inter-parameter correlations, we find little to no correlations between the Hamming distance, `chain_strength`, `anneal_time`, and penalty factor p. This is demonstrated by the nearly flat dependence of the data in Figure 8. This lack of correlation prevents us from obtaining optimized run parameters for this system, and unfortunately

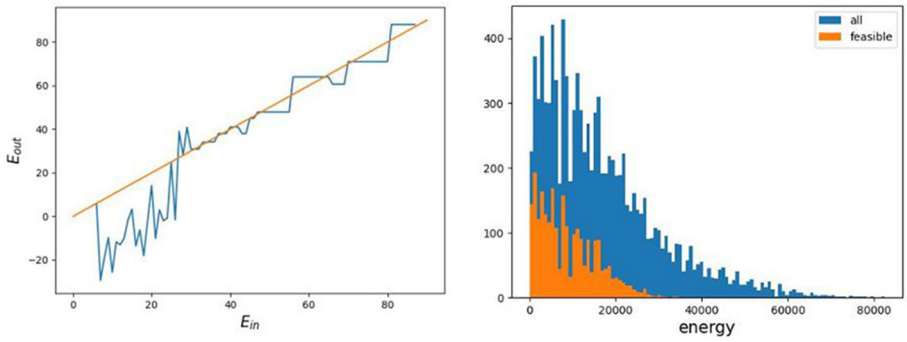


FIGURE 6
Decision tree method to find more feasible solutions based on D-Wave data at $p = 2$ and annealing time = 20 μs.

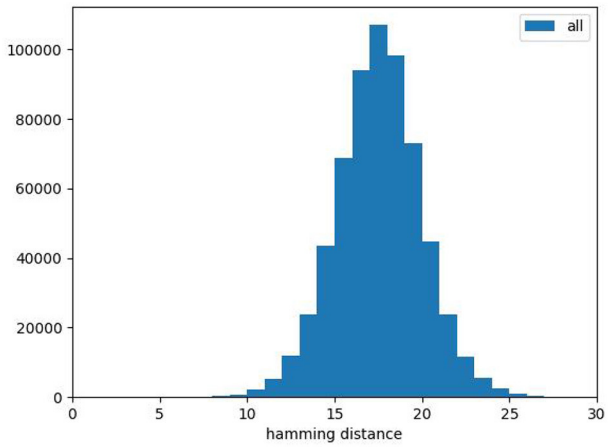


FIGURE 7
Histogram of Hamming distance of the all solutions for 3-node problem. There are no feasible solutions.

suggests that larger node problems will become just as difficult, if not more difficult, to optimize.

These findings already hint at the difficulties we encounter when applying an NN to this system, as we describe in the following section. But we nonetheless train a DT network on the energy and Hamming distance to optimal solution, exactly as described in Subsubsection 3.1.2.

4 Discussion

4.1 Interpretation of findings

The total number of feasible solutions of the trivial ILP problem is 1,536. As previously mentioned, D-Wave finds a little <10% of these solutions, but with our NN we can fully ascertain the full solution space distribution (compare the lower right panel of Figure 5 with that of Figure 6 and see also Supplementary Figures S2, S5). More concretely, we provide the exact number of addition feasible solutions found with our NN as a function of input parameters `annealing time` and penalty factor p in Figure 9 (see Supplementary Figure S4 for our other trivial ILPs). This means that, for our simple ILP problem, the decision tree after round off treatment provided 1,426 new independent feasible solutions. The distribution of new solutions as a function of Hamming distance provided by our ML technique is given in Figure 10. So combining our NN results with D-Wave's, all possible 1,536 solutions were found. Thus, our hybrid classical (ML)/quantum (D-Wave) method allowed us to fully map out the full solution space. We note that our NN is not generalizable to all trivial ILPs, but is unique for each ILP. This is because the solution vector space generated by D-wave is specific for each ILP, and so each NN is trained with this specific solution vector layout. Within our formalism a "master" NN for all trivial ILPs is not possible.

We now discuss our 3-node problem. Note that in this case, D-Wave could not find the optimal solution provided by CPLEX, despite our system parameter investigations mentioned in the previous section. It is not viable to assume that feasible solutions can be found by luck or random guessing. The probability to find the optimal (minimal and feasible) solution is $1/2^{63} \sim 10^{-19}$ in our case. The fact that we could not find feasible solutions within a set of 600.000 samples indicates that feasible solutions are very rare. This was already observed in our previous study (Witt et al., 2023). There, we were not able to find any feasible solutions for some of the test sets and in other cases around 0.2–11 per million samples. There are some possible hints for why this is the case here. First of all, the entire solution set contains only a small portion of feasible solutions that fulfill the ILP. Further, the annealer minimizes the energy of the QUBO Hamiltonian. As it is possible that the lowest energy state can be obtained with various solution vectors someone could find also a energy-wise optimized vector that does not fulfill the ILP. Furthermore, hardware imperfections like noise or limited detection resolution can cause this undesirable behavior.

At this point, critical voices could rate the annealer as an expensive random sampler. But this is not the case as we were able to show that trivial ILP problems are definitely solvable with D-Wave. In these cases, we explicitly used less samples than the solution space's size to avoid an oversampling—somebody could also solve small problems by oversampling even if the sampler is neither a random guess sampler where each solution is equal probable or a minimizing sampler like the quantum annealer. Thus, D-Wave performs better than a random sampler. Clearly the solvability is not the same for the network problem case, and this

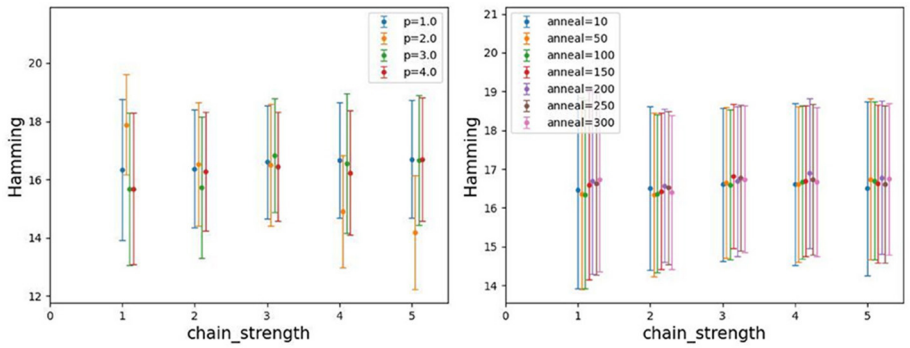


FIGURE 8
Hamming distance in dependence of penalty p and annealing parameters chain strength and annealing time. The Hamming distance is obtained by comparing with the best solution obtained by CPLEX.

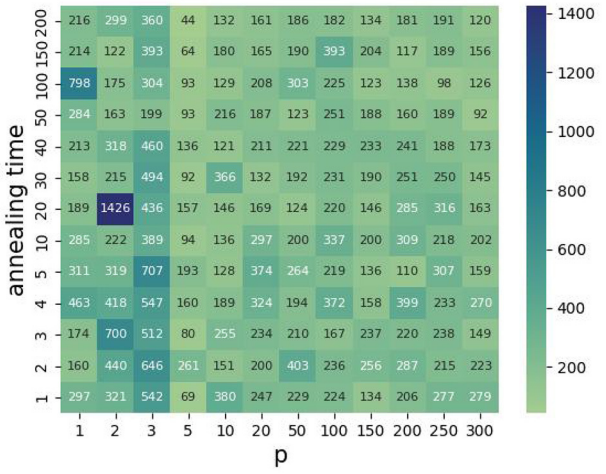


FIGURE 9
The number of new independent feasible solutions found by decision tree for the ILP problem.

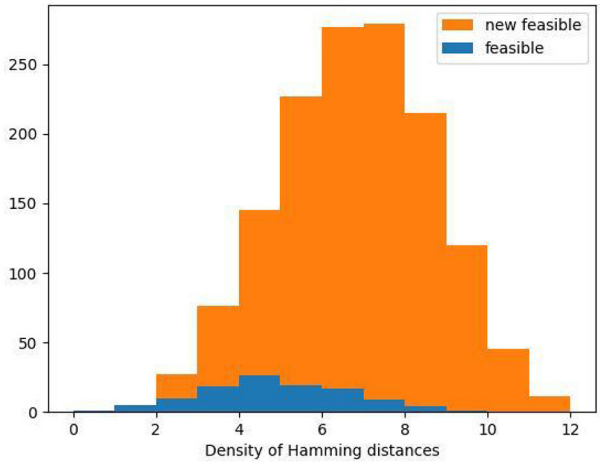


FIGURE 10
Histogram of hamming distance of the feasible solution obtained by D-Wave (blue) and the new feasible solutions obtained by decision tree method for the ILP problem.

may be due to (a) a higher connected QUBO and longer chains of qubits that represent a logical qubit, which cause chain breaks in the quantum annealing hardware to be more likely, (b) numbers in the QUBO matrix have a higher differing range that may be not represented in the hardware well-enough, and of course (c) other issues that are beyond our knowledge.

