



Universiteit
Leiden

The Netherlands

Quantum annealing for industry applications: introduction and review

Yarkoni, S.; Raponi, E.; Bäck, T.H.W.; Schmitt, S.

Citation

Yarkoni, S., Raponi, E., Bäck, T. H. W., & Schmitt, S. (2022). Quantum annealing for industry applications: introduction and review. *Reports On Progress In Physics*, 85(10). doi:10.1088/1361-6633/ac8c54

Version: Publisher's Version

License: [Licensed under Article 25fa Copyright Act/Law \(Amendment Taverne\)](#)

Downloaded from: <https://hdl.handle.net/1887/3504986>

Note: To cite this publication please use the final published version (if applicable).

Quantum Annealing for Industry Applications: Introduction and Review

SHEIR YARKONI, Leiden University, Netherlands

ELENA RAPONI, Leiden University, Netherlands

SEBASTIAN SCHMITT, Honda Research Institute Europe GmbH, Germany

THOMAS BÄCK, Leiden University, Netherlands

Quantum annealing is a heuristic quantum optimization algorithm that can be used to solve combinatorial optimization problems. In recent years, advances in quantum technologies have enabled the development of small- and intermediate-scale quantum processors that implement the quantum annealing algorithm for programmable use. Specifically, quantum annealing processors produced by D-Wave Systems have been studied and tested extensively in both research and industrial settings across different disciplines. In this paper we provide a literature review of the theoretical motivations for quantum annealing as a heuristic quantum optimization algorithm, the software and hardware that is required to use such quantum processors, and the state-of-the-art applications and proofs-of-concepts that have been demonstrated using them. The goal of our review is to provide a centralized and condensed source regarding applications of quantum annealing technology. We identify the advantages, limitations, and potential of quantum annealing for both researchers and practitioners from various fields.

1 INTRODUCTION

Quantum computing (QC) is a rapidly growing research field that promises a novel paradigm to solve challenging computational problems. It was first introduced in the early 1980s by physicist Paul Benioff [13, 14] and independently by Richard Feynman [56]. These original proposals of quantum computers refer to the use of quantum phenomena such as superposition and entanglement to perform computation and simulate nature. In the decades since, significant advancement of both algorithms and hardware has been made that expanded the potential and scope of use of quantum computers. Seminal works from Deutsch [44] and Grover [62] showed that quantum algorithms can provide an asymptotic speedup relative to their classical counterparts. In some cases this can even result in exponentially faster algorithms, such as Shor’s algorithm for factoring [150] and the quantum Fourier transform algorithm [37]. More recently, industrial efforts to build commercially-available QCs have emerged around the world, with vendors such as IBM [112], Google [11], Rigetti [141], D-Wave Systems [80], IonQ [42, 174] and others offering quantum computing services via cloud-based access.

There are many different paradigms and hardware implementations that can be used to build quantum computers. The two leading paradigms are gate-based quantum computing and adiabatic quantum computation (AQC) [6, 41, 54, 71, 113]. In the gate model of QC the computation is performed by applying a sequence of unitary gates to a set of quantum bits (qubits), whose states can be measured at the end of the computation [44]. In contrast, in AQC one prepares an initial many-qubit quantum state as the ground state of a simple Hamiltonian, to which an adiabatic time evolution is applied, changing the system to a final Hamiltonian whose ground state encodes the solution of the desired (optimization) problem. In principle, AQC has been shown to be polynomially equivalent to gate-based universal QC, as any quantum circuit can be represented as a time-dependent Hamiltonian with at most polynomial overhead [5, 17].

Authors’ addresses: Sheir Yarkoni, Leiden University, Leiden, Netherlands, s.yarkoni@liacs.leidenuniv.nl; Elena Raponi, e.raponi@liacs.leidenuniv.nl, Leiden University, Leiden, Netherlands; Sebastian Schmitt, Honda Research Institute Europe GmbH, Germany, Sebastian.Schmitt@honda-ri.de; Thomas Bäck, Leiden University, Leiden, Netherlands, t.h.w.baeck@liacs.leidenuniv.nl.

There are many possible technologies one could use to build functioning qubits, and with them quantum computers. The first such prototypes were based on nuclear magnetic resonance (NMR) technology as early as the 1990's [35, 81]. Since then, other technologies have shown some promise in implementing QCs such as trapped ions [42, 136], cold atoms [83, 148, 170], photon-based QCs [167], Rydberg atoms [148], and diamond cavities [47]. However, the most promising and (to date) scalable QCs use superconducting qubits. The technical motivations behind each technology (and their respective strengths and weaknesses) is beyond the scope of this review, but we refer the interested reader to the relevant citations above.

An approach closely related to AQC is quantum annealing (QA) [41, 71, 109, 113, 158], proposed in 1988 by Apolloni et al. [8]. QA can be viewed as a relaxation of the AQC model, where the conditions of adiabaticity are not met, resulting in a heuristic variational quantum algorithm. Therefore, this approach can be used to find the ground state of Ising models [108], a known NP-hard task [12]. It is well-established in literature how to transform canonical NP-hard and NP-complete combinatorial optimization problems to forms suitable for quantum annealers [100, 101]. These problems can be posed either in Ising form using a $\{-1, 1\}$ basis and spin variables, or as a quadratic unconstrained binary optimization (QUBO) problem using the $\{0, 1\}$ basis and binary variables. The two forms are equivalent and problems in one form can be easily represented in the other using a simple change of basis.

As such, QA lends itself naturally to practitioners interested in using quantum algorithms to solve combinatorial optimization problems. While the question of whether QA can provide a quantum speed-up over classical approaches is still subject of a scientific debate (see, for example, [71, 109, 145] and references therein), there is ongoing dedicated effort to use QA hardware to solve application-motivated problems. Numerous examples have been published of the use of QA to solve real-world problems in a variety of domains, particularly in the fields of optimization, scheduling, machine learning, and simulations of natural systems. Here, a similar conclusion has to be drawn as, to the best of our knowledge, there is no industrial application where QA unquestionably outperforms classical heuristic algorithms. However, the technological advances of recent years and the potential that may still be realized in the near future fuel the hope that QA hardware devices may be competitive in some domains in the future.

In QA, the solution of a problem is realized by going through several stages. The details of this process will be explained in the next sections but the general workflow is as follows (which is also visualized in Fig. 1):

- **Definition of a QUBO formulation and conversion into graph.** QUBO has become the standard input format for quantum annealers, to which the real-world application problems are converted. After definition, the QUBO is directly converted into a logical graph, where each node represents a variable and each edge denotes the interaction term between a pair of variables.
- **Minor-embedding.** When translating the logical interaction graph onto the physical hardware graph of the quantum processing unit (QPU), it is necessary to select sets of physical qubits to represent a single logical node and to identify the couplings between the physical qubits to realize the correct interactions between the logical variables.
- **Programming.** Programming the quantum annealer requires setting the parameters that define the problem to be solved, also called final Hamiltonian. This involves setting the weights for each qubit biases (to control the magnetic field acting on the qubit) and coupler strengths (to control the interaction between qubits).

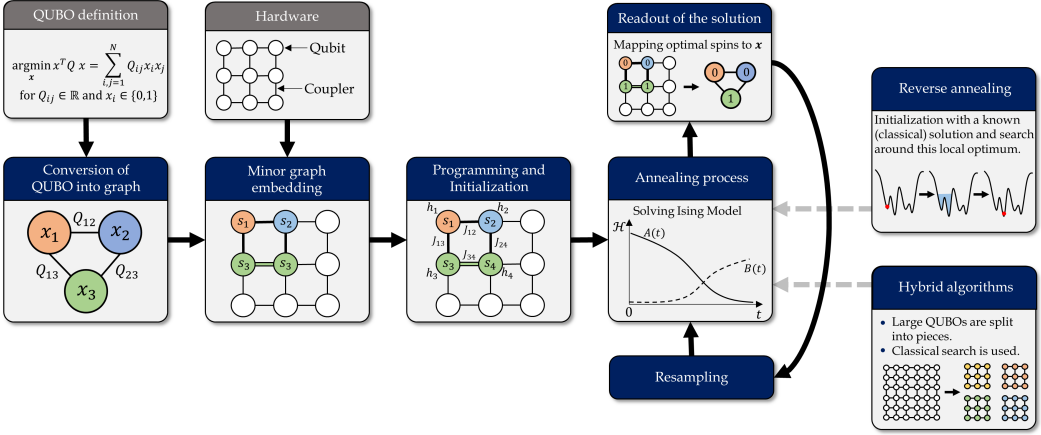


Fig. 1. Visualization of a typical quantum algorithm workflow on a quantum annealer. The image is an adapted and extended version of the workflow presented by Goodrich et al. [60].

- **Initialization.** After programming, the spin configuration of the QPU is initialized as the lowest energy configuration of an easy-to-implement initial Hamiltonian, where the qubits are placed in an equal superposition of all possible states.
- **Annealing process.** In this step, the Ising model is solved. The system transitions from the initial to the final Hamiltonian according to predefined annealing functions in an attempt to minimize energy. This step can also be the core of a hybrid approach where the inner loop of the calculation is performed by a quantum processor. Annealing can also be combined with a reverse annealing phase, which initializes the quantum annealer with a known (classical) solution and searches the state space around this local optimum.
- **Readout of the solution.** At the end of the annealing phase, the qubits are in classical spin states, and their configuration represents a minimum of the final Hamiltonian. The individual spin values of the final configuration are read out and stored externally representing a candidate solution to the original problem.
- **Resampling.** Quantum annealing is a heuristic, which means that there is only ever a non-zero probability that the computation results in a ground state of the system. Therefore, the anneal-readout cycle is repeated many times per input to acquire multiple candidate solutions.

In this work we give an overview of the current state-of-the-art industrial applications and proofs-of-concepts that have been demonstrated using QA hardware. We focus on industrial problems that have been addressed using QA, and also provide insights on the limitations imposed by utilizing these methods. We present results that mainly use D-Wave quantum annealers, which reflects the fact that these are currently the only commercially available large-scale quantum annealers. The purpose of this review is to give a guide to practitioners whether their problems might be addressable using QA technology. We discuss potential benefits and limitations regarding the type of problem, the manageable problem sizes, the type of constraints to be included and potentially other relevant aspects for practical applications. We highlight the most prominent examples in existing literature, and attempt to bridge the gap between the fundamental quantum physics governing QA and application development. The ultimate goal of this work is to concisely

present the most pertinent information required by newcomers to the field of quantum annealing to properly understand both the potential and limitations of QA for industry-related problems.

The rest of this paper is structured as follows. In Section 2 we introduce the foundations of quantum annealing. Starting from the central theoretical ideas, we go through the definition of QUBOs and explain how to embed constraints in QUBOs formulations. We illustrate how binary encoding schemes can be used to represent discrete and continuous variables, respectively. We introduce the hardware implementation of quantum processors, and describe how to embed the QUBO/Ising models into a physically realized QPU by minor-embedding. We address the problem of noise and discuss the mitigation strategies proposed so far. We describe currently available software and show how classical and quantum resources can leverage their complementary strengths by introducing the concept of hybrid quantum-classical algorithms. The role of time in quantum annealing is discussed in detail. Section 3 reviews a selected number of state-of-the-art applications and empirical studies involving QA. We focus on five relevant areas: traffic flow optimization, scheduling and logistics, quantum chemistry, physics and biology, quantum machine learning, and finance. We then elaborate on these ideas and provide some concluding remarks on the potential and limitations of QA for real-world applications. In Section 4, we summarize the key lessons learned at the end of our journey through quantum annealing. Finally, in Section 5, we draw our conclusions.

2 FOUNDATIONS OF QUANTUM ANNEALING

The purpose of this section is to briefly explain the physics behind adiabatic quantum computation (AQC) and quantum annealing (QA), with sufficient detail to give a newcomer to the field a basic understanding of the foundations of the approach. For a more in-depth analysis of the physics behind QA, we refer the reader to Refs. [6, 71, 158].

2.1 Theoretical background

The motivation for QA is found in the *adiabatic theorem* [22, 113], which states the following: a quantum system in its ground state will remain in the ground state, provided the Hamiltonian governing the dynamics changes sufficiently slowly. This condition of “sufficiently slowly” is dictated by the smallest energy difference between the ground state and the first excited state exhibited during the system’s evolution (also known as the minimum gap). This can be utilized for computation by preparing a system in the ground state of an easy to solve initial Hamiltonian and then switching the Hamiltonian sufficiently slow to a complex Hamiltonian. By the adiabatic theorem, the system will remain in the ground state and consequently one ends up with the ground state, i.e., the solution of the complicated Hamiltonian. Applying this to Ising Hamiltonians, where it is NP-hard to find the minimal energy configuration, one can theoretically use such a quantum approach to solve problems in NP (but not necessarily in polynomial time). Furthermore, the adiabatic theorem can be used to simulate any quantum circuit, making it a universal quantum computing paradigm [5]. This model of quantum computing is known as adiabatic quantum computing (AQC) or adiabatic quantum optimization (AQO).

The quantum algorithm is started by preparing the system in the ground state of a Hamiltonian, which is known and easy to prepare, denoted as \mathcal{H}_i (also known as the *initial Hamiltonian*). Then, the system is slowly changed such that the contribution of \mathcal{H}_i is slowly reduced while the magnitude of a final (also known as *target*) Hamiltonian, denoted \mathcal{H}_f is increased (using time parameter t):

$$\mathcal{H}(t) = A(t)\mathcal{H}_i + B(t)\mathcal{H}_f, \quad (1)$$

Time $t \in [0, T_a]$ and the annealing schedule is defined by the monotonic functions $A(t)$ and $B(t)$ with $A(t=0) = 1$, $B(t=0) = 0$ and $A(t=T_a) = 0$, $B(T_a) = 1$. The gradual transition from \mathcal{H}_i to \mathcal{H}_f

induces an evolution from the initial (ground) state $|\Psi(t=0)\rangle$ to the ground state $|\Psi(t=T_a)\rangle$ of the final Hamiltonian \mathcal{H}_f , where $|\Psi(t)\rangle$ represents the time-dependent wavefunction in the Dirac bra-ket notation (for details, see, for example, [71]). Typically, a transverse field in x -direction is used as the initial Hamiltonian,

$$\mathcal{H}_i = \sum_{i \in V} \sigma_i^x, \quad (2)$$

where σ_i^x is the x -Pauli matrix acting on the i -th qubit. σ^x induces flips of the σ^z -basis states as $\sigma^x|\pm\rangle = |\mp\rangle$ and its eigenstates are $|x\pm\rangle = \frac{1}{\sqrt{2}}(|+\rangle \pm |-\rangle)$, which are equal amplitude superpositions of the σ^z eigenstates. Thus, initially, the system is prepared in the ground state to \mathcal{H}_i , which is given by $|\Phi(t=0)\rangle = |x-, \dots, x-\rangle$. The final Hamiltonian is represented by:

$$\mathcal{H}_f = \sum_{i \in V} h_i \sigma_i^z + \sum_{(i,j) \in E} J_{ij} \sigma_i^z \sigma_j^z. \quad (3)$$

Here, V is the set of vertices of the graph $G(V, E)$ representing the lattice sites where qubits are located, E the set of edges of the graph connecting the qubits, $J_{ij} = J_{ji}$ are the symmetric interaction strength of the qubits i and j connected by an edge and h_i is the on-site energy (local field) of qubit i . For any given time t , the Hamiltonian $\mathcal{H}(t)$ of Eq. (1) is the transverse-field Ising Hamiltonian, which is well studied in literature [151]. As the magnitude of \mathcal{H}_i decreases, the quantum dynamics of the qubits slows down until at $t = T_a$ we obtain a purely classical system, where the final states of the qubits are measured. The classical Ising model is obtained by replacing the σ^z Pauli operators by spin variables (all other terms are the same as in Eq. (3)):

$$\mathcal{H}_{\text{Ising}} = \sum_i h_i s_i + \sum_{ij} J_{ij} s_i s_j. \quad (4)$$

Despite the theoretical guarantees of the adiabatic theorem, the condition of adiabaticity is difficult to maintain in practice. Specifically, the fact that open quantum systems are susceptible to background noise and thermal fluctuation means that the quantum system could be kicked out of its ground state by the environment. Furthermore, the condition of ‘‘sufficiently slowly’’ is hard to fulfil, as the allowed rate of change is determined by the low energy states and the gaps in the spectrum of $\mathcal{H}(t)$, which is not known *a priori*. QA can be seen as a relaxation of AQC, where the annealing schedule, i.e., the time to switch from \mathcal{H}_i to \mathcal{H}_f , is determined heuristically and thus the strict adiabatic conditions are not guaranteed. Therefore, instead of a deterministic quantum algorithm, we instead obtain a *heuristic optimization algorithm*, where a non-zero probability of remaining in the ground state is maintained during annealing, i.e., during the complete time evolution.

The resulting QA heuristic is rather similar in spirit to the well-known classical simulated annealing approach to optimization [87]. Simulated annealing exploits thermal fluctuations for the exploration of the search space. During the optimization procedure, the strength of the thermal fluctuations is gradually reduced and for sufficiently slow annealing schedule global optima can be found. QA can be viewed to implement a very similar logic but where the thermal fluctuations are replaced by quantum fluctuations. The term proportional to \mathcal{H}_i in Eq. (1) induces disorder with respect to the eigenstates of the Ising Hamiltonian in Eq. (3) and introduces energetic variations to the qubits during the annealing process. During the annealing, the strength of the quantum fluctuation, i.e., the weight of \mathcal{H}_i is gradually reduced. The hope that quantum mechanical effects lead to an advantage of QA over classical algorithms is related to the conjectured role of quantum superposition, tunneling, and entanglement [6]. The basic reasoning is that quantum superposition and tunneling allow for a direct transition between states even if there is a high energy barrier between them [140], which leads to the ability of a search algorithm to escape local minima by tunneling through energetic barriers. In simple experiments, it has been demonstrated rigorously

that superposition and entanglement are indeed present throughout the annealing process [95]. However, it is not clear how these properties translate to an important role in the optimization process in large ensembles of qubits (see, for example, Refs., [43, 71, 84, 117] and references therein for more detailed discussions).

Currently, state-of-the-art QA hardware can only implement a subset of the protocols necessary that allow for universal quantum computation. Nonetheless, it has been shown that allowing increased control of the annealing path can significantly improve the quality of results obtained while still in the absence of universality. Specifically, the introduction of *non-stoquastic*¹ Hamiltonian terms result in increased degrees of freedom during the computation, giving hope for increasing the power of QA implementations [133, 168]. Further evidence is given by the fact that it is a known hard problem to simulate such Hamiltonians classically, and is therefore a bottleneck in quantum Monte Carlo procedures [63, 105]. It has been shown that implementing these terms to the driver Hamiltonian \mathcal{H}_i can result in mitigation of prohibitive phase transitions at the avoided crossing [124]. This mitigation can result in exponential speedup of the quantum annealing algorithm with non-stoquastic \mathcal{H}_i [34]. However, the situation is currently not fully clear, as there are also indications that adding non-stoquastic terms to the Hamiltonian reduces the energy gap between the ground and first excited state, which makes QA less robust [39].

Another relatively novel protocol implemented in recent D-Wave QPUs is the *reverse annealing* procedure [40]. Contrary to regular (forward) annealing, the Hamiltonian starts as a purely classical one, with the transverse field turned off (i.e., \mathcal{H}_f being the only component). Then, each qubit's classical state can be set independently by the user as a starting point. The transverse field is then slowly raised, inducing a partial mixing of classical and quantum components to the Hamiltonian, and then lowered again (the relative magnitude of \mathcal{H}_i is a free parameter). It has been shown that by using this novel protocol, reverse annealing can be useful in ways that strictly forward annealing cannot. For instance, one could seed a classical candidate solution to \mathcal{H}_f and use reverse annealing to perform semi-local neighborhood search [40]. Or alternatively, using the reverse annealing as the Markovian step in Markov chain Monte Carlo [85], or as a mutation operator in genetic algorithms [86]. It has also been shown in multiple applications that reverse annealing may yield solutions that strictly forward annealing cannot, due to a fundamentally different annealing path being used [59, 61, 74, 164].

From a practitioner's standpoint, QA can overall be viewed as a black-box heuristic optimization algorithm that attempts to find the minimum of a classical Ising spin system. In the Ising formulation, optimization problems are transformed to the spin model in Eq. (4). It has been shown that finding the ground state configuration of this spin system is in the worst case NP-hard [12], and is therefore of practical interest to the field of optimization. Therefore, the role of the practitioner is to find a suitable Ising representation of the optimization problem of interest, which is then used as the final Hamiltonian in the QA algorithm, Eq. (3).

2.2 Quadratic unconstrained binary optimization formulation

The Ising model can be easily transformed into various other forms, such as the unconstrained binary quadratic program (UBQP) [57], the undirected graphical model [102], and the quadratic unconstrained binary optimization (QUBO) problem [101]. Most commonly, either the QUBO or Ising model is used to construct problems that can be solved by QA. The QUBO form is equivalent to the Ising model through a simple change of basis ($s \mapsto 2x - 1$). In this representation, the goal is

¹A Hamiltonian is called stoquastic in a given basis if it only has real and non-positive off-diagonal matrix elements. Therefore, a non-stoquastic Hamiltonian has positive or complex-valued off-diagonal elements. Stoquastic Hamiltonians can be simulated classically as no sign problem occurs.

to find the configurations of N binary variables $x_i \in \{0, 1\}$, $i = 1, \dots, N$ that minimize the objective function

$$\text{Obj}(\mathbf{x}, Q) = \mathbf{x}^T \cdot Q \cdot \mathbf{x} = \sum_{i,j=1}^N Q_{ij} x_i x_j, \quad (5)$$

where \mathbf{x} is the vector of the N binary variables and Q is the symmetric matrix defining interaction terms between the variables.

Problems that incorporate higher-order interactions, i.e., k -local terms with $k \geq 3$, can be reduced to a QUBO form by introducing ancillary variables (see, e.g., Ref. [71]), but this incurs a polynomial overhead in the number of variables leading to an undesirable scaling of the problem size [77, 162].

2.3 Constrained optimization problems

So far, we have discussed the use of quantum annealing purely in the context of unconstrained binary optimization. However, real-world applications typically involve other, more general forms of optimization problems, which may also be addressed using quantum annealing, albeit not natively. We review some of the prominent classes of non-binary and constrained optimization and their respective implementation, along with some relevant literature.

It is important to note that constraints on the binary variables cannot be directly enforced in the QUBO/Ising formulation. Once the Hamiltonian is programmed, it is immutable and the annealing procedure determines some minimal energy configuration, where conditions or constraints on the binary variables cannot be applied. As a consequence, all constraints must therefore be enforced by contributions to the cost function in the Hamiltonian. A common approach to solving this problem is by implementing penalty terms with large penalty factors.

The inclusion of linear equality constraints with integer coefficients is straight forward. Consider, for example, a simple one-hot constraint over three binary variables (x_1, x_2, x_3) , where exactly one variable must be set to one while the others need to be zero, i.e.,

$$x_1 + x_2 + x_3 = 1. \quad (6)$$

To ensure that such a constraint is obeyed by our QUBO formulation, we can use the following equivalent quadratic representation of the constraint,

$$\lambda (x_1 + x_2 + x_3 - 1)^2 = 0, \quad (7)$$

and add a corresponding quadratic term $\lambda (x_1 + x_2 + x_3 - 1)^2$ to the cost function. The minimum of this contribution is zero only when exactly one of the variables is 1 and the rest are 0, and it is larger than zero for any other configuration. Therefore, when adding such a term to the QUBO cost function, the contribution is vanishing only when the one-hot constraint is not violated. In this fashion, the inclusion of linear equality constraints $P(\mathbf{x}) = \sum_i a_i x_i - b = 0$ is realizable in a straight forward fashion by adding a penalty term

$$\lambda \left(\sum_i a_i x_i - b \right)^2 \equiv \lambda P(\mathbf{x})^2 \quad (8)$$

to the cost function.

Similarly, problems with linear inequality constraints can also be expressed as penalty terms, with the addition of auxiliary slack variables. Consider the following inequality for three binary variables (x_1, x_2, x_3) :

$$x_1 + x_2 + x_3 \leq 2. \quad (9)$$

As before, we must transform this inequality to its suitable QUBO form. We start by adding auxiliary variables such that the total slack is accounted for: in this case, two additional binary variables (y_1, y_2) are sufficient. We notice that the equality constraint

$$x_1 + x_2 + x_3 = 2 - y_1 - y_2, \quad (10)$$

is equivalent to the original inequality constraint, since it can be fulfilled for all values of the binary variables x_i satisfying the original inequality by choosing some appropriate values for the slack variables. For values violating the original inequality constraint, the equation can never be fulfilled regardless of the values of the slack variables. The equality constraint can be written in the equivalent quadratic form and added as a penalty term to the QUBO cost function,

$$\lambda (x_1 + x_2 + x_3 + y_1 + y_2 - 2)^2. \quad (11)$$

More general linear inequality constraints $P(\mathbf{x}) = \sum_i a_i x_i - b \leq 0$ with positive integer coefficients $a_i, b \in \mathbb{N}_0^+$ can be handled in this fashion by adding a penalty term

$$\lambda \left(\sum_i a_i x_i + \sum_j^W w_j y_j - b \right)^2 \equiv \lambda P(\mathbf{x})^2 \quad (12)$$

to the cost function. The number of slack variables W and their coefficient w_j can be derived from the coefficients a_i and b , where at most $W = b$ slack variables are needed. The more general case with arbitrary integer coefficients, including negative coefficients, can also be handled but the number of slack variables increases to $W = b + \sum_{a_i < 0} |a_i|$ in the worst case. The most general case with arbitrary real-valued coefficients cannot be handled in a straight forward manner with binary slack variables.

In this way, canonical NP-complete and NP-hard problems with linear inequalities can be transformed to QUBOs and Ising models [101]. However, this transformation comes at an increased cost in the number of optimization variables, the exact number of which depends on the exact problem being solved. For integer and binary problems this increase is usually polynomial in the number of original problem variables, which in turn increases the number of fully-connected components that need to be embedded. Notice that quadratic and higher order polynomial constraints cannot not be included in this fashion efficiently, as the quadratic penalty functions leads to even higher order interactions in the cost function, which incur additional overhead in auxiliary binary variables. More general non-linear constraints cannot be included in the QUBO formulation using penalty functions.

In summary, the full QUBO cost function including constraints takes the form

$$C(\mathbf{x}) = \text{Obj}(\mathbf{x}, Q) + \sum_{\text{constraints } k} \lambda_k P_k(\mathbf{x})^2 \quad (13)$$

where $\text{Obj}(\mathbf{x}, Q)$ is the original cost function of the problem formulation of Eq. (5) and the $P_k(\mathbf{x})$ are linear functions in the binary variables representing the constraints, which are satisfied when $P_k(\mathbf{x}) = 0$. However, we still need to ensure that it is never energetically favorable to minimize the objective function with a violated constraint. This is the role of the λ_k in Eq. (13): the penalty, i.e., the cost of violating a single constraint should be dominating and therefore, should be greater than any other reduction in objective function value. In classical computing, this can be resolved by setting λ_k to an arbitrarily large value. However, with quantum annealing hardware this value has to be chosen with care, due to the limited dynamic range of the possible coupling strengths and pre-factors. The larger this value, the less resolution is effectively realizable in the low energy-level regime of \mathcal{H}_f , respectively $\text{Obj}(\mathbf{x}, Q)$. Large penalty factors result in extreme amounts of distortion in the ground state distribution due to noise. Therefore, it is important to consider setting λ_k such

that the lowest possible value that maintains correctness of the constraints is used. While there is no universal way to set these parameters, simple rules-of-thumb may be found either algebraically or empirically [119, 177]. In the case of purely constraint-based optimization problems such as constraint satisfaction problem (CSP) or boolean satisfiability (SAT) problems, no λ is needed when summing independent constraints, since the minimum of each constraint, as well as the global optimum is at a cost function value of zero, $C(\mathbf{x}_{\text{solution}}) = 0$. Furthermore, it becomes trivial to verify whether a candidate solution is a satisfying statement for CSP/SAT by checking the objective value given the solution (finding such a solution is still NP-hard). Previous work has shown interesting academic results using SAT problems to construct Bloom-like filters using quantum annealers [48]. Furthermore, it has been shown how to design optimal penalty terms for constraint satisfaction problems to minimize the error for quantum annealers [18].