As part of an approach for improvements, we trained our NN for the 3-node problem with the distributions that we generated from our correlation studies in a comparable way as it was done in the simple ILP problem. Once trained, we found, however, that the NN was unsuccessful in finding any new feasible solutions, let alone the optimal solution. We attribute this to the fact that our D-Wave data distribution of energies (which is used to train the NN) does not cover the energy region of the optimal solution. In fact, as shown in Figure 11, the distribution of D-Wave solutions is far from the optimal solution. Our NN could therefore not generalize sufficiently to lower energy solutions. Compounding the issue is the fact that the distribution of D-Wave solutions contained no feasible solutions, and this in turn limited what the NN could "learn". Thus, our hybrid (ML)/quantum (D-Wave) method failed to produce any new solutions for our 3-node problem.

An obvious question to raise is whether another choice of NN is better suited for our 3-node problem. As we discussed in Subsection 2.5, one of the main advantages that motivated our choice of the decision tree NN is admittedly its ease of use, interpretability, and implementation. However, because of its potential lack of expressivity, one could argue that another choice of NN, e.g., convolutional or recurrent, might lead to better results. This indeed may be the case, and at the least warrants further research. We point out, however, that regardless of the NN architecture, our formalism requires that there exist correlations between hyper-parameters and the resulting D-Wave solutions vectors. It is these correlations that are "learned" by the NN. Since we found no such correlations in our 3-node problem, we suspect that any other type of NN will have similar difficulties as those encountered by our decision tree NN.

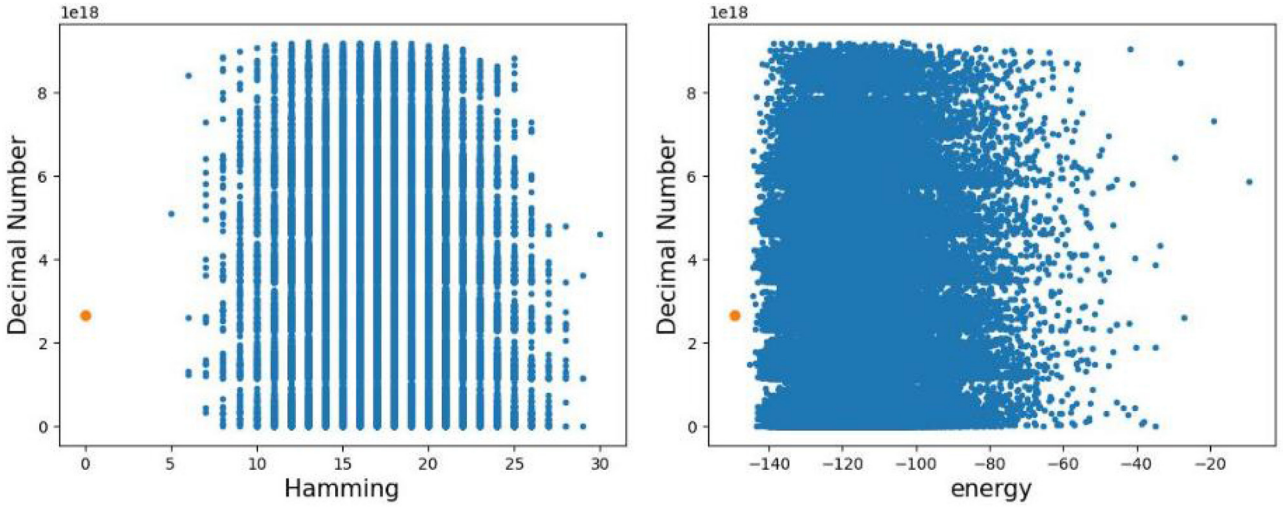


FIGURE 11
Comparing the data distribution and the best solution. We convert the solution vector to decimal number to plot.

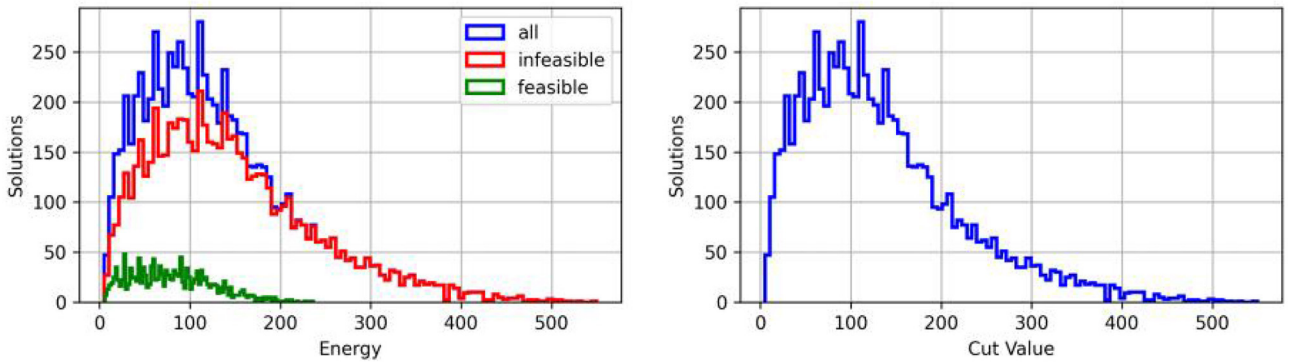


FIGURE 12
Evaluation of the trivial ILP problem in representation as Sherrington-Kirkpatrick (SK) graph. **(Left)** Distribution of energy for SK graph's Hamiltonian if all possible solutions are considered, **(Right)** distribution of all cut values in SK graph.

4.2 Outlook on further improvements

Still there may be ways to improve the situation. Our studies to date have only varied the annealing profile. Instead, one may perform reverse annealing, where the annealing is run "backwards" from a starting classical solution, allowing for exploration of the energy landscape around the classical solution. We are actively investigating this procedure. Reverse annealing may be also applicable to set initial states as shown in Pelofske et al. (2023). Thus, expected solutions or solutions that are close to an expected solution can be set as start value for the annealing process. If the optimizer is applied frequently—a typical situation in network optimization—the last obtained solution can be used for the initialization of the next run as new optimal network configurations might be close to the last configuration.

Annealing parameter like annealing schedules and various embeddings for our problems can be studied more detailed like in the study of Willsch et al. (2022). The authors of Willsch et al. (2022) discovered an increase in the success rate for proper settings in the annealing schedule. In our case we observed a more or less constant success rate, especially for the 3-node network problem. Apart from that it may be valuable to study our approach on a larger set of similar problems to get a more general perspective. Unfortunately, we had to restrict our study on a single problem instance as the amount of feasible solutions for our problem is very rare and large sampling sets are required for the analyzes. Besides, thermalization within the annealing process can be studied as well, see Dickson et al. (2013).

Furthermore, since the size of the problem that is embedded on the quantum annealer plays a crucial role for its solvability, methods for efficient embedding or problem reduction should be incorporated within future studies. We point out that the work of Thai et al. (2022) seems promising in reducing the demands on the number of physical qubits. Here the authors introduced a fast Hamiltonian reduction algorithm (FastHare) that defines non-separable groups of qubits, i.e., qubits that obtain the

same value in optimal solutions, and performed a reduction by merging non-separable groups into single qubits. This could be done within a worst case time complexity of $\mathcal{O}(\alpha n^2)$ with a user-defined parameter α. The authors of Thai et al. (2022) showed in a benchmark that their algorithm is capable of saving 62% of physical qubits on average within a processing time of 0.3 s, outperforming the roof duality–the reduction used within the D-Wave's software development kit SDK. We reviewed parts of their work. In particular, we mapped our trivial ILP problem to a so-called Sherrington-Kirkpatrick (SK) graph. We further evaluated all cut values within this graph. The results (Figure 12) show that the cut values in the SK graph correspond to the energy values of QUBO or Ising problem solution vectors. As the Hamiltonian reduction is based on graph compression on basis of minimal cuts, we expect that the proposed algorithm (Thai et al., 2022) can improve the situation, as a reduced Hamiltonian might be better solvable on the D-Wave quantum annealer.

Unfortunately, we were not able to fully implement and apply this sophisticated algorithm as we struggled at the following point. The algorithm applies a min-cut algorithm on the SK graph to detect non-separable qubit groups. Originally, we though that a standard min-cut algorithm could be applied here. Unfortunately, the for us available min-cut algorithms can be only applied in graphs with positive-weighted edges. But, due to the nature of QUBO, Ising or SK Hamiltonians, the edges in a SK graph may have negative-valued edge weights. This issue was not addressed in their work (Thai et al., 2022). However, it remains unsure, if the fast Hamiltonian Reduction (FastHare) algorithm can improve the solvability of our ILPs with D-Wave's annealer as the authors used randomly generated graph structures in their evaluation, i.e., the graphs are weakly connected and as such well-suited for graph compression.