Another way of handling constraints for the QA approach was proposed more recently [125, 180], which is very similar to Lagrangian relaxation in traditional combinatorial optimization (see, for example, Ref. [91]). The approach starts by formulating the constraints as quadratic penalty terms to the QUBO cost function as described above. By considering the partition function of the QUBO cost function and employing a Hubbard-Stratonovich transformation for the quadratic constraints P_k^2 followed by a Wick rotation in the auxiliary variables, an alternative cost function can be derived (see the original works [125, 180] for details). This alternative cost function is linear in the constraints at the expense of additional auxiliary real-valued variables $v_k \in \mathbb{R}$, one for each constraint, and takes the form:

$$C'(\mathbf{x}, \mathbf{v}) = \text{Obj}(\mathbf{x}, Q) - \sum_k \frac{v_k^2}{2\lambda_k} - \sum_k v_k P_k(\mathbf{x}) \quad (14)$$

Finding a solution that minimizes the original cost function $\text{Obj}(\mathbf{x})$ and fulfills the constraints $P_k(\mathbf{x}) = 0$ is equivalent to finding a saddle point of the alternative cost function $C'(\mathbf{x}, \mathbf{v})$ by calculating $\min_{\mathbf{x}} \max_{\mathbf{v}} C'(\mathbf{x}, \mathbf{v})$. Ohzeki [125] proposed an iterative scheme, where a gradient ascent is performed with respect to the variables \mathbf{v} , and in each step the minimal energy configuration for the binary variables \mathbf{x} is found using QA. So instead of solving one QUBO on the QA device, a whole series of QUBOs has to be solved, one for each proposed set of auxiliary variables, which incurs a substantial computational overhead. On the other hand, the advantage is that the constraints are not included as quadratic but linear terms in the QUBO, which reduces the overhead in binary variables related to the minor embedding of quadratic coupling terms. As a consequence larger problem instances involving more binary variables can be approached. An additional benefit of the proposed approach is that constraints that are quadratic in the binary variables can directly be incorporated into the QUBO formulation. It should be noted that this method of treating constraints has only been proposed recently and few application problems have been considered. Apart from some technicalities, which are not so clear from the original works, the effectiveness and robustness of the proposed method has to be evaluated on more problems in the future in order to arrive at a solid understanding of the benefits and shortcomings of the method.

2.4 Discrete variables

Discrete variables are of vital importance in many optimization problems, and have been investigated in the context of quantum annealing, for example in scheduling problems [143, 166] and graph colouring [106, 159, 160]. Each categorical variable can be encoded by using multiple qubits to represent a logical integer variable subject to a single constraint [67, 82, 157]. Binary encoding, analogous to classical binary encoding, is efficient in the number of qubits in the sense that one can encode d discrete states using only $\lceil \log_2(d) \rceil$ qubits. However, the binary encoding is not

used much in practice for application problems, since the interactions necessary to enforce the validity of the encoding as well as the realization of the couplings between logical variables are rather complicated to implement with the binary encoding. Additionally, this encoding has been investigated in previous works and has been shown to be detrimental to the QPU’s ability to find ground states [29, 30].

The one-hot encoding case is a standard technique where a logical variable with d possible states is represented by d qubits. Each of those qubits corresponds to one possible value of the discrete variable if set to the up-state $|+\rangle$, whereas all other qubits need to be in the down-state $|-\rangle$. The corresponding constraint is typically implemented as quadratic interaction terms between all d qubits. For limited-topology hardware this represents a drawback due to the embedding overhead, which causes noise to take a dominant role in a QPU’s ability to find ground states [165].

A recently developed alternative is the *domain-wall encoding*, which can encode discrete variables with one fewer qubit per variable, i.e., $d - 1$ qubits for d states. Additionally, it does not require interactions between all qubits that represent one variable, but instead only a linear coupling. It has been shown to be more efficient in different test problems [16, 29, 30]. In particular, a recent work showed that domain wall encoding leads to preferable dynamics during the annealing process, which in turn leads to an increased probability of finding feasible solutions, among other benefits [16].

2.5 Continuous variables

Representing continuous numbers using binary variables is not well-suited for quantum annealing due to precision issues with analog QPUs. Much like in classical computing, these numbers are represented by binary encoding schemes. A single floating-point variable \tilde{x} with N bits of precision would be encoded as:

$$\tilde{x} = c \sum_{i=0}^{N-1} b^i x_i, \quad (15)$$

where $x_i \in \{0, 1\}$ are the binary variables, $b > 1$ defines the logarithmic resolution and $c \in \mathbb{R}$ is a scale factor defining the range of the floating point variable. Such a continuous encoding is not well suited for QA approaches as the unavoidable noise of each binary variable on the QPU has profoundly different influence on the variable depending on its prefactor, which are exponentially spaced. Additionally, the number of qubits needed to encode such continuous variables needs to be rather large. For the very few practical applications that utilize such an encoding, the precision requirements are the most detrimental factor in QPU performance [131, 177]. This means that to solve high-precision problems necessarily means that (in the worst case) the precision level is below the noise threshold for QPUs.

2.6 Hardware architecture for quantum annealers

There exist numerous initiatives to build quantum annealing hardware at universities² and companies³. D-Wave Systems is currently the largest-scale and best-known of these initiatives, launching their first QA hardware product line in 2011 [80] and offering regularly updated hardware platforms ever since.

²See, for example, University of Southern California (FluQS <https://sites.usc.edu/qis/research>), University College London (www.ucl.ac.uk/quantum/study-here/cdt-delivering-quantum-technologies), and University of Innsbruck (<https://www.uibk.ac.at/th-physik/quantum-optimization/>)

³See, for example, D-Wave Systems Inc. (www.dwavesys.com), NEC (www.nec.com/en/global/quantum-computing/index.html), or Pasqal (<https://pasqal.io/>).

The core of an annealing QPU is an array of qubits, each of which realizes a simple two-level (e.g., spin- $\frac{1}{2}$) quantum system. D-Wave Systems utilize superconducting flux qubits to build their QPUs [68]. For a full technical description of the quantum processors and the physics behind each component, we refer the reader to [80]. In principle, annealers could be built from almost any of the qubit technologies mentioned in Section 1, although some exhibit certain properties that are amenable to quantum annealing (e.g., transmon qubits [97]).

One property that all quantum annealing QPUs have in common (especially in the absence of error-correction) is that the physical layout of the qubits is fixed, typically referred to as a *hardware graph* and denoted by U . This naming convention comes from the graph-theoretical description of the QPU, where each qubit (coupler) is represented by a node (edge) in an undirected graph. The exact topology of the hardware graph dictates the structure of graphs and possible \mathcal{H}_f that can be represented natively by the qubits' connectivity. The QPU topology is an engineering artifact arising from the design choices for the QPU. In general, due to engineering difficulties, connectivity between qubits comes at the cost of the total number of qubits in the QPU. Currently, no QPU has the ability to change its hardware topology after fabrication.

Due to various technological limitations in manufacturing, calibration, or other anomalies, a small portion of the qubits and couplers in the QPUs may be defective and hence not programmable. The percentage of qubits and couplers that remain functional once fully exposed to users is referred to as the *hardware yield*. For D-Wave QPUs, the qubit yield is typically above 97% for current processors [1]. The graph representing the functional qubits and couplers is referred to as the QPU's *working graph*.

The specific topologies of the D-Wave 2000Q and earlier generations of QPUs are referred to as *Chimera* topology. These graphs are composed of a 2D lattice of small complete bipartite graphs (each called a tile, or unit cell). Typically, these graphs are denoted as C_X (where X is the length of one side of the square lattice, for a total of X^2 tiles), and each tile is in itself a $K_{N,N}$ bipartite graph (full connectivity between left and right partitions). The D-Wave 2000Q has a size of C_{16} and $K_{4,4}$ tiles. Each qubit in the Chimera topology has six couplers, where four couplers are inside the unit cell and two are to different unit cells. A diagram showcasing this topology is in Fig. 2a.

The latest generation of D-Wave QPUs have a novel topology, called *Pegasus*, which differs in two major aspect from the Chimera graph: first, the graph degree is 15 (meaning each qubit is connected to 15 neighbors), and secondly, the hardware graph is not bipartite. Instances of the Pegasus topology that contain $N \times N$ unit cells are referred to as P_N and consist of $24N(N - 1)$ qubits. This results in a significantly more complex structure, allowing for denser graph structures of \mathcal{H}_f to be solved by the QPUs, and hence more difficult optimization problems (a technical description comparing Chimera and Pegasus can be found in Ref. [20]). For a diagram of a Pegasus graph, see Fig. 2b.

2.7 Minor-embedding techniques for hardware graphs

It is evident that not all interesting optimization problems defined by the interaction graph G specified in \mathcal{H}_f can be natively expressed in the hardware topologies U , i.e., the Chimera or Pegasus graphs. However, it is possible to reformulate any arbitrarily-structured \mathcal{H}_f into a new hardware-compatible \mathcal{H}'_f via the technique of graph *minor-embedding*, a well-studied problem in graph theory (see, e.g. Ref. [46]). This process produces a mapping between one graph to another such that the relevant topological properties of the original graph are preserved. For QA, this involves encoding variables as *chains* of multiple physical qubits on a QPU such that they act as a single logical qubit [32, 33]. In Ref. [20], the authors explore the various chain requirements when embedding certain graph families in both Chimera and Pegasus graphs.

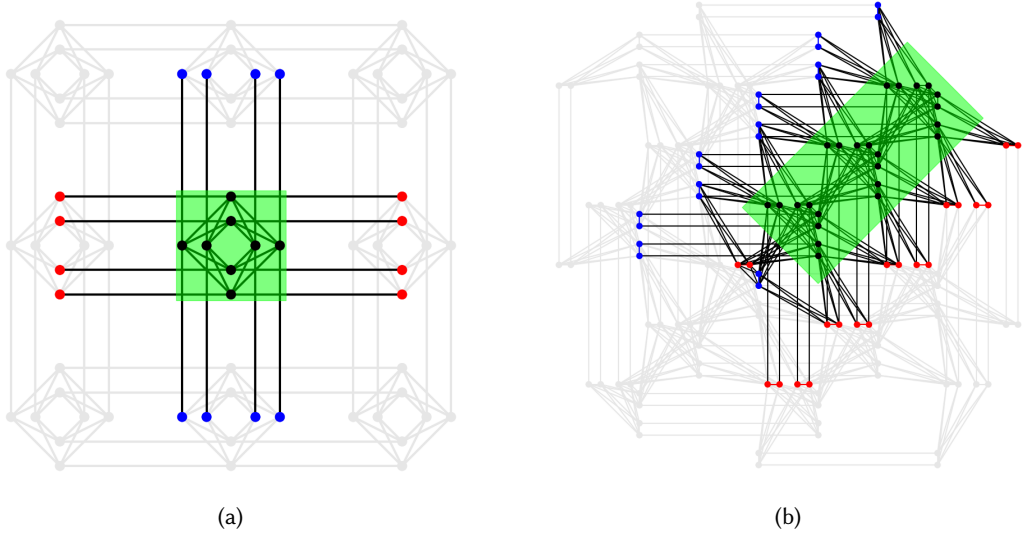


Fig. 2. Graphical representations of the topology graphs of two D-Wave hardware architectures. Each qubit is represented by a node in the graph, and every coupler is an edge between nodes. (a) Chimera graph: the highlighted green square is a $K_{4,4}$ unit cell. Blue and red nodes are vertically and horizontally connected qubits to the highlighted unit cell. (b) Pegasus graph: the highlighted region represents the equivalent unit cell, composed of K_4 and $K_{6,6}$ subgraphs. Blue and red nodes here represent vertical and horizontal pairs of qubits connected to the highlighted cell. Each qubit has at most degree 15 (explained in detail in https://docs.dwavesys.com/docs/latest/c_gs_4.html). Both figures were made using the D-Wave Ocean Python package [2].

There exist several difficulties when using minor-embedding techniques. First, deciding whether a graph can be minor-embedded into another is a known NP-complete problem, and so polynomial-time heuristics are used in practice [21, 26]. Secondly, the use of embeddings requires an additional set of constraints to be imposed on the qubits representing a logical qubit (the magnitude of which is called the *chain strength*). These enforce that the minimum energy of \mathcal{H}'_f is obtained only when all physical qubits representing a logical qubit are in the same state. It has been shown that the magnitude of the minimum chain strength increases with the degree of the graph of \mathcal{H}_f [33, 165, 176], which can create distortions in the resulting Hamiltonian \mathcal{H}'_f due to noise. Generally, determining the best suited chain strength coupling is nontrivial task and still subject of current research, see, e.g. [52].

An illustrative example of the minor embedding procedure and the necessary grouping of physical to logical qubits is depicted in Figure 3. It can be observed, that the logical nodes 4 and 8 have degree 5, which is larger degrees than the maximal degree 4 of the hardware graph, which is assumed to be a simple square lattice for illustrative purposes. In the example, two physical nodes are used to represent the logical node 4, i.e., form a chain. Also, the triangular coupling of the nodes 4, 5 and 7 is incommensurate with the square lattice, and therefore the additional chain representing node 5 has to be introduced. Three physical nodes are required to represent the logical node 8, since it has more than four connections as well as being part of a triangular coupling.

Various methods have been developed to find efficient minor-embeddings in various scenarios. Cai et al. [26] proposed a heuristic algorithm for finding a minor embedding G_{emb} of graph G to

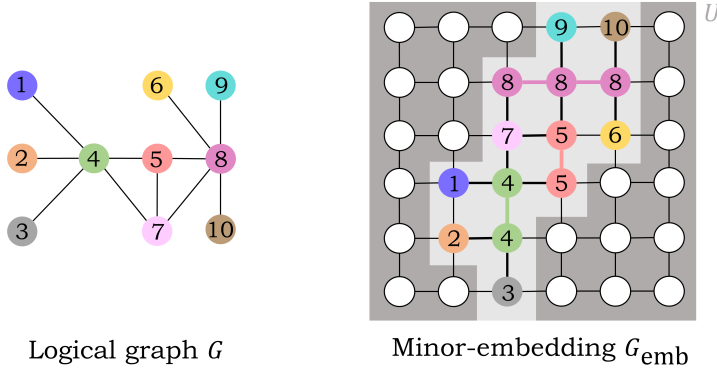


Fig. 3. Logical graph G and its minor-embedding G_{emb} in a general square lattice U . Each node of G is mapped to one or more nodes in U . Each group of nodes representing one logical node, i.e., each chain, is highlighted with a different label and color. Adapted from [32].

the hardware topology graph U that is practical for sparse U and G with hundreds of vertices. Previously, Robertson and Seymour [144] had proposed exact algorithms with the same aim, but these are impractical for graphs with more than tens of vertices. The success of a heuristic to find minor-embeddings heavily depends on the fact that, if G is significantly smaller than U , then there are probably a large number of distinct minor-embeddings G_{emb} in U , and their number grows exponentially with the number of vertices of U . Since the working graph U may be different in each QPU, Cai et al. [26] presented a heuristic algorithm where both G and U are given as an input. The heuristic algorithm aims to either minimize the maximal chain length (the maximal number of physical qubits representing any logical qubit), or the total embedding overhead (the total number of used physical qubits in U).