Beside the ILP to QUBO mapping formalism that was described in Chang et al. (2020) and Witt et al. (2023), someone could model the problem in a differing way. One possibility is the introduction of constraint-specific penalty factors, that create new degrees of freedom usable for problem-specific optimization of the algorithm. It can be achieved by the use of a penalty vector $\boldsymbol{p}^\top = [p_1, p_2, \ldots, p_m]$ and a corresponding penalty matrix $\boldsymbol{P} = \boldsymbol{I}\boldsymbol{p}$ inside the formulations. The QUBO Hamiltonian and thus the objective to be optimized is then

$$\mathcal{H}(q) = \boldsymbol{q}^\top \boldsymbol{Q}\boldsymbol{q} + C \rightarrow \min,$$

$$\text{with} \quad \boldsymbol{Q} = \begin{bmatrix} \boldsymbol{Q}_{xx} & \boldsymbol{Q}_{xs} \\ \boldsymbol{Q}_{sx} & \boldsymbol{Q}_{ss} \end{bmatrix}, \quad C = \boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{b} \text{ and}$$

$$\begin{aligned} \boldsymbol{Q}_{xx} &= \boldsymbol{Z}_x^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{Z}_x + \operatorname{diag}\left\{\left(2\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{A} + \boldsymbol{c}^\top\right)\boldsymbol{Z}_x\right\}, \\ \boldsymbol{Q}_{xs} &= \boldsymbol{Q}_{sx}^\top = \boldsymbol{Z}_x^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{Z}_s, \\ \boldsymbol{Q}_{ss} &= \boldsymbol{Z}_s^\top \boldsymbol{P}\boldsymbol{Z}_s + 2\operatorname{diag}\left\{\boldsymbol{Z}_s^\top \boldsymbol{P}\boldsymbol{b}\right\}. \end{aligned}$$

This extends the ILP to QUBO mapping formalism to a generalized form. Required details could be found in Witt et al. (2023), Section III-D.

4.3 Outcome

Our work can be summarized as follows. The approach aims to solve ILPs with a quantum annealing attempt. We tried to find optimal annealing parameters and discovered weak correlations between annealing parameters and success rates in the 3-node network case. Further, a decision tree ML approach was applied to increase the rate of feasible ILP solutions. We realized that further improvements are needed to overcome remaining hurdles and discussed some attempts therefore. Even as the results for the 3-node problem are not fully satisfying, we are able to show with less complicated ILP problems that the approach works in principle. Thus, we expect that the approach can be extended in a way that larger problem instances are also solvable.

Finally, fast ILP-solving methods can have a significant impact on systems that should be optimized in real time. As an example, a novel mode of real-time network operation in wide-area networks is studied in Witt (2024). Here, similar ILPs are used to define a frequently applied network optimization.

Data availability statement

The datasets presented in this article are not readily available because it is not intended to share the data publicly. Requests to access the datasets should be directed to AW, arthur.witt@ieee.org.

Author contributions

AW: Formal analysis, Investigation, Methodology, Resources, Software, Visualization, Writing – original draft. JK: Formal analysis, Investigation, Methodology, Resources, Software, Visualization, Writing – review & editing. CK: Data curation, Writing – review & editing. TL: Project administration, Writing – original draft.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. AW was supported by the German Federal Ministry of Education and Research (Project ID 16 KIS 1312) which is partly funding the work that has been performed in the framework of the CELTIC-NEXT EUREKA project AI-NET ANTILLAS (Project ID C2019/3-3). JK and TL were supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through the funds provided to the Sino-German Collaborative Research Center TRR110 "Symmetries and the Emergence of Structure in QCD" (DFG Project-ID 196253076—TRR 110). The Jülich Supercomputing Centre funded this project by providing computing time through the Jülich UNified Infrastructure for Quantum computing (JUNIQ) on the D-Wave Advantage™ quantum system. This publication was funded by the German Research Foundation (DFG) grant "Open Access Publication Funding/2023–2024/University of Stuttgart" (512689491).

Acknowledgments

The authors gratefully acknowledge the Jülich Supercomputing Centre for funding this project by providing computing time through the Jülich UNified Infrastructure for Quantum computing (JUNIQ) on the D-Wave Advantage™ quantum system. They further acknowledge the project funding received.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2024.1356983/full#supplementary-material

References

Adler, I., Papadimitriou, C., and Rubinstein, A. (2014). "On simplex pivoting rules and complexity theory," in *Integer Programming and Combinatorial Optimization*, eds. J. Lee, and J. Vygen (Cham: Springer International Publishing), 13–24.

Breiman, L., Friedman, J., Stone, C., and Olshen, R. (1984). *Classification and Regression Trees*. New York, NY: Taylor & Francis.

Chang, C. C., Chen, C.-C., Körber, C., Humble, T. S., and Ostrowski, J. (2020). *Integer Programming from Quantum Annealing and Open Quantum Systems*. Ithaca, NY.

Chiaraviglio, L., Mellia, M., and Neri, F. (2009). "Energy-aware backbone networks: a case study," in *2009 IEEE International Conference on Communications Workshops* (Dresden), 1–5.

Dickson, N. G., Johnson, M. W., Amin, M. H., Harris, R., Altomare, F., Berkley, A. J., et al. (2013). Thermally assisted quantum annealing of a 16-qubit problem. *Nat. Commun.* 4:2920. doi: 10.1038/ncomms2920

Enderle, T., Witt, A., and Christou, F. (2020). "Delay-differentiated routing in meshed backbone networks," in *Photonic Networks; 21th ITG-Symposium* (Berlin: VDE Verlag), 20–27.

Feller, F. (2012). "An optimization-heuristic approach to dynamic optical bypassing," in *13th ITG Fachtagung on Photonic Networks* (Leipzig: VDE Verlag).

ITU-T (2020). *Recommendation ITU-T G.694.1, SERIES G: Transmission Systems and Media, Digital Systems and Networks Transmission Media and Optical Systems Characteristics—Characteristics of Optical Systems; Spectral Grids for WDM Applications: DWDM Frequency Grid*. Geneva: International Telecommunication Union - Telecommunication Standardization Sector.

Karp, R. M. (1972). *Reducibility among Combinatorial Problems*. Boston, MA: Springer US, 85–103.

Lenstra, H. W. (1983). Integer programming with a fixed number of variables. *Math. Operat. Res.* 8, 538–548. doi: 10.1287/moor.8.4.538

Nash, J. (2000). The (Dantzig) simplex method for linear programming. *Comp. Sci. Eng.* 2, 29–31. doi: 10.1109/5992.814654

Nguyen, D., and Pak, I. (2017). "The computational complexity of integer programming with alternations," in *32nd Computational Complexity Conference (CCC 2017), number 6, 6:1-6:18. Leibniz International Proceedings in Informatics*, ed. R. O'Donnell (Dagstuhl).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* 12, 2825–2830. doi: 10.48550/arXiv.1201.0490

Pelofske, E., Hahn, G., and Djidjev, H. (2023). Initial state encoding via reverse quantum annealing and h-gain features. *IEEE Transact. Quant. Eng.* 4, 1, 21. doi: 10.1109/TQE.2023.3319586

Thai, P., Thai, M. T., Vu, T., and Dinh, T. N. (2022). "FastHare: fast hamiltonian reduction for large-scale quantum annealing," in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)* (Los Alamitos, CA: IEEE Computer Society), 114–124.

Tornatore, M., Maier, G., and Pattavina, A. (2002). "WDM network optimization by ILP based on source formulation," in *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, Volume 3* (New York, NY), 1813–1821.

Willsch, D., Willsch, M., Gonzalez Calaza, C. D., Jin, F., De Raedt, H., Svensson, M., et al. (2022). Benchmarking advantage and D-wave 2000Q quantum annealers with exact cover problems. *Quant. Inf. Process.* 21, 1–22. doi: 10.1007/s11128-022-03476-y

Witt, A. (2024). "Queue-aware network control algorithm with a high quantum computing readiness-evaluated in discrete-time flow simulator for fat-pipe networks," in *IEEE 25th International Conference on High Performance Switching and Routing (HPSR) 2024* (Pisa).

Witt, A., Körber, C., Kirstädter, A., and Luu, T. (2023). "Tactile network resource allocation enabled by quantum annealing based on ILP modeling," in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (Bellevue, WA), 670–680.

frontiers | Frontiers in Computer Science

TYPE Brief Research Report
PUBLISHED 12 June 2024
DOI 10.3389/fcomp.2024.1335369

OPEN ACCESS

EDITED BY
David Esteban Bernal Neira,
Purdue University, United States

REVIEWED BY
Teng Bian,
Facebook, United States
Marcos César de Oliveira,
State University of Campinas, Brazil
Steven Adam Abel,
Durham University, United Kingdom

*CORRESPONDENCE
Roberto Sebastiani
✉ roberto.sebastiani@unitn.it

RECEIVED 08 November 2023
ACCEPTED 20 May 2024
PUBLISHED 12 June 2024

CITATION
Ding J, Spallitta G and Sebastiani R (2024) Experimenting with D-Wave quantum annealers on prime factorization problems. *Front. Comput. Sci.* 6:1335369. doi: 10.3389/fcomp.2024.1335369

Experimenting with D-Wave quantum annealers on prime factorization problems

Jingwen Ding, Giuseppe Spallitta and Roberto Sebastiani*

Department of Information Science and Engineering, University of Trento, Trento, Italy

This paper builds on top of a paper we have published very recently, in which we have proposed a novel approach to prime factorization (PF) by quantum annealing, where $8,219,999 = 32,749 \times 251$ was the highest prime product we were able to factorize—which, to the best of our knowledge is the largest number which was ever factorized by means of a quantum device. The series of annealing experiments which led us to these results, however, did not follow a straight-line path; rather, they involved a convoluted trial-and-error process, full of failed or partially-failed attempts and backtracks, which only in the end drove us to find the successful annealing strategies. In this paper, we delve into the reasoning behind our experimental decisions and provide an account of some of the attempts we have taken before conceiving the final strategies that allowed us to achieve the results. This involves also a bunch of ideas, techniques, and strategies we investigated which, although turned out to be inferior wrt. those we adopted in the end, may instead provide insights to a more-specialized audience of D-Wave users and practitioners. In particular, we show the following insights: (*i*) different initialization techniques affect performances, among which flux biases are effective when targeting locally-structured embeddings; (*ii*) chain strengths have a lower impact in locally-structured embeddings compared to problem relying on global embeddings; (*iii*) there is a trade-off between broken chain and excited CFAs, suggesting an incremental annealing offset remedy approach based on the modules instead of single qubits. Thus, by sharing the details of our experiences, we aim to provide insights into the evolving landscape of quantum annealing, and help people access and effectively use D-Wave quantum annealers.