Klymko et al. [89] were the first to provide a polynomial-time algorithm for generating large clique minors in subgraphs of a hardware graph. Via tree decomposition techniques, they develop an algorithm taking $O(N^4)$ time for a $C_{N,4}$ Chimera topology. This algorithm was one of the first embedding techniques constructed specifically to be resilient in the face of missing qubits in the hardware graph U . Later, Boothby et al. [21] defined a combinatorial class of *native clique minors* in Chimera graphs with vertex images of uniform and near minimal size, and provide a polynomial-time algorithm that finds a maximum native clique minor in a given induced subgraph of a Chimera graph in only $O(N^3)$ time.

Zaribafiyani et al. [182] proposed a systematic, deterministic, and scalable embedding algorithm for embedding the Cartesian product of two complete graphs into D-Wave Systems' Chimera hardware graph. They defined an application-specific algorithm that exploits both the intrinsic structure of combinatorial optimization problems and the structure of the quantum annealer, defined by the Chimera topology. Their Cartesian product of complete graphs (CPCG) embedding algorithm showed ability in obtaining high-quality embeddings on ideal Chimera chips with equal-length chains by using far fewer physical qubits, advantageous running time scaling and cost-effectiveness in scaling up for larger chip architectures over more-general heuristic methods. However, they did not explore optimal modifications to the method to handle inoperable qubits and couplers.

Although well established algorithms exist, the minor-embedding process is by itself a non-trivial problem, since it is the dominant costly step time-wise and the largest detractor on performance for QA. Depending on the problem and the hardware connectivity, some minor embedding algorithms might be inefficient or even fail to find a valid embedding. Typically, the number of physical qubits

needed to formulate a specific problem with only pairwise interactions on the Chimera topology of the actual hardware scales quadratically with the number of logical qubits of the problem, $N_{\text{physical}} \sim N_{\text{logical}}^2$. For problems where true three-body and higher-order interaction terms have to be included in the Hamiltonian, as for instance it is naturally the case in the quantum chemistry domain, the scaling is much worse, typically exponential in the number of variables [154, 162].

Recently, the increase in the time-to-solution due to the embedding overhead was compared between several embedding schemes [90]. The authors investigated the embedding of problems with fully connected graphs, i.e., with all-to-all couplings, onto the 2D planar square lattice graph, the Chimera graph of the actual D-Wave hardware and the recently proposed Lechner-Hauke-Zoller (LHZ) scheme [96]. All embedding schemes induced unfavorable exponential overhead in the time-to-solution with increasing problem size, underlining the necessity to investigate novel approaches to QA. When approaching practical application problems, D-Wave provides various tools and guidance for minor embedding in their documentation, see [3].

2.8 Noise and mitigation strategies

The disordered, interacting, time-dependent and open nature of the many-body Ising Hamiltonian realizable in actual QA hardware systems makes it difficult to draw universal conclusions about the power of programmable QA [181]. However, QA is an algorithm that seems likely to be more robust against environmental noise than quantum algorithms implemented on gate-model quantum computers. With QA, the complexity of the algorithm is simpler: due to the limited scope of the implemented Ising Hamiltonian, the set of operators are fixed for each qubit and coupler. For example, in transverse-field Ising Hamiltonians, this is limited to the single-qubit σ^x and σ^z operators, and the two-qubit $\sigma_i^z \sigma_j^z$ operator. Therefore, full programmability of the gate set is not needed, and only the relative magnitudes of \mathcal{H}_i and \mathcal{H}_f are varied during the QA algorithm. Furthermore, QA is resilient against dephasing errors, as the relative phases between the computational eigenstates do not encode any information. Even in the case of decoherence, the annealing process is still completed and a candidate solution to \mathcal{H}_f is obtained. It has been shown in practice that annealers still perform heuristic optimization in the absence of quantum coherence and entanglement [7, 43].

Dickson et al. [45] performed the first experimental demonstration of the effect of thermal noise on D-Wave QA processors [68, 80]. In this context, robustness against environmental noise was defined as the ability of an open QA system to yield the correct solution with acceptable probability within a time comparable to the closed-system adiabatic timescale. The authors examined this by constructing a 16-qubit toy problem designed to have a small energy gap between the lowest two energy levels. It was demonstrated that in the limit of weak coupling to the environment (i.e., relatively low levels of thermal noise), annealing non-adiabatically across the minimum gap did not hinder QA, but rather enhanced its performance. This was demonstrated by manually raising the operating temperature of the chamber in which the QPU was operating, thus allowing for a controllable amount of noise to perturb the system. The specific point demonstrated was that thermal noise does not necessarily inhibit obtaining ground states at the end of the annealing, even with small minimum gaps and non-adiabatic conditions. However, the caveat remains that the annealing times need to be sufficiently long, and the system must be only weakly coupled to the environment⁴.

Despite the relative robustness of QA in terms of ground state probabilities, noise can also directly perturb problem formulations in the analog system, i.e., \mathcal{H}_f . The Hamiltonian parameters

⁴Due to thermal relaxation, in the worst case the time to reach equilibrium in an open quantum system can grow orders of magnitude with respect to the time needed in a closed system.

are subject to low-frequency noise that can be modeled as static Gaussian disorder, introducing artificial longitudinal fields. In this context, Venturelli et al. [165] compared QA and simulated thermal annealing to measure the effects of disorder in ground state distributions on the Sherrington-Kirkpatrick Model (SKM) with couplings randomized from a bimodal distribution of values ± 1 [149]. The two key findings from this work in regards to noise are as follows. Firstly, it was evident that the ferromagnetic couplings needed to encode chains representing logical qubits on the annealer needed to increase as \sqrt{N} (up to logarithmic corrections) in order to properly represent the SKM for larger sizes. Secondly, that static noise (i.e., thermal noise whose timescale is much slower than the annealing dynamics) causes a reshuffling of the energy levels in the problem Hamiltonian. It was ultimately observed that the performance of simulated thermal annealing was similarly degraded in the face of such ground state disorder. However, when considering same-magnitude noise/disorder, it was observed that quantum annealing had a limited scaling advantage for this dataset. This further gives evidence that well-separated eigenstates are particularly important for the success of the quantum annealing algorithm.

There are multiple ways to mitigate some of the effects of noise in quantum annealing. The most common technique is *spin-reversal transforms*, which are used to effectively average over the asymmetric final Hamiltonian distortion due to noise. This is done by creating random spin vectors $S = \{-1, 1\}^N$ and recalculating a new \mathcal{H}'_f subject to this shift, where $h'_i = h_i \cdot S(i)$, and $J'_{ij} = J_{ij} \cdot S(i)S(j)$. It is simple to show that these transformations do not change the distribution of ground states in the system, only the signs of a subset of terms. This new Hamiltonian \mathcal{H}'_f is then submitted to the annealer. To recover the samples subject to the original Hamiltonian \mathcal{H}_f , we simply multiply the solutions by the vector $S(i)$. Typically, many such transforms are used when solving a single \mathcal{H}_f . While this method is effective in reducing the error due to static noise, each new \mathcal{H}'_f results in another programming cycle, increasing the total wall-clock time of using QA. Furthermore, it has been shown that increasing the number of transforms has diminishing returns for a fixed number of samples [139].

With its third generation quantum processors, D-Wave Systems introduced the possibility to control one of the annealing terms for each qubit individually. Specifically, it is possible to delay or advance the annealing start time of each qubit independently, within some predefined ranges. This feature is called *anneal offsets*. The motivation behind this feature is that each qubit may experience a slow-down in dynamics at a different point in the annealing process, defined by the magnitude of \mathcal{H}_f . A naive example is embedded problems, which require chains of qubits with ferromagnetic couplings. Longer chains have a lower effective tunneling energy and may freeze in a fixed state earlier in the annealing process. Anneal offsets can therefore be used to shift the freeze-out time of each chain (or qubit) to improve the probability of finding the ground-state. Using this technique, Andriyash et al. [51] highlight the potential of this feature and addressed the problem of integer factoring, where given an input number p one tries to find the integers a and b such that $p = a \cdot b$. They addressed the problem by executing a multiplication circuit backwards, i.e., the circuit has inputs a and b and achieves an optimal state that minimizes the optimization problem when $a \cdot b = p$. Employing the anneal offsets, they showed that the computation can be about 1000 times faster than without the offsets. The same technique was also applied to 2-SAT problems with 12 Boolean variables [73]. Annealing offsets have been shown to mitigate first-order phase transitions in simple problems that could be solved exactly, exponentially increasing performance compared to SA [155]. Yarkoni et al. [179] investigated the use of anneal offsets for random instances of the maximum independent set problem. They used the covariance matrix adaptation evolutionary strategy (CMA-ES) [66] to heuristically optimize the annealing offset of the qubit chains, i.e., the Δt of advance or delay for each qubit in physical time. Therefore, the functions $A(t)$ and $B(t)$ in

Eq. (1) are in principle different for each qubit, i.e., $A(t_i)$ and $B(t_i)$. They showed that by tuning the annealing offsets, quantum annealing can improve by an order of magnitude in ground state probability, and is able to also find better solutions than without annealing offsets. In general, they observed that the improvement increases with problem size until a certain point, after which the gains mostly disappear due to the increase in variables to be optimized and the consequently harder optimization problem to be solves. Similarly, Adame and McMahon [4] employ inhomogeneous annealing offsets that depend on the local connectivity of each qubit and show for certain problems that the probability of finding a ground state can be increased over the homogeneous driving case by factors on the order of 10^3 to 10^7 .

Alternative schemes to suppress errors in QA have been proposed that utilize ancillary qubits in the problem Hamiltonian [138]. The role of these extra qubits is to enforce penalties that act as analogues to classical bit-flip error mitigation strategies. Logical variables are encoded redundantly with multiple physical qubits, and classical states are recovered by a simple majority vote. While this procedure incurs a polynomial overhead in the number of qubits required to describe a problem, the authors show numerically that adding such penalty terms can increase the minimum gap, giving hope that QA may experience a speedup using these methods.

Overall, it is clear that current state-of-the-art QA does not have as clear or established error-correction methodologies as gate-model QC does. Furthermore, it is clear that error-correction in QA requires fundamentally novel paradigms to be implemented. Nonetheless, ongoing research in both hardware development (additional control features, or alternative drivers) as well as software (spin reversals, tuning of annealing parameters) implies that QA may still significantly benefit from error-correction (and suppression).

2.9 Software and hybrid algorithms

Despite the impressive advancement of quantum hardware in recent years, development of high-level software to address quantum annealers has been slow. Many different aspects of using QA still require classical software, such as hardware control, formulation of Ising models/QUBOs, APIs, etc. Specifically for QA hardware access, the majority of the relevant software tools are owned, operated and developed only by D-Wave Systems.

D-Wave Systems maintains a set of publicly available repositories for application development in Python, named Ocean [2]. We briefly describe the most relevant ones that are relevant for practitioners. Constructing simple Ising/QUBO problems is done using the classes available in `dimod`. To submit these problems to D-Wave QPUs and solvers, D-Wave provides abstracted composite layers for both embedded and topology-specific problems in `dwave-system`. These composites also expose to the user the necessary parameters to implement hardware-specific control, such as spin-reversals, annealing offsets, pausing, reverse annealing, and others. There is also a separate library specifically dedicated to minor-embedding, `minorminer`.

An extension of the graph construction and visualization tool NetworkX [65], is available through Ocean (`dwave-networkx`). This package allows users to interact directly between D-Wave software tools and the functionality of NetworkX, which is particularly useful for graph theory problems. Ocean also provides a set of classical solvers (implemented in Python) that can solve QUBO/Ising problems, such as simulated thermal annealing (`dwave-neal`) and Tabu search (`dwave-tabu`).

In principle, any library that develops Ising/QUBO models for practical purposes can be used for quantum annealing. For example, Google's OpenFermion project [107] has tools for analysis of simple quantum chemistry models using quantum computers. Although originally designed for gate-model QPUs, some algorithms (such as Hartree-Fock) can also be used for quantum

annealing [154]. For optimization, there is a public repository from the authors in [29] to implement domain-wall encoding for discrete variables⁵.

Other commercial entities are working on developing specialized tools for quantum annealing. Of those, a promising technology is being developed by ParityQC to implement novel methods to improve quantum annealing performance [70]. The specific compiler being developed utilizes counter-diabatic protocols to increase the ground state probability in LHZ annealing architecture. The ParityQC compiler also has utility for gate-model QCs, and has partnered with multiple hardware providers (both annealing and gate-model based) to continue their development.

To combat many of the issues raised in previous sections regarding solving problems directly with quantum annealers, a hybrid approach is often used. Specifically in the context of current-generation quantum annealers, hybrid algorithms are used to solve problems that are too large to embed directly on QPUs. In this approach, the original problem is partitioned into multiple sub-problems, each of which is solved using the quantum annealer, and then they are recombined to get comprehensive solution to the original problem. The first such hybrid algorithm that utilized quantum annealing processors was `qbsolv` [19], which used an outer classical loop of Tabu search [58] and a D-Wave QPU in its inner loop to flip the clusters of variables found by Tabu. Later, this algorithm was open sourced, and eventually resulted in the Python Ocean library `dwave-hybrid`, which allows users to construct their own hybrid algorithms. The package provides multiple graph partitioning tools and classical techniques to break apart problems and merge solutions. This allows users to construct hybrid quantum-classical algorithms tailored to the specific problem being solved. It is important to note that the vast majority of industry applications with quantum annealing have used hybrid algorithms (rather than the QPU directly) because of the relatively small number of available qubits. In general, partitioning a large problem into a set of smaller problems involves difficulties and can result in sub-optimal solutions. It has been shown in practice that embedding larger sub-problems in the inner loop of the optimization that uses the QPU results in improved performance of the hybrid algorithm [127].

In general, any algorithm that offloads some part of its calculation to a quantum processor can be considered a hybrid quantum-classical algorithm. Genetic and evolutionary algorithms have also been explored in this context, as some of the inner algorithmic steps lend themselves naturally to the QPU. Chancellor [28] tries to answer the question of whether QA can be used to gain advantages over modern classical algorithms, by embedding both quantum and classical search techniques, or by using multiple quantum searches sequentially. He constructs analogues to state-of-the-art classical algorithms, but made of quantum building blocks, leading to subroutines that are less sensitive to noise in the form of problem mis-specification than QA. Local searches deal with small subspaces of the total solution space, and hence only errors that occur on states within these subspaces are relevant. A local search can therefore give meaningful results even in a problem where the global optimum is no longer correctly specified due to noise. Chancellor [28] suggests to use local quantum search as subroutines in well known classical algorithms like the ones from Monte Carlo's family (e.g., parallel tempering, population annealing, and isoenergetic cluster updates).