KEYWORDS

quantum computing, quantum annealing, prime factorization, embedding, experimental analysis

1 Introduction

Quantum computing has emerged as a novel paradigm in computer science, offering the potential capabilities to solve complex problems that have long remained intractable for classical computers. Among the various approaches within quantum computing, *quantum annealers (QA)* stand out as a promising tool for tackling challenging computational tasks. To this extent, *prime factorization (PF)*—i.e., the problem of breaking down a number into its prime factors—is a good candidate to be effectively solved by quantum computing, in particular by quantum annealing. This problem is of utmost significance in modern cryptography, where the security of systems often relies on the presumed computational intractability of PF (Rivest et al., 1978). Several approaches have been presented to address PF by quantum computing (e.g., Vandersypen et al., 2001; Lucero et al., 2012; Martín-López et al., 2012; Monz et al., 2016; Amico et al., 2019; Selvarajan et al., 2021), by quantum annealing

(e.g., Dridi and Alghassi, 2017; Jiang et al., 2018; Mengoni et al., 2020), or by hybrid quantum-classical technologies (e.g., Wang et al., 2020; Karamlou et al., 2021). See Willsch et al. (2023) and Ding et al. (2024) for a summary.

This paper builds on top of a paper we have published very recently (Ding et al., 2024), in which we have proposed a novel approach to PF by quantum annealing, with two main results. First, we have presented a very compact modular *encoding* of a binary multiplier circuit into the Pegasus QA architecture, which allowed us to encode up to a 21×12-bit multiplier (or alternatively a 22×8-bit one) into the Pegasus 5760-qubit topology of D-Wave Advantage annealers. Due to the modularity of the encoding, this number will scale up automatically with the growth of the qubit number in future chips. Second, we have investigated the problem of actually *solving* encoded PF problems by running an extensive experimental evaluation on a D-Wave Advantage 4.1 quantum annealer. In these experiments we have introduced different approaches to initialize the multiplier qubits, and adopted several performance-enhancement annealing strategies. Overall, within the limits of our QPU resources, $8,219,999 = 32,749 \times 251$ was the highest prime product we were able to factorize—which, to the best of our knowledge, is the largest number which was ever factorized by means of a "pure" quantum device (i.e., without adopting hybrid quantum-classical techniques).

In this paper we delve into the reasoning behind our experimental decisions and provide a more comprehensive account of the steps and attempts we have taken before conceiving the final strategies which allowed us to achieve the results in Ding et al. (2024). We illustrate a bunch of ideas, techniques, and strategies we investigated which, although turned out to be inferior wrt. those we adopted in the end —and as such were not of interest for the more general public targeted in Ding et al. (2024)— may instead provide insights to a more-specialized audience of D-Wave QA users and practitioners. In particular, we show the following insights: (*i*) different initialization techniques affect performance, among which flux biases are effective when targeting locally-structured embeddings; (*ii*) chain strengths have a lower impact in locally-structured embeddings compared to problems relying on global embeddings; (*iii*) there is a trade-off between a broken chain and excited CFAs, suggesting an incremental annealing offset remedy approach based on the modules instead of single qubits. Thus, by sharing the details of our experiences, including both successes and setbacks, we aim to provide insights into the evolving landscape of quantum annealing and help people access and effectively use D-Wave quantum annealers.

2 Methods

We first summarize a few concepts from Ding et al. (2024). The prime factorization problem (PF) of a biprime number N can be addressed by SAT solvers by encoding a $n \times m$ multiplier into a Boolean formula, fixing the values of the output bits s.t. to encode N. In Ding et al. (2024), we presented a modular *embedding* of a binary multiplier circuit into the Pegasus QA architecture, based on locally-structured embedding of SAT problems (Bian et al., 2020). The multiplier circuit, represented in terms of a conjunction of *Controlled Full-adder (CFA)* Boolean functions linked by means of equivalences between variables, is embedded into the Pegasus topology, with each CFA embedded into a 8-qubit tile and with the variable equivalences implemented through *chains*. Each CFA $F(\underline{\mathbf{x}})$ is encoded in terms of a *penalty function*:

$$P_F(\overbrace{\underline{\mathbf{x}},\underline{\mathbf{a}}}^{\underline{\mathbf{z}}}|\underline{\boldsymbol{\theta}}) \stackrel{\text{def}}{=} \theta_0 + \sum_{z_i\in V}\theta_i z_i + \sum_{(z_i,z_j)\in E, i<j}\theta_{ij} z_i z_j; \quad z_i \in \{-1,1\}; \tag{1}$$

$$s.t.\ \forall \underline{\mathbf{x}}\ \ min_{\{\underline{\mathbf{a}}\}} P_F(\underline{\mathbf{x}},\underline{\mathbf{a}}|\underline{\boldsymbol{\theta}}) \begin{cases} = 0 & \text{if } F(\underline{\mathbf{x}}) = \top \\ \geq g_{min} & \text{if } F(\underline{\mathbf{x}}) = \bot \end{cases} \tag{2}$$

where the Boolean variables $\underline{\mathbf{x}}$ and $\underline{\mathbf{a}}$ are mapped into a subset $\underline{\mathbf{z}} \subseteq V$ of the qubits in the topology graph (V, E), s.t. the qubit values $\{1, -1\}$ are interpreted as the truth values $\{\top, \bot\}$ respectively; θ_0, θ_i, θ_{ij} and g_{min} are called respectively *offset*, *biases*, *couplings* and the *gap*; the offset has no bounds, whereas the range for biases and couplings is $[-4, +4]$ and $[-2, +1]$ respectively. (The ancilla variables $\underline{\mathbf{a}}$ are needed to address the over-constrainedness of the encoding problem.) The $\underline{\boldsymbol{\theta}}$ values in $P_F(\underline{\mathbf{x}},\underline{\mathbf{a}}|\underline{\boldsymbol{\theta}})$ have been synthesized by means of OptiMathSAT (Sebastiani and Trentin, 2020) s.t. to maximize g_{min}.[1] The penalty function of the whole multiplier is thus produced as the sum of the penalty functions of the CFAs, plus a term $(2 - 2zz')$ for every chain $\langle z, z'\rangle$. Then it is fed to the annealer, forcing the values of the output qubits so that to represent the biprime number N, and forcing to -1 the value of the carry-in qubit of the rightmost CFA of each row, and the value of the in2 qubit of the CFAs in the first row in the multiplier. Therefore, if the annealer finds a ground state s.t. such penalty function is zero, then the values of the qubit represent a solution of the PF problem.[2] (We refer the reader to Ding et al. (2024) for a much more detailed explanation).

2.1 Alternative approaches to initialize qubits

Solving prime factorization of a specific number N requires some of the qubits to be initialized to some fixed value in $\{-1, 1\}$. For instance, given an 8-bit multiplier and $N = 42$, whose binary representation is 00101010, then the qubits of the CFAs corresponding to the output number should be initialized respectively to $\{-1, -1, 1, -1, 1, -1, 1, -1\}$; also, e.g., the carry-in qubit of the CFA for the least significant bit in a number must be set to -1. D-Wave API offers a native function, fix_variables(), that replaces the truth values of the qubits into the penalty function.

Abbreviations: CFA, Controlled Full-Adder; OMT, Optimization Modulo Theories; PF, Prime Factorization; QA, Quantum Annealer; SAT, Propositional Satisfiability.

1 The bigger is g_{min}, the easier is for the annealer to discriminate solutions from non-solutions (Bian et al., 2020).

2 From Equations (1) and (2) we notice that, due to non-minimum values of $\underline{\mathbf{a}}$, in principle we can have solution scenarios where $F(\underline{\mathbf{x}}) = \top$ and $0 < P_F(\underline{\mathbf{x}},\underline{\mathbf{a}}|\underline{\boldsymbol{\theta}}) < g_{min}$, which we can recognize as solutions, or even s.t. $P_F(\underline{\mathbf{x}},\underline{\mathbf{a}}|\underline{\boldsymbol{\theta}}) \geq g_{min}$, for recognizing which we need testing $F(\underline{\mathbf{x}}) = \top$ explicitly, which can be performed very easily.

Unfortunately, this causes a subsequent rescaling of all weights if one bias or coupling does not fit into the proper range, reducing thus the gap g_{min} accordingly.