Another approach that has been proposed uses the reverse annealing feature in QA as the mutation operator in a genetic algorithm (quantum-assisted genetic algorithm, QAGA) [86]. Since reverse QA has been demonstrated for local refinement of candidate solutions, it may be considered a tunable mutation of any classically prepared state prior to the reverse annealing procedure. Here, the point $s = s'$ in the anneal (s being normalized time, $0 \leq s \leq 1$) determines the degree of mutation of a single candidate solution. Indeed, this is not possible with forward QA, since the optimization

⁵Code available here: doi.org/10.15128/r27d278t029.

does not provide any information from the initial superposition to the final answer. QAGA was tested on three input classes (bimodal spin glasses, anti-cluster inputs, and deceptive cluster loops) against a suite of classical algorithms: simulated annealing (SA) [87], parallel tempering (PT) [156] and parallel tempering with isoenergetic cluster moves (PT-ICM) [49]. When tested in terms of time-to-solution (TTS), QAGA performed better than QA, having a smallest variance in the output. Moreover, in comparison to classical solvers, QAGA dominated on all the input classes, even if failed to reach ground states on a limited number of inputs.

2.10 Role of time in quantum annealing

There is some ambiguity in the quantum annealing community in regards to the description of time when using quantum processors (and in some cases classical algorithms). To clarify this, we define a consistent overview of the role of physical time (wall-clock time), and relate this to the QA algorithm from both a theoretical and experimental point of view. From the practical application standpoint, a QA processor represents an analog processor that samples from a semi-classical Boltzmann distribution described by a Hamiltonian \mathcal{H}_f in a predefined fixed amount of time. The computation proceeds in three main steps [109]:

- (1) The initialization phase consists of programming the weights h_i and setting the coupling strengths J_{ij} of Eq. (3). The qubits are then initialized according to \mathcal{H}_i . The time needed for such a programming phase is denoted T_{init} .
- (2) The annealing phase, where the transition from the initial to the final Hamiltonian according to the annealing schedule $A(t)$ and $B(t)$ from Eq. (1) is performed. The physical annealing time is denoted as T_a , and is a user-provided parameter to the QA algorithm. In situations where reverse annealing is used, the total annealing time has two contributions, the time duration T_{rev} of the reverse annealing plus the above mentioned forward annealing time T_a , i.e., $T_a^{\text{tot}} = T_a + T_{\text{rev}}$.
- (3) The readout phase, where the qubits are measured at the end of the annealing process. The classical configuration of the spins s_i is obtained, which provides a candidate solution to the optimization problem defined by \mathcal{H}_f . The readout time is denoted as T_r .

Step (1) is performed only once per use of the QPU for a given problem \mathcal{H}_f . Steps (2) and (3) are repeated k times to obtain proper statistics for the results and increase the probability to find the global optimum, as is typical for heuristic optimization algorithms. Therefore, the total wall-clock time to obtain k samples for a given problem instance is $T_{\text{total}} = T_{\text{init}} + k(T_a^{\text{tot}} + T_r)$. Typically, the time needed to obtain the ground state⁶ scales with the size of the input problem, \mathcal{H}_f . It is important to pause and appreciate that the scaling of total time (T_{total}) relative to problem size is not the same as the “true” algorithmic scaling of the underlying annealing algorithm, T_a , and one should not be conflated with the other. The former, T_{total} , is purely an engineering artifact, with constant terms T_{init} and T_r that are dictated by the specific implementation of hardware, and a number of repetitions k provided by the user. The latter, T_a , is the amount of time used for a *single* annealing cycle. Theoretically, it is known that slower (longer) annealing times increase the probability of remaining in the ground state (although in practice, it is often more efficient to reduce T_a and execute more annealing cycles to compensate). Thus, the smallest such T_a that maximizes the ground state probability can be considered the true run-time of QA for a particular problem. Practically, the optimal trade-off value for T_a should be long enough to give sufficiently high probability to find the ground state and at the same time should be as short as possible. The value of T_a is intimately related to the problem structure and as such not known *a priori*. However,

⁶Different metrics can be used to define this time. A standard choice is time-to-solution (TTS) that is needed to observe an average probability $X\%$ of obtaining the ground state, denoted by $\text{TTS}|_{X\%}$.

T_a is an input parameter to the QA heuristic, which results in a burden for the user to determine a value for T_a that is appropriate for the specific problem. In this sense, the QA heuristic is not a fixed-time algorithm in the classical sense, but merely a fixed-time algorithm for one a particular problem once it has been calibrated properly.

3 QUANTUM ANNEALING IN REAL-WORLD APPLICATIONS

In this section we introduce the various domains in which there has been considerable progress related to industrial applications involving quantum annealing. We review the main works that are representative of the methods developed in those application areas, and highlight the individual contributions of each work as well as their collective importance. Due to the rapid and ongoing development in the field of quantum annealing, we note there may be papers that are relevant but not included in our summary due to scope. Ultimately, we chose the works to include based on three main criteria: original research upon which the application area is based, work that has significantly advanced each domain area, and results or methods that have considerable relevance to a practitioner. Table 1 provides a more extensive list of papers focusing on the application fields of QA discussed in this section.

Table 1. Publications on real-world applications of QA. References discussed in the main text are in bold.

Application field	References
Mobility	Traffic flow optimization: [119] , [178] , [36] , [126] , [153] , [75] .
Scheduling and logistics	Scheduling Problems: [143] , [166] , [74] , [176] , [180] , [130] , [125] , [152] .
	Vehicle Routing Problem: [55] , [23] , [177] .
Quantum Simulation	Chemistry: [175] , [154] , [116] , [110] , [134] , [72] .
	Physics: [69] , [85] .
	Biology: [98] , [172] .
Machine Learning	Classification: [114] , [122] , [123] .
	Reinforcement learning: [121] , [38] .
	Cluster analysis: [120] , [93] .
	Matrix factorization: [128] , [132] , [59] .
Finance	Portfolio optimization: [146] , [50] , [164] , [115] , [61] , [147] , [135] .
Miscellaneous	Finite-element design: [163] .
	Material design: [88] , [173] .

3.1 Traffic flow optimization

The problem of traffic congestion on a road network occurs if the demand volume exceeds the road capacity, leading to important problems such as delays, increased fuel consumption, and additional pollution. Optimizing traffic flow hence represents one of the real-world challenges that can be modeled by using different strategies, such as classical optimization (see Refs. **[10, 25, 137]** for recent examples), reinforcement learning **[53, 142, 169]** and neural networks **[94, 171]**. More recently, solutions that take advantage of QA technologies have been also proposed.

Neukart et al. **[119]** were the first to show how to transform real-world data – GPS coordinates in this particular case – and formulate a relatively simple QUBO to be solved as a real-time application. Their target was to minimize the congestion in road segments on overlapping routes in order to minimize the time for a set of cars to travel between individual sources and destinations. The dataset tested was composed of 418 cars with 3 alternative routes generated for each car to redistribute

traffic, which resulted in 1254 logical variables to represent the problem. Using the hybrid-quantum solver `qbsolv`, they partitioned the problem into sub-problems targeted for execution on a D-Wave 2X system, obtaining a significant improvement over randomly uniform assignment of cars to routes. Hence, although a simplified version of traffic flow (limited set of cars, no communication to infrastructure, no time window) was presented, the paper showed that some types of real-time optimization problems can be posed for the D-Wave systems and the hybrid tools that use them.

Theory was put into practice with a follow-up by Yarkoni et al. [178]. In this paper, the authors implemented the paradigm proposed in [119] to navigate a small bus fleet in real time for the duration of the Web Summit 2019 conference in Lisbon, Portugal. Volkswagen AG partnered with the local city authorities of Lisbon to present the project, named the “Quantum Shuttle”, which operated a public shuttle bus service between the Web Summit conference venue and the city center of Lisbon. To adapt to a real-world scenario, GPS points of live traffic data are used in the objective function definition instead of road segments. The largest QUBO that was solved was quite limited in size (12 variables, with 5 buses being navigated concurrently), and three different methods of using QA were tested: direct embedding, hybrid algorithms, and D-Wave’s Hybrid Solver Service. Each of these methods was integrated into a custom-build quantum optimization service build using cloud computing which was used to navigate the bus fleet. Thus, this was the first commercial application that depended on a quantum processor to perform a critical live task.

Another extension of [119], by Clark et al. [36], showed how to route multiple robots on a grid in real time. While their approach is similar to the original, the authors presented an additional constraint to avoid collisions (or overlaps) in time. Their objective was to find a route of minimum cost for each robot, where the cost is dependent on the distance the robot travels and the change in direction in the chosen route. They also used the decomposing solver `qbsolv` to address a maximum problem size of 800 variables. According to this study, the performance of QA is promising, although performance bottlenecks were evident on both the classical part (creating the QUBO problem and collision detection) and the quantum part (communication and embedding time) of the hybrid algorithm.

Further work based on [119] was proposed by Ohzeki et al. [126]. The novelty is that they considered the time dependence on the binary objective characterizing a certain route. The application in this work was the controlling problem of automated guided vehicles (AGVs), which are portable robots for moving materials in manufacturing facilities and warehouses. The authors considered problem instances with 10 AGV and 6 candidate routes leading to a maximum size of 60 variables, which can be directly solved on the D-Wave 2000Q hardware. Larger problem instance up to 900 variables are solved using `qbsolv`. Comparison of the QA performance to alternative approaches using the Fujitsu digital annealer[9] and Gurobi software solver[64] did not show a clear benefit of utilizing the QA hardware.

Stollenwerk et al. [153] solved the conflict resolution problem for wind-optimal airplane flight trajectories. The trajectories are obtained by minimizing the fuel cost over the routes with given origins and destinations and desired departure times under consideration of wind forecasts. Here too, the presented version of the problem is a simplification of the real-world use-case since they do not consider all the physically realistic trajectories, but rather perturb trajectories by delaying the corresponding flight on the ground prior to departure. The heuristic embedding algorithm described in [26] was used to embed instances with up to 50 flights and 104 conflicts on the D-Wave 2X, as well as up to 64 flights and 261 conflicts on the D-Wave 2000Q. The paper also introduces a novel pre-processing algorithm to eliminate potential conflicts that, given a maximum delay, can never occur and clustering adjacent conflicts. The authors concluded that, for reasonable modeling choices, the most challenging sub-problems that are programmable in the current devices are solved to optimality with 99% probability within a second of annealing time. However finer model

discretizations, as well as larger problem sizes, decrease the success probability due to the limited precision QA processors.

Very recently, Toyota Central R&D Labs also presented a study aimed to globally control traffic using signals arranged on a square lattice with a D-Wave 2000Q quantum annealer [75]. They follow a 2-step approach. First, they formulate a signal optimization problem that minimizes the imbalance of traffic flows in two orthogonal directions. Afterward, they reformulate the problem as an Ising Hamiltonian. The Hamiltonian is divided into 42 groups (by the graph partitioning technique in the Ocean library [2] provided by D-Wave) and the optimization problem is solved in parallel. For comparison purposes, the problem was also solved using SA, but QA performed better in the specific parameter domain. Moreover, the difference between the time average of the cost function with problem partitioning to a problem without partitioning seemed to increase with the rate at which vehicles go straight at intersections, which indicates a performance drawback of the partitioned problems.

In conclusion, all presented methodologies to address traffic flow optimization problems are based on assumptions that make them far from faithfully representing real-world problem challenges. However, QA processors have been proven to be valuable tools to address simplified formulations and show high potential, in particular if we take into account the rapid development of successive generations of QA systems over the recent times.

3.2 Scheduling and logistics

Many planning applications include scheduling, which deals with allocating resources and time to tasks while taking constraints into account [31]. While many studies were still looking at the performance of quantum annealers on problems native to their architecture, Rieffel et al. [143] explored general mappings to QUBOs for hard combinatorial optimization problems that arise in operational planning, and applied them to two parameterized families of planning problems: navigation-type planning problems and scheduling-type planning problems. In particular, they explored two different approaches for mapping problems to QUBOs, which are the time-slice and conjunctive normal form (CNF) methods. They provided comparisons of the effectiveness of different embedding mappings onto hardware and parameter choices for their solution on the 509-qubit D-Wave Two machine. In all cases, they used an annealing time of 20 μ s to take full advantage of the machine capabilities (the shortest times available). They considered a range of problem sizes from 4 to 12 for the navigation-type and 8 to 16 for the scheduling-type problems. Although their study did not produce results competitive with state-of-the-art classical approaches, they investigated the embeddability in future architectures and stated that increasing the local connectivity within QPU unit cells would have a much greater impact on the ability to run instances of applications than simply increasing the number of unit cells.

Later, Venturelli et al. [166] presented in detail an implementation of the job-shop scheduling problem (JSP) that makes use of the same D-Wave quantum annealer. They formulated a QUBO version of the JSP decision problem as a constraint satisfaction problem (CSP): optimizing the allocation of resources in the execution of sequences of operations was done by encoding constraints on location and time. The JPS involves a set of jobs (sequence of operations) that must be scheduled on a set of machines, with the final target of minimizing the makespan, i.e., the completion time after the run of the last job is over [79]. The analysis was performed on up to $N = M = 9$ and $p_{\max} = 20$ using 200 instances, where N is the number of jobs, M is the number of available machines, and $p_{\max} = 20$ is the maximum execution time of the operations. They compared the performance of the D-Wave QPU to two classical algorithms: the branch and bound algorithm [24] and the MS algorithm by Martin and Shmoys [104]. While the QPU performed better than the MS method, the QPU took longer than the branch and bound algorithm to generate valid solutions.

More recently, Ikeda et al. [74] addressed the Nurse Scheduling Problem (NSP) with hard constraints using the D-Wave 2000Q quantum annealing device. They formulated the problem as a constrained QUBO and implemented it through the randomized embedding algorithm based on Cai et al. [27]. With N as the number of nurses whose shifts are to be scheduled on D working days, then the optimization problem consists of $N \times D$ variables, where the single variable specify the assignment of nurse n to day d , for $n = 1, \dots, N$ and $d = 1, \dots, D$. The authors compared QA to simulated annealing for $N = \{3, 4\}$ and $D = \{5, 6, \dots, 14\}$ and drew statistics on 1000 samples for 10 shots for D-Wave 2000Q and 500 annealing times 10 shots for SA. The authors tested both forward and reverse annealing and showed that for a fixed sample size and fixed annealing duration, the success probability of completely satisfying all constraints decreases as the size of the schedule D and the size of the roster N increase. Although reverse annealing improved the success probability, this did not happen uniformly, and simulated annealing gave better results, highlighting the need for further investigation into the parameters of the annealing process.