The initialization of qubits can be implemented either at the encoding level [i.e., by imposing qubit values directly into the penalty function $P_F(\underline{\mathbf{x}}|\underline{\boldsymbol{\theta}})$], or at the hardware level (i.e., by imposing the qubit values through the tuning of the quantum annealer hardware). In Ding et al. (2024) we adopted the latter implementation by tuning flux biases, and showed the benefits they brought to the success probability of reaching the ground state. In this paper, we mainly focus on the former type of implementation, proposing a few alternatives to fix_variables():

- ***Ad-hoc* encoding for the CFAs**: we substitute the values of the input variables into the corresponding CFAs and then re-encode these initialized CFAs, with reduced graphs, into new CFA penalty functions. For instance, suppose we want to set the value of *c_in* to false. Then we feed to the OMT solver the extended formula $F'(\underline{\mathbf{x}}) = F(\underline{\mathbf{x}}) \wedge \neg c\_in$ to generate a new specialized penalty function. To prevent the g_{min} from being scaled down due to the input values, during the re-encoding process we take into account all combinations of possible inputs that occur in the CFAs.[3] This results into the generation of an *offline library of specialized CFAs*, with increased minimal gaps, $g_{min} \in [3, 18]$. Notice that, using these modified encodings, we obtained some solutions where $F(\underline{\mathbf{x}}) = \top$ and $0 < P_F(\underline{\mathbf{x}}, \underline{\mathbf{a}}|\underline{\boldsymbol{\theta}}) < g_{min}$ (see text footnote 2), which never occurred in the experiments reported in Ding et al. (2024). Both the gap increment and the extra solutions can increase the probability to find solutions.
- **Extra chaining**: we notice that in the penalty functions of CFA we have obtained, the biases of the qubits are all within $[-1, 1]$, whereas the range for the D-Wave Advantage 4.1 is $[-4, 4]$. Based on these facts, we have explored a simple alternative way to initialize qubits, without the risk of rescaling down the g_{min} value. Specially, in order to assign qubit z to 1 [resp. -1], we can add the penalty function for $z = 1$ $(2 - 2z)$ [resp. for $z = -1$ $(2 + 2z)$] to the penalty function of the multiplier, s.t. the bias of z safely remains in $[-3, 3]$. Equivalently, we can find an unused neighbor qubit z' (if any), add an equivalence chain between z and z', and then initialize z' to 1 (resp -1) by fix_variables().

2.2 The impact of chain strength in QA for modular encodings

The effect of chain strength has been previously studied in the context of *global embedding*, where the input problem is first encoded into a QUBO problem, which is then embedded into the hardware by means of embedding algorithms. The main issue of that approach is that the QUBO model does not know in advance how many chains are there in a specific topology and where they will be placed. Thus, the addition of chains a-posteriori—whose length and placement are out of the control of the user—and the consequent re-scaling of biases and coupling may affect the performances of the algorithm.

Our *locally-structured embedding* approach in Ding et al. (2024) differs from the above approach because the Ising model that is generated is already hardware-compliant, so that there is no risk of weights rescaling, and we do not need a fine-grained analysis of chain strength. Given the modularity of our encoding and the presence of chains to allow communication between neighboring modules, however, it is still important to investigate the side effects of chain strength in modular encoding. To this extent, we choose a set of chain strengths, $c \in \{1, 1.5, 2\}$, as representatives for investigating their effects on the performance of our locally structured embedding approach on QA systems.

2.3 Incrementally remedying excited CFAs

We assume that if all CFAs in a multiplier reach the ground state with high probability, then the success probability of the whole multiplier will be positively affected. Based on this assumption, we have proposed an incremental remedy strategy, to remedy the most excited CFAs during the solving process.

The remedy approach is based on *anneal offsets* (DWave, 2021). In the standard annealing process of D-Wave systems, the annealing schedules are set identically for all qubits. However, the system also allows for adjusting the annealing schedule for each qubit. This is implemented by offsetting the global, time-dependent bias signal $c(s)$ that controls the annealing process. More specially, for a qubit q_i, its anneal offsets $\pm\delta c_i \neq 0$ correspond to advancing and delaying the standard annealing schedule, respectively.

In a fashion similar to Andriyash et al. (2016), Lanting et al. (2017), Yarkoni et al. (2019), and Adame and McMahon (2020) we adopted the idea of incrementally fixing annealing offset weights to increase the probability of reaching a ground state. Differently from these papers, however, where the annealing offset is set to qubits, we set modules of our encoding (i.e., CFAs) as the target of annealing offset tuning, and we choose the number of excitations of these modules as a measure to guide the remedy strategy process.

In each step of our incremental remedy approach, we first find the most-excited CFA —i.e., the CFA whose number of excitation occurrences out of the 1,000 samples is maximum— and then continue to advance the annealing process of all its qubits by annealing offset $\delta c_i = 0.01$, on top of the previous remedying history, until the CFA is no longer the most excited. The procedure terminates either if the system reaches one ground state or if it reaches a certain number of steps set as a threshold. This threshold is chosen according to the

3 This is made necessary by one further technique, namely *qubit sharing*, which we have introduced in Ding et al. (2024) and which is not explained here.

TABLE 1 Different initialization approaches for solving small-size PF, with the annealing time $T_a = 10\ \mu s$ and 1,000 samples for each problem instance.

Size	Inputs	CFA0				CFA1		
		API	*Ad-hoc*	Chain	Flux-bias	API	Chain	Flux-bias
3×3	25 = 5×5	161	154	93	308	327	173	136
	35 = 5×7	389	666	286	711	410	379	951
	49 = 7×7	450	577	312	906	344	295	997
4×4	121 = 11×11	17	4	30	63	9	33	0
	143 = 11×13	40	52	28	129	122	32	67
	169 = 13×13	31	54	4	312	84	69	5
5×5	289 = 17×17	5	0	0	1	3	1	0
	323 = 17×19	2	0	1	7	22	3	0
	361 = 19×19	1	1	0	1	11	1	3
	391 = 17×23	6	1	4	119	5	19	9
	437 = 19×23	17	0	3	67	3	2	0
	493 = 17×29	3	6	0	4	8	0	2
	527 = 17×31	21	11	6	91	6	5	37
	529 = 23×23	5	0	3	8	0	1	8
	551 = 19×29	0	11	4	24	2	3	4
	589 = 19×31	16	13	11	7	1	22	52
	667 = 23×29	0	6	2	3	8	9	105
	713 = 23×31	11	12	3	26	2	1	138
	841 = 29×29	5	9	8	148	14	8	7
	899 = 29×31	17	76	5	222	7	13	343
	961 = 31×31	1	43	0	37	1	0	338

limitation on the access of QuPU, e.g., the perimeter of the multiplier embedded.

3 Results

3.1 Results of different initialization approaches of qubits

In the experiments, we compare the proposed initialization approaches on D-Wave Advantage system 4.1 for factoring small integers of up to 5 × 5 bits, with the annealing time ($T_a = 10\mu s$) and 1,000 samples set for each problem instance. Table 1 by comparing the performances of the initialization techniques , we notice that the *ad-hoc* re-encoding outperforms the native API and the extra-chain approaches, but it still does not perform as well as the flux-bias tuning, which we finally adopted in Ding et al. (2024). In Ding et al. (2024) we also proposed a variant of the CFA function, namely CFA1, minimizing the number of unsatisfying assignments with g_{min} equal to 2. For the sake of completeness, we also tested this encoding in combination with initialization techniques other than flux biases. These results confirm that the combination of the flux-bias initialization and the improved CFA1, which we adopted in Ding et al. (2024), produces the highest success probability for D-Wave Advantage 4.1 in finding solutions. For this reason, we continue to use this combination, the flux-bias initialization + CFA1, in the following experiments of this paper.

3.2 Results of different chain strengths

Using the initialization approach based on CFA1 + flux biases and the same configuration of the annealing system ($T_a = 10\mu s, 1,000$ samples for each problem instance) of previous experiments, we test different chain strengths ($c \in \{1, 1.5, 2\}$) for QA factoring integers from 3 × 3 up to 11 × 8 bits, using the 10 highest co-prime number for each multiplier size.

In Figure 1, left we summarize the results of all samples provided the QA. Sorting them by the size of the input problem (x-axis), we plot respectively the number of samples successfully reaching the ground state (first plot), the number of samples having no broken chain (second plot), and the number of samples having no excited CFA (third plot). In general, we report the score of the median sample among all problems (dashed line) as a summary of the annealer behavior for each sample size. In addition, for each sample size, we provide information on the problem that reaches the ground state the

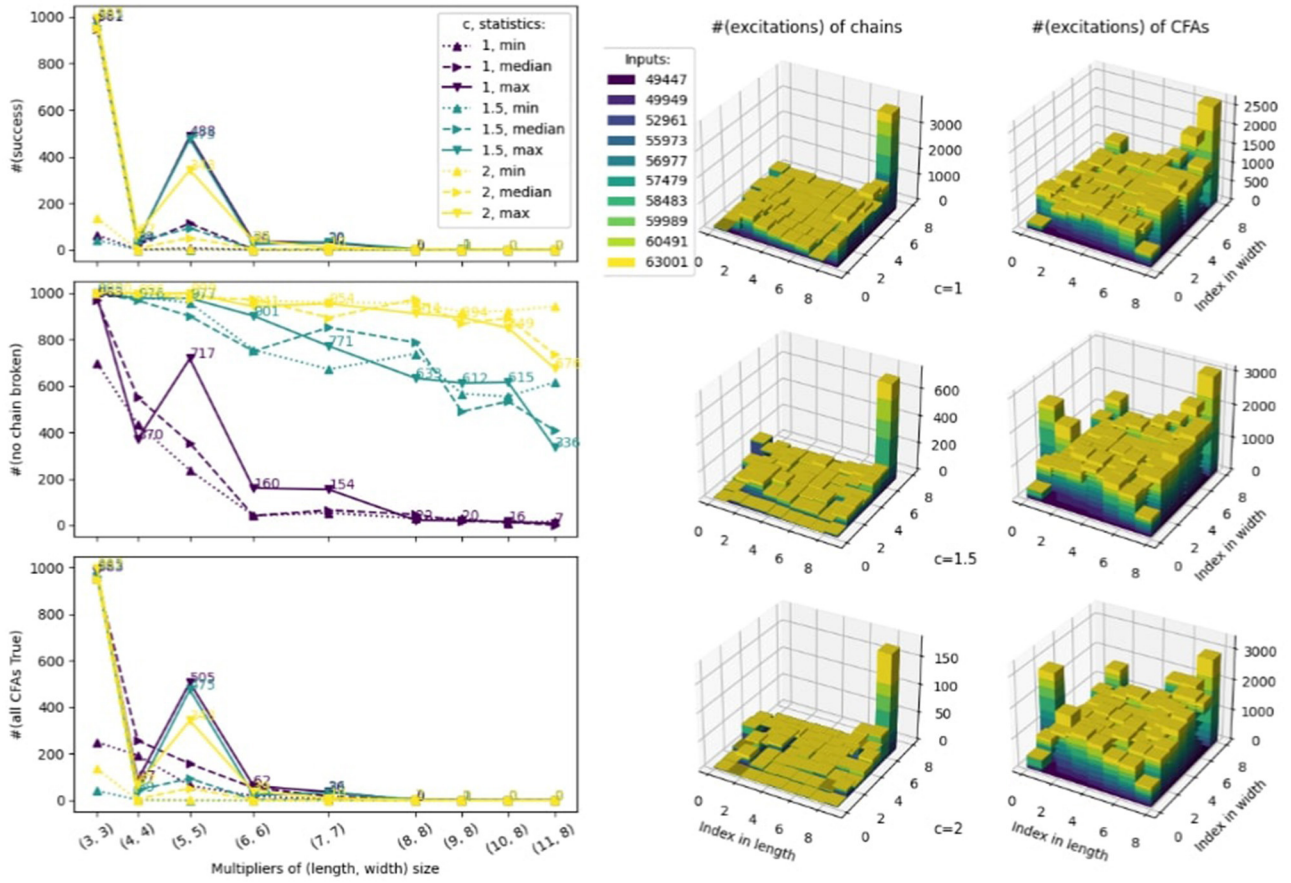


FIGURE 1
(Left) Comparison of different chain strengths $c, c \in \{1, 1.5, 2\}$, for QA to factor integers of $3 + 3$ bits up to $11 + 8$ bits, with the annealing time $T_a = 10\ \mu s$ and 1,000 samples set for each problem instance. **(Right)** Excitations distribution of chains (first column) and CFAs (second column) for factoring 10 integers of $8 + 8$ bits tested in the previous experiments, with chain strength equal to $c \in \{1, 1.5, 2\}$ (respectively top, middle, and bottom row).

least frequently (represented by the minimum dotted line in Figure 1, left), as well as the one that reaches the ground state the most frequently (represented by the maximum solid line in Figure 1, left).

We see that stronger chains ($c \in \{1.5, 2\}$) do not always bring us a higher success probability in general for the chosen problem sizes, and that weaker chains ($c = 1$) can produce higher success probabilities than stronger chains occasionally for middle-size problems. Notice that this result, in terms of the success probability, is consistent with what is mentioned by Lanting et al. (2017), suggesting that locally-structured embedding does not behave differently from global embedding regarding chain strengths. We also observe that as the problem size increases, weaker chains tend to be broken more easily than stronger chains. The rapidly declining dotted yellow lines confirm this phenomenon, approaching 0 for problems of bigger size. Based on these two observations, we speculate that the strongest chain, which was chosen in Ding et al. (2024), is the best candidate for factoring integers of up to 17×8 bits, the maximal problem size they could encode into the target QA system with a locally-structured embedding.

3.3 Results of incrementally fixing excited CFAs

From the experiment of the previous subsection, we can see that there seems to be a trade-off between broken chains and the excitations of CFAs: the weaker the chains are, the more likely they are broken, and the fewer the samples where the CFAs are excited. Moreover, the excitations of CFAs are not uniformly distributed. To this extent, we studied the distribution of broken chains and CFAs in 10 8×8 factoring problems, shown in Figure 1, right. The results on excitations of chains and CFAs are reported as 3D bar plots in Figure (3rd and 4th row, respectively). Each problem instance is mapped with its color. The x and the y axis correspond to the column and row of the multiplier respectively; the z axis represents the sum of excitations of each chain or CFA for the tested 10 problem instances. These results support testing an incremental remedy strategy based on modules.

With the strongest chain strength and the same configuration of the annealing system as the other experiments in the paper, we test the approach of incrementally fixing excited CFAs for QA factoring the highest integers of 8×8 bits up to 10×8 bits from the experiments shown in Figure 1. The results are shown in Table 2.

TABLE 2 Results of incrementally remedying excited CFAs for factoring integers of 8 × 8 bits up to 10 × 8 bits, with the same annealing time $T_a = 10\ \mu s$ with the number of samples ranging from 1,000 to 3,000 set for each problem instance.

Size	#Samples	1000					2000					3000				
	input	P_F	(CFA, #excs)	i	P_F	(CFA, #excs)	P_F	(CFA, #excs)	i	P_F	(CFA, #excs)	P_F	(CFA, #excs)	i	P_F	(CFA, #excs)
8×8	49,447 = 251×197	6.25	[(6, 7), 395]	5	**0.0**	[(0, 1), 315]	6.0	[(0, 1), 1,268]	32	4.083	[(6, 5), 490]	4.0	[(0, 1), 1,377]	32	4.083	[(4, 7), 894]
	49,949 = 251×199	6.083	[(6, 7), 466]	4	**0.0**	[(7, 6), 347]	4.0	[(0, 1), 982]	32	4.0	[(6, 5), 429]	2.0	[(0, 1), 1,461]	25	**0.0**	[(5, 7), 631]
	52,961 = 251×211	2.083	[(7, 5), 454]	32	6.083	[(6, 7), 250]	6.167	[(7, 5), 994]	32	6.167	[(5, 7), 452]	6.083	[(7, 5), 1,705]	6	**0.0**	[(7, 6), 1,001]
	55,973 = 251×223	4.0	[(5, 7), 242]	2	**0.0**	[(7, 3), 267]	**0.0**	[(0, 1), 569]	0	**0.0**	[(0, 1), 569]	**0.0**	[(7, 6), 896]	0	**0.0**	[(7, 6), 896]
	56,977 = 251×227	2.083	[(7, 6), 457]	31	**0.0**	[(0, 1), 351]	4.083	[(7, 6), 921]	8	**0.0**	[(7, 6), 593]	4.083	[(7, 6), 1,555]	16	**0.0**	[(0, 1), 739]
	57,479 = 251×229	6.0	[(5, 7), 277]	32	4.0	[(6, 5), 200]	4.083	[(0, 1), 722]	32	4.083	[(6, 6), 471]	4.0	[(5, 7), 925]	1	**0.0**	[(0, 1), 905]
	58,483 = 251×233	4.083	[(7, 7), 338]	32	4.0	[(0, 3), 242]	4.083	[(7, 7), 779]	32	4.083	[(1, 4), 452]	4.0	[(7, 7), 1,069]	32	4.0	[(2, 1), 669]
	59,989 = 251×239	**0.0**	[(7, 7), 252]	0	**0.0**	[(7, 7), 252]	**0.0**	[(7, 7), 815]	0	**0.0**	[(7, 7), 815]	**0.0**	[(7, 7), 1,282]	0	**0.0**	[(7, 7), 1,282]
	60,491 = 251×241	2.0	[(7, 7), 237]	32	4.083	[(7, 7), 276]	2.0	[(7, 7), 856]	32	2.0	[(1, 4), 461]	2.0	[(7, 7), 1,082]	32	2.0	[(3, 0), 589]
	63,001 = 251×251	4.083	[(7, 7), 492]	4	**0.0**	[(0, 2), 292]	2.0	[(7, 7), 836]	1	**0.0**	[(7, 7), 889]	2.0	[(7, 7), 1,397]	6	**0.0**	[(7, 7), 999]
9×8	100,273 = 509×197	8.167	[(7, 4), 629]	34	4.083	[(6, 7), 281]	8.083	[(7, 4), 1,413]	34	8.0	[(1, 7), 490]	4.083	[(7, 4), 1,834]	34	4.083	[(1, 7), 754]
	101,291 = 509×199	8.0	[(7, 3), 461]	34	6.25	[(6, 8), 288]	6.083	[(7, 3), 859]	34	8.0	[(6, 8), 540]	8.083	[(6, 8), 1,273]	34	6.083	[(5, 8), 692]
	107,399 = 509×211	8.0	[(7, 3), 479]	34	4.0	[(7, 5), 210]	4.083	[(0, 1), 1,100]	34	6.083	[(7, 6), 431]	6.0	[(7, 3), 1,485]	34	4.0	[(1, 4), 701]
	113,507 = 509×223	8.0	[(0, 1), 373]	13	**0.0**	[(7, 6), 244]	4.083	[(0, 1), 1,133]	3	**0.0**	[(7, 7), 995]	4.083	[(0, 1), 1,803]	34	6.0	[(6, 7), 612]
	115,543 = 509×227	8.083	[(0, 1), 541]	34	8.0	[(7, 3), 214]	8.0	[(0, 1), 1,394]	34	6.167	[(7, 6), 460]	6.0	[(0, 1), 1,633]	34	6.083	[(0, 0), 794)
	116,561 = 509×229	6.167	[(0, 1), 434]	34	8.167	[(2, 5), 226]	6.0	[(0, 1), 1,002]	34	6.083	[(6, 7), 522]	6.083	[(7, 8), 1,305]	34	6.083	[(7, 8), 660)
	118,597 = 509×233	6.0	[(0, 1), 379]	34	4.0	[(2, 6), 211]	8.0	[(7, 8), 880]	34	6.167	[(6, 7), 363]	6.083	[(7, 8), 1,743]	34	6.083	[(5, 8), 683)
	121,651 = 509×239	8.083	[(7, 8), 628]	34	4.083	[(7, 7), 274]	8.0	[(7, 8), 1,035]	9	**0.0**	[(0, 2), 508]	4.0	[(0, 1), 1,307]	4	**0.0**	[(0, 2), 974]
	122,669 = 509×241	6.083	[(7, 8), 600]	34	4.083	[(7, 6), 272]	10.0	[(7, 8), 1,515]	34	4.0	[(0, 4), 557]	6.083	[(7, 8), 2,154]	34	4.0	[(7, 5), 685]
	127,759 = 509×251	6.0	[(0, 1), 651]	2	**0.0**	[(0, 1), 542]	6.0	[(7, 8), 1,261]	2	**0.0**	([7, 8), 1,026]	4.0	[(0, 1), 1,799]	2	**0.0**	[(0, 1), 1,837]