Also related to JSP is the vehicle routing problem (VRP) [161], which is a combinatorial optimization and integer programming problem to determine the optimal set of routes to be taken by vehicles to serve a given set of customers as efficiently as possible. Belonging to the category of NP-hard combinatorial problems, Feld et al. [55] were the first to address the capacitated vehicle routing problem (CVRP), where each vehicle has a capacity constraint on how much it can transport, by mapping it onto a QUBO formulation and solving it using a D-Wave's 2000Q quantum annealing hardware. The problem sizes considered in this work ranged from 14 to 38. In order to fit the QUBO directly onto the QPU, they used the minor embedding technique by Cai et al. [27] in combination with D-Wave's qbsolv tool, leading to a sub-problem size of 20 logical variables. In particular, they presented a hybrid method based on a two-phase approach (a clustering and a routing phase) which could compete with other classical and 2-phase heuristics, and in some cases even provided better solution quality. However, this hybrid method did not provide overall advantages in terms of solution quality or computation time compared to classical heuristics, which, according to the authors, was due to both the size of the hardware used and the need to use additional tools such as qbsolv.

VRP and CVRP were also addressed by Borowski et al. [23], who presented four quantum-classical hybrid algorithms. The full QUBO solver (FQS) and the average partition solver (AVS) were designed to solve the plain VRP. The DBSCAN solver (DBSS) also addressed CVRPs provided the capacities of all vehicles are equal, and the solution partitioning solver (SPS) was able to solve CVRPs with arbitrary capacities. For their experiments, the authors used Leap, D-Wave's cloud-based quantum service for application development. The presented methods were compared on benchmark and novel datasets to the well-known metaheuristics for solving VRP and CVRP, such as simulated annealing, evolutionary annealing, and recursive DBSCAN with simulated annealing. According to their results, the authors could state that the introduced hybrid methods were able to find solutions of similar or even better quality than the ones found by classical algorithms.

The problem of scheduling the charging processes of a fleet of electric vehicles (EVs) was addressed by Yu and Nabil [180]. The task was to charge N_{bus} EV buses on N_{station} charging stations where charging is only possible in predefined time windows distributed over the day. The objective was to determine the charging schedule for a bus fleet for a complete day which minimizes the total electricity cost, where the electricity prices were given as part of the problem specification. Additionally, the charging process for each bus needed to respect a constraint on the allowed state of charge of the battery to be within 30% and 100% capacity. The authors study various problem instances with varying numbers of buses and charging stations where the maximal problem size consisted of about 220 logical variables for an instance with three buses and two charging stations. In order to incorporate the problem constraints into the QUBO, the authors employed the method

based on the Hubbard-Stratonovich transform described in Sec. 2.3. With this, the problem could be directly embedded on the D-Wave 2000Q hardware without partitioning. Even though the authors could show that they were able to consistently find valid solutions with the QA, the comparison to SA showed that it took more iterations to arrive at a solution and that the solution quality found with QA were not better than those of SA.

Scheduling the loading processes of vessels at an oil refinery was considered in Ossorio-Castillo and Pena-Brage [130]. The problem was formulated as a mixed-integer linear programming (MILP) problem, with both binary and real-valued variables. The authors employed a decomposition and hybrid scheme where the QUBO problem was formulated using binary variables and solved by the D-Wave 2X system. The authors report that benchmark problem instances can be solved with such a framework but also identify and discuss the shortcomings and problems of such an approach.

A factory scheduling problem was presented by Yarkoni et al. [176], derived from a real-world optimization problem involving re-ordering color sequences in a paint shop during the manufacturing of cars. The task was to assign colors to sequences of cars such that all customer orders are fulfilled while simultaneously minimizing the number of color switches within the sequence. In this work, the authors investigated problem instances ranging from 10-3000 cars in a single sequence. The major contribution of this work is that the multi-car paint shop problem described in the paper has a simple Ising model representation, and instances derived directly from real-world data were solved using D-Wave QPUs, hybrid algorithms, and classical algorithms. The results showed that while the quantum hardware and hybrid algorithms were able to provide adequate solutions to the problem for smaller sizes, the performance of these algorithms approached that of a simple greedy algorithm in the large size limit. However, the analysis also concluded that QA is approaching the limit of industrially-relevant problem scales.

Currently, the scheduling-type problems that can be solved on current and near-term QA hardware are already somewhat realistic and not too far away from real-world applications. The special structure of those problems, especially the constraints typically employed, can lead to small problems instances being difficult and even infeasible to solve with classical approaches already (for example Refs. [76, 78, 92]). Such small but hard problem instances may be particularly interesting for quantum approaches, and consequently, scheduling problems seem to be a very promising field for future application of QA to real-world problems.

3.3 Quantum Simulation for Chemistry, Physics and Biology

The original motivation of quantum computing was the realization of a programmable quantum system that is capable of simulating general quantum systems. To simulate the full dynamics of complex molecules, prototyping efficient batteries, simulating photosynthesis processes in organic materials, and other challenging problems, finding the ground state of individual atoms and molecules is a problem characterized by an extremely high complexity due to exponentially growing Hilbert spaces. QA processors, being complicated quantum mechanical systems themselves, can represent an option to find solutions to these complex problems.

Streif et al. [154] followed an approach outlined by Xia et al. [175] to map electronic structure Hamiltonian calculations to a classical spin-glass system, and subsequently obtain a precise estimation of the true ground state energy of the molecule with QA. In their research, they computed the ground state energies of molecular hydrogen (H_2) and lithium hydride (LiH) for various interatomic distances on a D-Wave 2000Q system. To accomplish this, since a direct embedding of the terms containing Pauli matrices in the electronic structure Hamiltonian is not possible, they introduced r ancillary qubits for each of the n qubits of the original Hamiltonian. The factor r is called *scaling factor*. Although the investigated method works in general since it is able to provide energy estimations very close to the exact energies, due to the currently limited Hamiltonian terms in the QPU,

in general the transformations in [175] incur exponential overhead in transforming the quantum simulation to a spin-glass. The authors underlined that larger systems that were difficult to address might become accessible with more efficient embedding techniques and reverse QA.

Mulligan et al. [116] address the protein design problem by using a D-Wave 2000Q system. They expressed the optimization problem as a QUBO in such a way that the QPU could interface with Rosetta, a state-of-the-art design software suite that used to address protein design tasks by using simulated annealing-based heuristics. The quantum design algorithm was compared with a classical exact branch-and-bound algorithm and the classical simulated annealing method used by default by Rosetta. Results were based on 400 design tasks with the number of designable positions ranging from 1 to 40 and the geometric number of rotamers per position ranging from 1.03 to 6.77, so that the tasks could be implemented in the QPU. Additional 5376 higher-dimensional tasks were addressed with the classical strategies, which could not be addressed with the QA-based strategy. While the classical branch-and-bound and simulated annealing strategies scale exponentially with the problem size (as expected), the QA-based strategy exhibits constant time-to-solution. However, for the small problem instances addressable on the D-Wave 2000Q system, the absolute time to arrive at a solution with the QA-based approach was found to be one to four orders of magnitude slower than the classical approaches.

The problem of sampling valid ensemble configurations of dense polymers mixtures was addressed by Micheletti et al. [110]. The problem was formulated as sampling self-avoiding walks and rings on a predefined lattice for dilute to dense polymer concentrations. The QUBO formulation incorporated terms for enforcing the number of monomers and bonds, the chain connectivity as well as self-avoidance, enabling direct control over the types of polymers to consider. The authors empirically observed an exponential decrease in the success rate of finding valid ensemble samples with increasing problem size for classical simulated annealing. In contrast, the hybrid quantum-classical QA approach tested did not suffer from the same deficiencies.

Harris et al. [69] used a D-Wave quantum annealer to simulate the phases of a correlated electron system and investigate the properties of a quantum magnet modelled by the transverse-field Ising Hamiltonian. In general, these systems are difficult to simulate with limited computational resources. The authors tuned the amount of frustration in a three dimensional cubic lattice of dimensions $L \times L \times L$ and varied the transverse field in order to observe phase transitions between a paramagnetic, an antiferromagnetic, and a spin-glass phase. Experiments were conducted for three system sizes $N = L^3$, with $L = \{4, 6, 8\}$, and results were based on ensembles of up to 1000 random instances, where each instance corresponded to specific sets of ferromagnetic/antiferromagnetic interactions between the spins. The results compared well to the theoretic results and thus validated that the transverse-field Ising model can be used in an integrated circuit as a materials physics simulator, thanks to the ability to manipulate individual spins and bonds.

King et al. [85] also simulated a transverse-field Ising model to investigate the a Kosterlitz-Thouless (KT) phase transition, a specific type of topological phenomenon on frustrated lattices. In particular, the authors used a D-Wave QPU to simulate a fully-frustrated square-octagonal and triangular lattice with cylindrical boundary conditions of size $L \times L$. These lattices were embedded onto the Chimera connectivity graph of a D-Wave's 2000Q processor. Since the translation of the geometry of the lattice onto the Chimera topology leaves necessarily unused qubits along the outside boundary of the processor, the largest instance studied, for $L = 15$, used 1800 of the 2048 qubits available in the processor. By using a reverse annealing protocol, the authors could successfully realize the exotic KT phase of matter. The flux qubit implementation on the D-Wave hardware agreed with theoretical predictions and quantum Monte Carlo simulation over a range of system sizes and Hamiltonian parameters, thus highlighting the potential of QA processors to simulate systems of which classical simulations are hard or even intractable.

Biology is another field characterized by high-complexity dynamics. The first application of QA to real biological data was proposed by Li et al. [98], who explored the possibility of using the D-Wave 2X processor to solve the simplified formulation of a biologically relevant transcription factor (TF)-DNA binding problem. Given a transformed feature vector that represents a DNA sequence, their goal was to compute a predicted binding score that best matches the actual binding score. Although the D-Wave 2X processor used for this study had 1098 functional qubits, because of a sparse connectivity between qubits, only about 40 features could actually be implemented on the device. For small training sizes, the quantum annealer performed better than its classical counterparts (simulated annealing, simulated quantum annealing, multiple linear regression, LASSO, and extreme gradient boosting) for classification, and competitively for ranking tasks. However, the authors conclude that for larger datasets classical algorithms may be a better choice.

In Ref. [172], the authors studied the implementation of kernel-based support vector machines (SVMs) on a D-Wave 2000Q quantum annealer, by representing the objective function of the SVM as a QUBO. They applied both the classical SVM and their version of SVM using QA to synthetic two-dimensional data and real data from biology experiments. To measure the classification performance, they used an experimental dataset with 4352 training samples and 454 test samples. The results showed that the ensemble of classifiers produced by the quantum annealer was superior to the single classifier obtained by the classical SVM for the same computational problem. Therefore, the quantum version of SVM has potential in practical application for classification problems in computational biology, especially when the dataset is rather small.

3.4 Quantum Machine Learning

Quantum machine learning refers to quantum algorithms that are used to solve machine learning tasks, improving and often accelerating classical machine learning techniques. For instance, the outcome of the measurement of a qubit can represent the result of a binary classification task. Here we discuss different machine learning problems that have been addressed by making use of QA: classification [114] and reinforcement learning problems [121], cluster analysis [93, 120], and matrix factorization [59, 128, 132]. For an introduction on the general topic, see, for example, Refs. [111, 118].

Early applications of QA to classification problems can be found in Refs. [122, 123], where the authors propose a boosting-based algorithm and compare it to the AdaBoost approach on simple benchmark examples. The authors conclude that the proposed QA based approach shows advantages over AdaBoost as the generalization error is lower and the classification is faster as it involved less weak learners and fewer boosting steps.

Mott et al. [114] implemented a classifier on classical and quantum annealing QPUs to solve a Higgs particle signal-versus-background machine learning optimization problem. The authors built a set of weak classifiers based on the kinematic observables of the Higgs boson signature through photon decay, which was then combined to construct a strong classifier. Implemented on the D-Wave 3X system, the optimization involved 36 logical binary variables used for the weak classifiers, and 432 physical qubits. The study could show that for very small sizes of the training dataset the quantum annealer-trained classifiers had some advantage over traditional tree- and deep neural network-based machine learning methods, but still comparable to simulated annealing. For larger training sets, the performance of simulated as well as quantum annealing was inferior to traditional methods.

An example of QA in the area of reinforcement learning was provided by Neukart et al. [121], who showed how to partially embed policy evaluation for discrete state spaces on a D-Wave QPU, and to find quantum-classical algorithms capable of dealing with continuous state spaces. By taking the card game blackjack as an example, their aim was to approximate the optimal policy under

consideration of all states from a certain number of complete episodes. From 2 to 7 sub-optimal state-value functions were generated, using 100-500 episodes each. It was shown that quantum-enhanced Monte Carlo policy evaluation allows for finding equivalent or better state-value functions for a given policy with the same number episodes compared to a purely classical Monte Carlo algorithm. However, the authors make clear that the target of their study is to explain how to represent and solve parts of the concerned problems by using QPU, and not to prove an advantage over every existing classical policy evaluation algorithm.

A quantum-assisted clustering algorithm (QACA) was introduced by Neukart et al. [120]. The algorithm is described as showing similarities with the self-organizing feature map, in the sense that the topological properties of the D-Wave QPU are exploited for cluster assignments. Moreover, the algorithm is developed in such a way that a singular embedding, without splitting the QUBO, is possible. The Iris data set (150 points) was used for verification and QACA was compared to expectation minimization, k-means, and self-organizing feature map approaches. Each data sample had three features represented by three logical qubits, which lead to a total of 450 logical qubits. It was shown that QACA can compete with the classical algorithms in terms of accuracy. However, the quality of the results strongly depend on the cluster-form, insinuating that improved techniques for cluster-form initialization must be developed. The authors conclude that the main advantage of QACA is that it does not require training in contrast to classical algorithms, but the main disadvantage is given by necessary embedding onto the QPU.

A similar approach was taken by Kumar et al. [93], which compared k -means clustering with one-hot encoding and binary clustering for problem instances with up to 2000 data samples and two to six clusters to be clustered.

D-Wave quantum annealers have been also used to perform matrix factorization. O'Malley et al. [128] focused on learning features in a set of facial images based on non-negative binary matrix factorizations (NBMF), and developed an algorithm implemented on a D-Wave 2X QPU. Given a matrix V , the NBMF looks for a decomposition of the matrix $V = WH$ such that W is composed of non-negative floating point numbers and H is binary. To do so, the authors used an alternating least-squares algorithm [99] where QA is used for finding the matrix H in such a way that, given a matrix V , the norm between V and the matrix product $W \cdot H$ is minimized. Although H can be found by solving a series of linear least squares problems in binary variables, the same problem can be readily solved as a QUBO. The analysis was limited to 35 binary variables, each represented by at most 19 physical qubits, leading to a total of 660 physical qubits in the worst case scenario. In this particular study, QA outperforms two classical approaches (qbsolv and Gurobi [64]) in a benchmark that was not tailored to the QPU as other problems have been. Moreover, although QA does not outperform the best classical heuristics in general, it can be considered the best choice in case only a short computational time is available.