(Continued)

10.3389/fcomp.2024.1335369

TABLE 2 (Continued)

Size	#Samples	1000					2000					3000				
	input	P_F	(CFA, #excs)	i	P_F	(CFA, #excs)	P_F	(CFA, #excs)	i	P_F	(CFA, #excs)	P_F	(CFA, #excs)	i	P_F	(CFA, #excs)
10×8	201,137 = 1,021×197	8.083	[(7, 2), 424]	36	10.083	[(7, 2), 234]	2.0	[(7, 2), 881]	36	4.083	[(4, 9), 463]	4.0	[(7, 2), 1,430]	36	6.083	[(2, 9), 690]
	203,179 = 1,021×199	10.083	[(0, 1), 474]	36	6.083	[(0, 2), 281]	8.083	[(0, 1), 940]	36	6.167	[(5, 7), 423]	8.083	[(0, 1), 1,431]	36	8.0	[(7, 6), 1,033]
	215,431 = 1,021×211	8.0	[(0, 1), 574]	36	8.0	[(7, 8), 265]	6.0	[(0, 1), 1,033]	36	6.083	[(3, 1), 422]	6.0	[(0, 1), 1,817]	36	8.0	[(7, 3), 594]
	227,683 = 1,021×223	8.083	[(0, 1), 586]	36	6.167	[(5, 9), 213]	4.0	[(0, 1), 1,318]	36	4.167	[(6, 8), 419]	6.0	[(0, 1), 1,897]	36	4.083	[(7, 4), 639]
	231,767 = 1,021×227	10.0	[(0, 1), 592]	36	8.083	[(5, 9), 269]	8.083	[(0, 1), 1,146]	36	6.083	[(7, 7), 452]	8.083	[(0, 1), 1,709]	36	6.0	[(7, 6), 641]
	233,809 = 1,021×229	8.083	[(7, 9), 361]	36	6.167	[(5, 9), 248]	6.0	[(0, 1), 922]	36	8.0	[(2, 9), 453]	6.167	[(7, 9), 1,207]	36	6.0	[(7, 5), 776]
	237,893 = 1,021×233	6.0	[(7, 9), 456]	36	6.083	[(0, 1), 185]	6.0	[(7, 9), 886]	36	6.0	[(1, 4), 378]	4.0	[(7, 9), 1,480]	36	6.0	[(2, 2), 553]
	244,019 = 1,021×239	8.083	[(0, 1), 600]	36	4.0	[(7, 9), 234]	6.167	[(0, 1), 1,252]	36	6.0	[(1, 2), 427]	6.083	[(7, 9), 1,595]	30	**0.0**	[(0, 1), 733]
	246,061 = 1,021×241	2.083	[(7, 9), 619]	36	6.0	[(1, 1), 232]	10.0	[(7, 9), 1,056]	36	8.0	[(1, 5), 499]	8.0	[(7, 9), 1,478]	36	4.0	[(2, 9), 615]
	256,271 = 1,021×251	4.083	[(7, 9), 659]	36	4.083	[(7, 8), 226]	6.083	[(0, 1), 1,256]	10	**0.0**	[(0, 4), 695]	2.083	[(0, 1), 1,900]	36	4.083	[(7, 8), 787]

For each problem, we first report the starting point sample, including its energy, the most excited CFA, and the number of its excitations respectively. Then, we report the number of iterations performed by the remedy strategy (a bold number means we did not reach the step threshold and a ground state has been found), together with the energy and the current most excited CFA.

10.3389/fcomp.2024.1335369

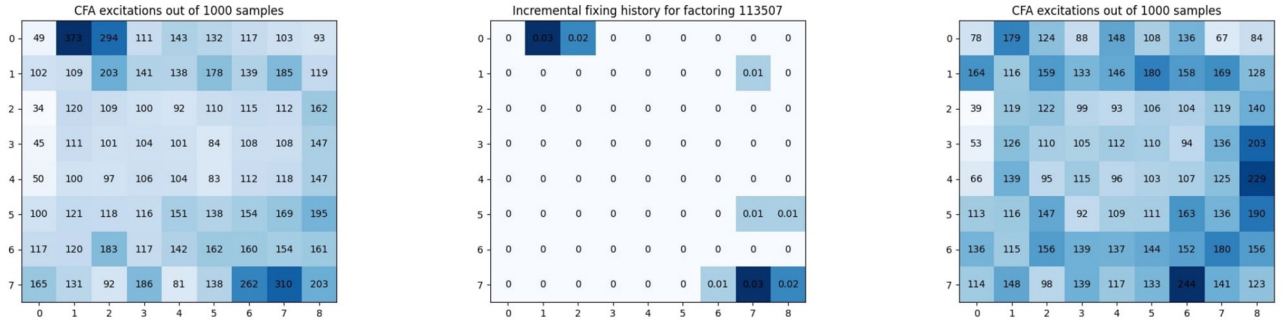


FIGURE 2
Fixing results for factoring 16-bit 113,507 in Table 2. Each coordinate corresponds to the CFA in the multiplier circuit. The numbers in the figures in the leftmost and rightmost columns represent the number of excitations out of 1,000 samples, whereas the number in the figure in the middle denotes the anneal offset used in the whole fixing process, for advancing the annealing schedule of the specific CFA. Notice the less homogeneous distribution of excitations of CFAs in the leftmost figure compared to the rightmost figure. **(Left)** Before the fixing. **(Middle)** The fixing history. **(Right)** After the fixing.

To get an extensive analysis of the novel remedy strategy, we tested three different configurations, with the only difference being the number of samples obtained for each iteration (respectively 1,000, 2,000, and 3,000). We also show in Figure 2 the behavior of the remedy strategy on one of the problem instances.

From the results, we can see that the remedy strategy helps in solving some of the problem instances. In particular, this approach works under the assumption the user has a limited amount of QPU time (i.e., the annealing time is confined to values $\lesssim 20(\mu)s$, showing its effectiveness when users are bound to tight constraints in accessing the D-Wave devices. This approach works more effectively with smaller instances, reaching the ground state more frequently and with fewer iteration steps. Moreover, increasing the sample size does not impact performances, showing sporadically improvements in reaching the ground state when the number of samples increases. Nevertheless, setting the annealing offset scores based on the modules' properties instead of targeting qubits independently seems promising, and further investigations could define different conditions to prioritize the annealing of some CFAs.

4 Discussion

This paper has built upon the recent work presented in our previous publication (Ding et al., 2024), which introduced a novel approach to the problem of PF through quantum annealing. In contrast to our previous paper, which showcased exclusively the effective techniques that highly benefited our task, here we discussed several intermediate and less successful approaches. This comprehensive exploration provides insights into the intricacies that influenced our final results in Ding et al. (2024). The code to replicate these experiments is reported in the following publicly available repository: https://gitlab.com/jingwen.ding/multiplier-encoder-2nd.