The same authors [59] as well as other researchers [132] addressed matrix factorization with the D-Wave 2000Q system by exploiting its reverse annealing feature. Reverse annealing performs series of local searches that refine existing solutions obtained through forward annealing. Golden and O'Malley [59] presented an algorithm to overcome the downside of the QA approach, which was that improvement in solution quality from iteration to iteration quickly plateaus due to the global search nature of forward annealing search. It was observed that, in terms of the quality of the factorization given a fixed time, reverse annealing performs well once the error of the factorization is set sufficient low, i.e., lower than the plateau reached with forward annealing, leading to an average improvement on the quality of the NBMF factorization by 12%. However, the exact nature of the matrix being factorized appeared to play a role in determining how effective the algorithm is, which highlights the need for further investigation on the topic. Ottaviani and Amendola [132] went one step further, by addressing the same decomposition problem as in [128], but where neither

W nor H are binary matrices. The ultimate goal was to make it possible to use QA in real-world applications that do not directly involve binary variables. They also introduced an adaptive strategy to maximize the efficiency of the reverse annealing. The algorithm searches for the right value of the mean of Hamming distances by adapting the holding time parameter of the reverse annealing search. Results showed that the proposed method was able to find the global minimum point and required much less iterations to reach good factorizations compared to classical algorithms. Although a very limited number of variables was considered, this study showed that possible applications for QA could include more variable types than strictly binary.

So far, no clear advantage of QA-based approaches over classical machine learning could be demonstrated, but some aspects are promising. However, many of the favorable features of QA approaches only manifest themselves if specific conditions are met, for example, sparsity of the data or small data set sizes.

3.5 Finance

Many applications of quantum computing and in particular QA are proposed for the finance domain [129], but approaches that actually use real QA hardware are rare.

Portfolio optimization represents a real-world application area in finance which one could use to benchmark QA. In particular, QA has been applied to a well-defined class of unconstrained optimization problems derived from the application of Markowitz portfolio theory [103]. Markowitz portfolio optimization is a variant of binary optimization tasked with selecting the subset of investment assets expected to yield the highest return value and minimal risk and transaction costs, including temporary and permanent market impact. Several versions of portfolio optimization problems have been used to benchmark QA performance [50, 61, 146, 164].

Rosenberg et al. [146] demonstrated several encodings of a discrete multi-period Markowitz portfolio optimization formulation using D-Wave Systems QPUs (D-Wave Two and D-Wave 2X systems), and found promising initial results for the probability to find the optimal portfolio. Mean-variance portfolio optimization problems are traditionally solved as continuous-variable problems, while in this case a discrete solution is necessary. The portfolio optimization problem was written as a quadratic integer optimization problem, where integer variables were recast as binary variables, and the constraints were incorporated into the objective function as a penalty term. The study presents a double encoding: the first one from integer to binary variables (binary/unary encoding), the second and standard one from the logical graph to the physical graph, using 584 physical qubits to represent up to logical 72 variables. The analysis highlighted a level of intrinsic noise that manifests as a misspecification of the optimization problem. The authors conclude more qubits would allow the use of an integer encoding scheme that is less sensitive to noise levels.

Elsokkary et al. [50] also tested whether the use of a D-Wave One QPU allowed for a meaningful improvement in computational performance for solving the Markowitz portfolio for data gathered from the Abu Dhabi Stock Exchange website in Abu Dhabi, United Arab Emirates. They formulated – with some relaxations since fractional shares of stocks cannot be purchased – the Markowitz portfolio model as a QUBO and compared the solutions found with QA to the ones obtained through a MATLAB’s genetic algorithm solver. The data set was too large for direct embedding, but `qbsolv` showed good agreement with the MATLAB-derived solutions.

Venturelli and Kondratyev [164] benchmarked a similar mean-variance model of portfolio optimization using a hybrid quantum-classical method that coupled quantum annealing with a genetic algorithm (GA). The authors found that the best approach was to seed a D-Wave 2000Q quantum annealer with a candidate solution found by a greedy local search and then perform a reverse annealing protocol. Since all the instances were also solvable by an industry-established genetic algorithm, they used the latter as the classical benchmark. From the results comparing GA, QA

and reverse QA, the best solver was reverse QA at minimum annealing time and pause time. In the median case, one to three orders of magnitude speed-up was observed when applying the proposed hybrid algorithm with respect to forward annealing alone.

Mugel et al. [115] performed portfolio optimization using the D-Wave Advantage QPU, which features a qubit connectivity based on the Pegasus topology, i.e., 5000 qubits and 15-way qubit connectivity. The authors also showcased the performance of a D-Wave proprietary hybrid algorithm available in the Leap quantum cloud service, called the Hybrid Solver Service (HSS), which had been used in previous studies as well [178]. Mugel et al. [115] solved the portfolio optimization problem for two indices, the Nikkei225 and the S&P500, and compared the results of HSS with simulated annealing, a genetic algorithm, the Gurobi solver [64], and a LocalSolver [15], which combines exact and heuristic techniques. Among these solvers, LocalSolver performed the best and found the optimal solution very quickly in all cases. In terms of TTS, HSS was able to find a solution within 3% of the optimal solution for the S&P500 index up to 400 stocks.

More recently, Grant et al. [61] presented a study to benchmark QA controls and their relative effects on computational accuracy using portfolio optimization. They evaluated forward and reverse annealing methods and identified control variations that yield optimal performance in terms of probability of success and probability of chain breaks. They focused on the controls during the phases of pre-processing (Hamiltonian definition and embedding onto the quantum annealer), annealing (control on the time-dependent physics of the device and underlying quantum state), and post-processing (readout and decoding of observed results). In particular, a variety of quantum annealing controls were explored: the embedding algorithm, the forward annealing time, the number of spin reversal transforms, reverse annealing initial state, annealing pause point and duration, and others. The novelty of this paper is that the authors provided insights into the best selections of controls for solving binary optimization problems with the highest accuracy that may help guide the future use of QA as a metaheuristics for optimization of industrially relevant applications.

In summary, it is not clear whether QA will be the most compelling solver for finance applications, but as technology and theory advance, it could be a competitive choice. These results also highlight the importance of hybrid quantum/classical approaches that allow for solving much larger problems than those that can be embedded in the topology of current quantum processors.

3.6 Final remarks on QA for real-world applications

In summary, quantum annealing methods have been tested for numerous applications, from traffic flow optimization and job-shop scheduling to machine learning, portfolio optimization, protein folding, molecular similarity, and computational biology. The various optimization problems presented here share the solution strategy of formulating QUBO problems, which imply freely programmable and typically fully connected interactions between Ising variables. The mapping from these dense logical graphs to the sparse connectivity in QPUs prevents efficient computation in the current generation quantum annealers. This is probably the most crucial and common bottleneck of QA for real-world applications.

What we can learn from current applications of QA to traffic flow optimization is that all problems are based on two main models: either every variable is represented by a node of a graphical network and optimization is directly performed on the network itself, or decision variables represent properties of the graphical network and constraints are imposed on these logical variables. All traffic flow optimization problems consider paths and congestion. However, time structure is often missing, which is crucial if one wants to consider real-world application cases. On the other hand, we can state that scheduling applications are closer to real-world needs. Problems that are not far from what we need in practical life are already addressed, even though still limited by the available

hardware size. A similar problem arises in financial applications. Indeed, what to do in real financial problems is beyond the capabilities of quantum annealers. Moreover, the time required to execute these algorithms is much longer than that required by financial traders to be considered useful. Due to this timing delay, it is impossible to solve real-world finance problems directly using QPUs, and some mathematical modeling is therefore required. This in turn opens the possibility of using classical competition algorithms, making it difficult to argue how QA can offer an advantage.

Conversely, in the field of chemistry, simulation, or ML, the timescales of QA offer an advantage rather than a bottleneck. Sampling from complex crystal structure ground states or mathematical models useful for machine learning algorithms is hard, and long timescales are typical classically. However, the scaling of the relevant terms for impactful applications in these fields is disadvantageous to QPUs. For ML, it is standard to have many layers of densely connected variables, something that is directly orthogonal to the regular sparse topology of current quantum devices. While there is more potential for quantum simulation, as evidenced by recent works in which D-Wave QPUs successfully simulate previously intractable quantum crystal models, the interaction terms and limited size of the QPUs offer strict limitations in the kinds of simulations that QPUs can execute. Furthermore, decomposing arbitrary quantum simulations into simpler models incurs an exponential overhead in the number of qubits, leading to unfavorable scaling of the problem size.

While it is clear that realistic applications using QPUs are still far, it is important to understand the different limiting factors. Most obviously, the number of qubits and couplers in current QPUs is low relative to the number of variables typically used in each of the application domains. An ideal practical problem needs to be small enough for significant parts of it to fit on current hardware, but also large and hard enough that it cannot be solved quickly and efficiently by a purely classical solver. Moreover, QA is a heuristic algorithm, which means that a QPU is able to provide approximations of optimal solutions with each annealing cycle, but may require a much larger number of anneals to find a global optimum. In the worst case, it is not even always guaranteed to get feasible solutions at all. Finally, QA benefits from a classical support framework that provides tools for transforming, decomposing, and embedding inputs. However, there is still much room for improvement, both in terms of performance and scope. In the near future, it is hoped that hybrid methods will outperform pure classical or quantum methods in some – application-specific – cases. Together with improvements in QPU size and performance and more efficient infrastructure support, the development of efficient and effective hybrid algorithms will hopefully combine the best features of classical and quantum processors: finding optimal solutions and doing so quickly.

4 LESSONS LEARNED ABOUT QUANTUM ANNEALING

The main goal of this work is to provide researchers and practitioners from various fields with a condensed source explaining the principles of QA, as well as summarizing the shortcomings and potential benefits of QA in view of real-world application problems. Considering the body of work presented and analyzed, some basic insights can be formulated, which should be kept in mind when considering the application of QA:

- **Quantum annealing can only address specific problem formulations.** Any problem intended to be solved by a practitioner with QA needs to be formulated as an Ising model or QUBO problem, where spin or binary variables are used to encode the problem. Currently, state-of-the-art quantum annealers can only encode linear and quadratic terms, and therefore all terms and constraints in the objective function must be implemented in that form. Furthermore, QPUs have fixed topologies of qubit layouts, which either restrict the connectivity of applicable problems, or require minor-embedding/hybrid techniques to overcome. In short: while the QA algorithm is in theory intended to solve NP-hard problems, in practice, arbitrary

problems may incur significant polynomial (or even exponential) overhead when transformed to Ising or QUBO.

- **Only some problems are good candidates for quantum annealing.** Not all problems are currently good fits for QA, and it is difficult to know *a priori* if any given real-world problem will benefit from QA. Some common properties of problems that were not good fits for QA are: continuous optimization or high-precision problems, problems known to be in the computational complexity class P , decision problems, and fully-connected graph problems. However, problems that did seem to exhibit good results with QA typically have simple combinatorial properties and/or constraints where the number and connectivity of the qubits required to represent individual contributions of the optimization problem do not scale strongly with the problem size.
- **Quantum annealing is a heuristic quantum optimization algorithm.** Although inspired by Adiabatic Quantum Computation, QA is implemented with more relaxed conditions, thus removing the theoretical guarantees of the adiabatic theorem in favor of near-term applicability. The result is a heuristic quantum optimization algorithm that exploits quantum effects for computation. There is reason to believe that quantum fluctuations are effective for transitions between spin configurations during the annealing, thus aiding global optimization; however, conclusive evidence has remained elusive despite rigorous research. For all practical purposes, QA is therefore a stochastic black-box optimization approach for combinatorial optimization, similar to simulated thermal annealing.
- **Hybrid algorithms are currently necessary.** Ising models or QUBO instances arising from real-world problems typically require more complicated logical variables, constraint terms, or interactions than can be realized with current hardware QA systems. This has inspired development of hybrid quantum-classical algorithms to overcome some of these limitations. Although there are many ways to construct such algorithms, for real-world problems reviewed in this article, problem partitioning is the most common approach. Problems are decomposed into sub-problems that are individually submitted to the QPU, with their respective results stitched together. Classical search and optimization routines are used to construct the outer loops in hybrid algorithms that orchestrate the sub-problem generation and solution recombination in an attempt to solve the original full problem.
- **Quantum annealing can only reach its full potential through quantum hardware.** Simulating quantum systems with classical hardware is in general inefficient and sometimes even intractable. There is evidence that QA implemented on D-Wave QPUs can simulate some quantum mechanical models more efficiently than classical algorithms. This, however, was strongly dependent on mapping intrinsic properties of the problem directly to some core functionality implemented in the quantum hardware (e.g., reverse annealing and pause/ramp of the transverse field).

5 CONCLUSIONS

Quantum annealing (QA) is a heuristic optimization algorithm used to solve problems in combinatorial optimization. It can be motivated by the idea of utilizing quantum fluctuations for an annealing process through which heuristic optimization is performed. Recently, advances in quantum technologies have enabled the development of quantum processors that implement the QA algorithm for programmable use in hardware. The first commercially available hardware was provided by D-Wave Systems Inc. and as a result, this platform is used most frequently to solve optimization problems across different disciplines. Quantum annealing processors – in their current form – have been developed specifically to be easier to build and operate compared to gate-based quantum computers. However, many questions and challenges still remain open, and continuous

efforts are made to evaluate and extend QA hardware and its efficient utilization for addressing real-world application problems. The ultimate question of whether QA offers a provable advantage over classical approaches has yet to be answered.

In this paper, we first provided a general presentation of QA by explaining the foundations of this heuristic quantum optimization algorithm and the software and hardware characterizing present quantum processors. These concepts are essential to evaluate the potential and limitations of the method based on the state-of-the-art studies. We reviewed some example applications and empirical studies to provide the reader with updated information about the current characteristics and the applications status of this relatively new quantum technology, such as what problems can be addressed, what problem sizes are allowed, what is the expected scaling behavior, what constraints can be represented, etc. The classification of this research was mainly organized according to the most relevant application fields.

In summary, based on all the works reviewed in this article, the empirical performance of the QA algorithm is highly dependent on the problem being addressed. The results are influenced by the interactions between the problem formulation, the encoding of the logical variables into qubits, and the embedding in the hardware platform. Regardless of the application field, only relatively small problem sizes can currently be addressed efficiently on actual hardware. In addition to the small number of physical qubits, a major limiting factor is the overhead resulting from the restricted connection topology of the physical qubits that is currently realizable in hardware. Typically, the number of physical qubits needed to address a problem scales polynomially with the number of logical variables in the problem formulation. However, after the careful analysis of a given problem class and its mapping to QA hardware, the hardware can be programmed for that problem class and then each instance of that problem can be solved robustly in a fixed amount of time, which is a major advantage of the quantum annealing approach.

In order to take full advantage of QA's potential, it is clear that the unique properties of the QA hardware must be exploited in ways that classical software cannot compensate for. This indicates that novel features such as additional qubit control, driver Hamiltonians and operators, and higher connectivity must be developed and implemented in hardware for QA to be applicable in practice. The fast development of current and future hardware systems fuels the hope that exciting developments can be expected in the near to mid term and more realistic application problems can be addressed with quantum annealing.

ACKNOWLEDGMENTS

The authors thank Markus Olhofer, Andrea Skolik, and Michele Cattelan for fruitful discussions and feedback. This work is part of the research programme C2D–Horizontal Data Science for Evolving Content with project name DACCOMPLI and project number 628.011.002, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

REFERENCES

- [1] [n. d.]. D-Wave System Documentation: QPU-Specific Characteristics. See documents at https://docs.dwavesys.com/docs/latest/doc_physical_properties.html, last checked November 2021.
- [2] [n. d.]. D-Wave Systems has produced an open-source software stack in Python (Ocean tools) for accessing its quantum hardware, formulating problems for execution, and classical QUBO/Ising solvers. More information can be found at <https://docs.ocean.dwavesys.com/en/stable/>, last checked November 2021.
- [3] [n. d.]. Problem-Solving Handbook, QPU Solvers: Minor-Embedding. Available at https://docs.dwavesys.com/docs/latest/handbook_embedding.html, last checked November 2021.
- [4] Juan I Adame and Peter L McMahon. 2020. Inhomogeneous driving in quantum annealers can result in orders-of-magnitude improvements in performance. *Quantum Science and Technology* 5, 3 (2020), 035011. <https://doi.org/10.1088/2058-9565/ab935a>

- [5] Dorit Aharonov, Wim van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. 2008. Adiabatic Quantum Computation Is Equivalent to Standard Quantum Computation. *SIAM Rev.* 50, 4 (2008), 755–787. <https://doi.org/10.1137/080734479> arXiv:<https://doi.org/10.1137/080734479>
- [6] Tameem Albash and Daniel A. Lidar. 2018. Adiabatic quantum computation. *Rev. Mod. Phys.* 90 (Jan 2018), 015002. Issue 1. <https://doi.org/10.1103/RevModPhys.90.015002>
- [7] Evgeny Andriyash and Mohammad H. Amin. 2017. Can quantum Monte Carlo simulate quantum annealing? *arXiv:1703.09277* (2017).
- [8] B. Apolloni, N. Cesabianchi, and D. De Falco. 1988. A Numerical Implementation of Quantum Annealing. In *International Conference on Stochastic Processes, Physics and Geometry*. Ascona-Locarno, 97–111.
- [9] Maliheh Aramon, Gili Rosenberg, Elisabetta Valiante, Toshiyuki Miyazawa, Hirotaka Tamura, and Helmut G. Katzgraber. 2019. Physics-Inspired Optimization for Quadratic Unconstrained Problems Using a Digital Annealer. *Frontiers in Physics* 7 (2019), 48. <https://doi.org/10.3389/fphy.2019.00048>
- [10] Rolando Armas, Hernán Aguirre, Fabio Daolio, and Kiyoshi Tanaka. 2017. Evolutionary design optimization of traffic signals applied to Quito city. *PLOS ONE* 12, 12 (12 2017), 1–37. <https://doi.org/10.1371/journal.pone.0188757>
- [11] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (01 Oct 2019), 505–510. <https://doi.org/10.1038/s41586-019-1666-5>
- [12] Francisco Barahona. 1982. On the computational complexity of Ising spin glass models. *Journal of Physics A: Mathematical and General* 15, 10 (1982), 3241.
- [13] Paul Benioff. 1980. The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines. *Journal of Statistical Physics* 22, 5 (1980), 563–591. <https://doi.org/10.1007/BF01011339>
- [14] Paul Benioff. 1982. Quantum Mechanical Models of Turing Machines That Dissipate No Energy. *Physical Review Letters* 48, 23 (1982), 1581–1585. <https://doi.org/10.1103/PhysRevLett.48.1581>
- [15] Thierry Benoist, Bertrand Estellon, Frédéric Gardi, Romain Megel, and Karim Nouioua. 2011. LocalSolver 1.x: a black-box local-search solver for 0-1 programming. *4OR* 9 (2011), 299. See the website for current information: <https://www.localsolver.com>.
- [16] Jesse Berwald, Nicholas Chancellor, and Raouf Dridi. 2021. Understanding domain-wall encoding theoretically and experimentally. *arXiv:2108.12004* (2021).
- [17] Jacob D. Biamonte and Peter J. Love. 2008. Realizable Hamiltonians for universal adiabatic quantum computers. *Phys. Rev. A* 78 (2008), 012352. Issue 1. <https://doi.org/10.1103/PhysRevA.78.012352>
- [18] Zhengbing Bian, Fabian Chudak, Robert Brian Israel, Brad Lackey, William G. Macready, and Aidan Roy. 2016. Mapping Constrained Optimization Problems to Quantum Annealing with Application to Fault Diagnosis. *Frontiers in ICT* 3 (2016), 14. <https://doi.org/10.3389/fict.2016.00014>
- [19] Michael Booth, Steven P Reinhardt, and Aidan Roy. 2017. Partitioning optimization problems for hybrid classical/quantum execution. *Technical Report* (2017), 01–09.
- [20] Kelly Boothby, Paul Bunyk, Jack Raymond, and Aidan Roy. 2020. Next-Generation Topology of D-Wave Quantum Processors. arXiv:arXiv:2003.00133 <https://arxiv.org/abs/2003.00133>
- [21] Tomas Boothby, Andrew D. King, and Aidan Roy. 2016. Fast clique minor generation in Chimera qubit connectivity graphs. *Quantum Information Processing* 15, 1 (2016), 495–508. <https://doi.org/10.1007/s11128-015-1150-6>
- [22] M. Born and V. Fock. 1928. Beweis des Adiabatsatzes. *Zeitschrift für Physik* 51, 3 (March 1928), 165–180. <https://doi.org/10.1007/BF01343193>
- [23] Michał Borowski, Paweł Gora, Katarzyna Karnas, Mateusz Błajda, Krystian Król, Artur Matyjasek, Damian Burczyk, Miron Szewczyk, and Michał Kutwin. 2020. New Hybrid Quantum Annealing Algorithms for Solving Vehicle Routing Problem. In *Computational Science – ICCS 2020 (Lecture Notes in Computer Science)*, Valeria V. Krzhizhanovskaya, Gábor Závodszy, Michael H. Lees, Jack J. Dongarra, Peter M. A. Sloot, Sérgio Brissos, and João Teixeira (Eds.). Springer International Publishing, Cham, 546–561. https://doi.org/10.1007/978-3-030-50433-5_42

- [24] Peter Brucker, Bernd Jurisch, and Bernd Sievers. 1994. A Branch and Bound Algorithm for the Job-Shop Scheduling Problem. *Discrete Applied Mathematics* 49, 1 (March 1994), 107–127. [https://doi.org/10.1016/0166-218X\(94\)90204-6](https://doi.org/10.1016/0166-218X(94)90204-6)
- [25] Maximilian Böther, Leon Schiller, Philipp Fischbeck, Louise Molitor, Martin S. Krejca, and Tobias Friedrich. 2021. Evolutionary minimization of traffic congestion. 937–945. <https://doi.org/10.1145/3449639.3459307>
- [26] Jun Cai, William G. Macready, and Aidan Roy. 2014. A practical heuristic for finding graph minors. arXiv:arXiv:1406.2741
- [27] Jun Cai, William G. Macready, and Aidan Roy. 2014. A Practical Heuristic for Finding Graph Minors. arXiv:1406.2741 [quant-ph] (June 2014). arXiv:1406.2741 [quant-ph]
- [28] Nicholas Chancellor. 2017. Modernizing quantum annealing using local searches. *New Journal of Physics* 19, 2 (Feb. 2017), 023024. <https://doi.org/10.1088/1367-2630/aa59c4>
- [29] Nicholas Chancellor. 2019. Domain wall encoding of discrete variables for quantum annealing and QAOA. *Quantum Science and Technology* 4, 4 (Aug. 2019), 045004. <https://doi.org/10.1088/2058-9565/ab33c2>
- [30] Jie Chen, Tobias Stollenwerk, and Nicholas Chancellor. 2021. Performance of Domain-Wall Encoding for Quantum Annealing. *IEEE Transactions on Quantum Engineering* 2 (2021), 1–14. <https://doi.org/10.1109/TQE.2021.3094280>
- [31] S. Chien, M. Johnston, J. Frank, M. Giuliano, A. Kavelaars, C. Lenzen, and N. Policella. 2012. A Generalized Timeline Representation, Services, and Interface for Automating Space Mission Operations. <https://doi.org/10.2514/6.2012-1275459>
- [32] Vicky Choi. 2008. Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. *Quantum Information Processing* 7, 5 (2008), 193–209. <https://doi.org/10.1007/s11128-008-0082-9>
- [33] Vicky Choi. 2011. Minor-Embedding in Adiabatic Quantum Computation: II. Minor-Universal Graph Design. *Quantum Information Processing* 10, 3 (June 2011), 343–353. <https://doi.org/10.1007/s11128-010-0200-3>
- [34] Vicky Choi. 2021. Essentiality of the Non-stoquastic Hamiltonians and Driver Graph Design in Quantum Optimization Annealing. arXiv:arXiv:2105.02110
- [35] Isaac L. Chuang, Lieven M. K. Vandersypen, Xinlan Zhou, Debbie W. Leung, and Seth Lloyd. 1998. Experimental realization of a quantum algorithm. *Nature* 393, 6681 (01 May 1998), 143–146. <https://doi.org/10.1038/30181>
- [36] James Clark, Tristan West, Joseph Zammit, Xiaohu Guo, Luke Mason, and Duncan Russell. 2019. Towards Real Time Multi-Robot Routing Using Quantum Computing Technologies. In *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region (Guangzhou, China) (HPC Asia 2019)*. Association for Computing Machinery, New York, NY, USA, 111–119. <https://doi.org/10.1145/3293320.3293333>
- [37] D. Coppersmith. 2002. An approximate Fourier transform useful in quantum factoring. arXiv:arXiv:quant-ph/0201067
- [38] Daniel Crawford, Anna Levit, Navid Ghadermarzy, Jaspreet S. Oberoi, and Pooya Ronagh. 2019. Reinforcement Learning Using Quantum Boltzmann Machines. arXiv:1612.05695 [quant-ph] (Jan. 2019). arXiv:1612.05695 [quant-ph]
- [39] Elizabeth Crosson, Tameem Albash, Itay Hen, and A. P. Young. 2020. De-Signing Hamiltonians for Quantum Adiabatic Optimization. *Quantum* 4 (Sept. 2020), 334. <https://doi.org/10.22331/q-2020-09-24-334>
- [40] D-Wave Systems. 2017. Reverse Quantum Annealing for Local Refinement of Solutions. *D-Wave Whitepaper Series* (2017).
- [41] Arnab Das and Bikas K. Chakrabarti. 2008. Colloquium: Quantum annealing and analog quantum computation. *Rev. Mod. Phys.* 80 (Sep 2008), 1061–1081. Issue 3. <https://doi.org/10.1103/RevModPhys.80.1061>
- [42] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe. 2016. Demonstration of a small programmable quantum computer with atomic qubits. *Nature* 536, 7614 (01 Aug 2016), 63–66. <https://doi.org/10.1038/nature18648>
- [43] Vasil S. Denchev, Sergio Boixo, Sergej V. Isakov, Nan Ding, Ryan Babbush, Vadim Smelyanskiy, John Martinis, and Hartmut Neven. 2016. What is the Computational Value of Finite-Range Tunneling? *Phys. Rev. X* 6 (Aug 2016), 031015. Issue 3. <https://doi.org/10.1103/PhysRevX.6.031015>
- [44] David Elieser Deutsch and Roger Penrose. 1989. Quantum Computational Networks. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 425, 1868 (1989), 73–90. <https://doi.org/10.1098/rspa.1989.0099>
- [45] N G Dickson, M W Johnson, M H Amin, R Harris, F Altomare, A J Berkley, P Bunyk, J Cai, E M Chapple, P Chavez, F Cioata, T Cirip, P deBuen, M Drew-Brook, C Enderud, S Gildert, F Hamze, J P Hilton, E Hoskinson, K Karimi, E Ladizinsky, N Ladizinsky, T Lanting, T Mahon, R Neufeld, T Oh, I Perminov, C Petroff, A Przybysz, C Rich, P Spear, A Tcaciuc, M C Thom, E Tolkacheva, S Uchaikin, J Wang, A B Wilson, Z Merali, and G Rose. 2013. Thermally assisted quantum annealing of a 16-qubit problem. *Nature Communications* 4, 1 (May 2013). <https://doi.org/10.1038/ncomms2920>
- [46] Reinhard Diestel. [n. d.]. *Graph Theory*. Springer.
- [47] Florian Dolde, Ville Bergholm, Ya Wang, Ingmar Jakobi, Boris Naydenov, Sébastien Pezzagna, Jan Meijer, Fedor Jelezko, Philipp Neumann, Thomas Schulte-Herbrüggen, Jacob Biamonte, and Jörg Wrachtrup. 2014. High-fidelity spin entanglement using optimal control. *Nature Communications* 5, 1 (28 Feb 2014), 3371. <https://doi.org/10.1038/ncomms4371>

- [48] Adam Douglass, Andrew D. King, and Jack Raymond. 2015. Constructing SAT Filters with a Quantum Annealer. In *Theory and Applications of Satisfiability Testing – SAT 2015 (Lecture Notes in Computer Science)*, Marijn Heule and Sean Weaver (Eds.). Springer International Publishing, Cham, 104–120. https://doi.org/10.1007/978-3-319-24318-4_9
- [49] David J. Earl and Michael W. Deem. 2005. Parallel Tempering: Theory, Applications, and New Perspectives. *Physical Chemistry Chemical Physics* 7, 23 (Nov. 2005), 3910–3916. <https://doi.org/10.1039/B509983H>
- [50] Nada Elsokkary, Faisal Shah Khan, Davide La Torre, Travis S Humble, and Joel Gottlieb. 2017. *Financial portfolio management using D-Wave quantum optimizer: The case of Abu Dhabi Securities Exchange*. Technical Report. Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States).
- [51] Evgeny Andriyash, Zhengbing Bian, Fabian Chudak, Marshall Drew-Brook, Andrew D. King, William G. Macready, Aidan Roy. 2016. Boosting integer factoring performance via quantum annealing offsets.
- [52] Yan-Long Fang and P. A. Warburton. 2020. Minimizing minor embedding energy: an application in quantum annealing. *Quantum Information Processing* 19 (2020). <https://doi.org/10.1007/s11128-020-02681-x>
- [53] Ahmed