Our experiments revealed several insights:

- **Effectiveness of flux biases tuning:** We showed that the techniques to initialize qubits implemented at the encoding level were not as effective as flux-biases tuning. Nevertheless, they can be considered as viable alternative to the usage of fix_variables() in other contexts.
- **Chains coupling strength:** Even though using the highest value for chains coupling strength might not be optimal for small-sized problems, it proved crucial for solving more complex problems. This highlights the delicate balance between problem size and annealing parameters, e.g., chain strength.
- **Trade-off between broken chains and CFA excitations:** We observed a trade-off between the presence of broken chains and the excitations of CFAs when the QA generates its samples. This further highlights the importance of monitoring chain strength in other contexts.
- **Non-uniform distribution of CFA excitations:** The excitations of CFAs were found to be non-uniformly distributed for different samples on the same problem instance. Understanding this distribution can be valuable for tailoring annealing strategies to specific problem instances.
- **Remedy strategy for middle-size problems:** The remedy strategy we proposed in Section 2.4, based on the above observations, showed minor benefits in solving middle-sized problems. Nevertheless, it could be useful in other contexts.

By delving into the details of our experimental journey, listing both our successes and setbacks, we aim to provide valuable insights to a more specialized audience of D-Wave Quantum Annealer users and practitioners. Our work contributes to the evolving world of quantum annealing and equips researchers and professionals with additional knowledge to effectively use D-Wave quantum annealers in their applications.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found at: https://gitlab.com/jingwen.ding/multiplier-encoder-2nd.

Author contributions

RS: Conceptualization, Formal analysis, Funding acquisition, Methodology, Project administration, Supervision, Writing – original draft, Writing – review & editing. JD: Data curation, Investigation, Software, Writing – original draft, Writing – review & editing. GS: Data curation, Validation, Writing – original draft, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work was funded by the European Union. Grant Agreement No. 101120763—TANGO and supported by Q@TN, the joint lab between University of Trento, FBK—Fondazione Bruno Kessler, INFN—National Institute for Nuclear Physics, and CNR—National Research Council. We acknowledge the support from CINECA in terms of QUPU-time availability via the project WCRI-QCSC. We acknowledge the previous support from D-Wave Systems Inc, in terms of both financial support and QUPU-time availability, via the former project QuASI. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article, or the decision to submit it for publication. This research was supported in part by TAILOR, a project funded by the EU Horizon 2020 research and innovation program under GA No. 952215. The work was partially supported by the project: AI@TN funded by the Autonomous Province of Trento. We acknowledge the support of the MUR PNRR project FAIR—Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

Author disclaimer

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HaDEA). Neither the European Union nor the granting authority can be held responsible for them.

References

Adame, J. I., and McMahon, P. L. (2020). Inhomogeneous driving in quantum annealers can result in orders-of-magnitude improvements in performance. *Quant. Sci. Technol.* 5:e035011. doi: 10.1088/2058-9565/ab935a

Amico, M., Saleem, Z. H., and Kumph, M. (2019). Experimental study of Shor's factoring algorithm using the IBM Q experience. *Phys. Rev. A* 100:e012305. doi: 10.1103/PhysRevA.100.012305

Andriyash, E., Bian, Z., Chudak, F., Drew-Brook, M., King, A. D., Macready, W. G., et al. (2016). *Boosting Integer Factoring Performance via Quantum Annealing Offsets. D-Wave Technical Report Series*, 14.

Bian, Z., Chudak, F., Macready, W., Roy, A., Sebastiani, R., and Varotti, S. (2020). Solving SAT (and MaxSAT) with a quantum annealer: foundations, encodings, and preliminary results. *Inform. Comput.* 275:104609. doi: 10.1016/j.ic.2020.104609

Ding, J., Spallitta, G., and Sebastiani, R. (2024). Effective prime factorization via quantum annealing by modular locally-structured embedding. *Sci. Rep.* 14:3518. doi: 10.1038/s41598-024-53708-7

Dridi, R., and Alghassi, H. (2017). Prime factorization using quantum annealing and computational algebraic geometry. *Sci. Rep.* 7:43048. doi: 10.1038/srep43048

DWave (2021). *Anneal Offsets*. Available online at: https://docs.dwavesys.com/docs/latest/c_qpu_annealing.html (accessed October 25, 2023).

Jiang, S., Britt, K. A., McCaskey, A. J., Humble, T. S., and Kais, S. (2018). Quantum annealing for prime factorization. *Sci. Rep.* 8:17667. doi: 10.1038/s41598-018-36058-z

Karamlou, A., Simon, W., and Katabarwa, A. E. A. (2021). Analyzing the performance of variational quantum factoring on a superconducting quantum processor. *Quant. Inf.* 7:478. doi: 10.1038/s41534-021-00478-z

Lanting, T., King, A. D., Evert, B., and Hoskinson, E. (2017). Experimental demonstration of perturbative anticrossing mitigation using nonuniform driver hamiltonians. *Phys. Rev. A* 96:e042322. doi: 10.1103/PhysRevA.96.042322

Lucero, E., Barends, R., Chen, Y., Kelly, J., Mariantoni, M., Megrant, A., et al. (2012). Computing prime factors with a josephson phase qubit quantum processor. *Nat. Phys.* 8, 719–723. doi: 10.1038/nphys2385

Martín-López, E., Laing, A., Lawson, T., Alvarez, R., Zhou, X. Q., and O'Brien, J. L. (2012). Experimental realization of Shor's quantum factoring algorithm using qubit recycling. *Nat. Photon.* 6, 773–776. doi: 10.1038/nphoton.2012.259

Mengoni, R., Ottaviani, D., and Iorio, P. (2020). *Breaking RSA Security With a Low Noise D-Wave 2000Q Quantum Annealer: Computational Times, Limitations and Prospects.*

Monz, T., Nigg, D., Martinez, E. A., Brandl, M. F., Schindler, P., Rines, R., et al. (2016). Realization of a scalable Shor algorithm. *Science* 351, 1068–1070. doi: 10.1126/science.aad9480

Rivest, R. L., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* 21, 120–126.

Sebastiani, R., and Trentin, P. (2020). OptiMathSAT: a tool for optimization modulo theories. *J. Automat. Reason.* 64, 423–460. doi: 10.1007/s10817-018-09508-6

Selvarajan, R., Dixit, V., and Cui, X. E. A. (2021). Prime factorization using quantum variational imaginary time evolution. *Sci. Rep.* 11:20835. doi: 10.1038/s41598-021-00339-x

Vandersypen, L. M. K., Steffen, M., Breyta, G., Yannoni, C. S., Sherwood, M. H., and Chuang, I. L. (2001). Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature* 414, 883–887. doi: 10.1038/414883a

Wang, B., Hu, F., and Yao, H. (2020). Prime factorization algorithm based on parameter optimization of ising model. *Sci. Rep.* 10:2020. doi: 10.1038/s41598-020-62802-5

Willsch, D., Willsch, M., Jin, F., De Raedt, H., and Michielsen, K. (2023). Large-scale simulation of Shor's quantum factoring algorithm. *Mathematics* 11:4222. doi: 10.3390/math11194222

Yarkoni, S., Wang, H., Plaat, A., and Bäck, T. (2019). "Boosting quantum annealing performance using evolution strategies for annealing offsets tuning," in *Quantum Technology and Optimization Problems: First International Workshop, QTOP 2019, Munich, Germany, March 18, 2019, Proceedings 1* (Berlin: Springer), 157–168.

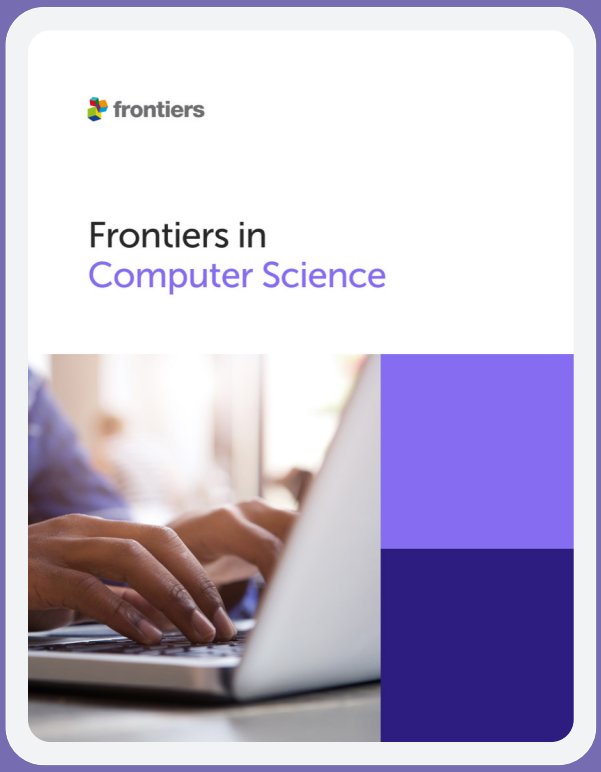
Frontiers in Computer Science

Explores fundamental and applied computer science to advance our understanding of the digital era

An innovative journal that fosters interdisciplinary research within computational sciences and explores the application of computer science in other research domains.

Discover the latest Research Topics

See more →



Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact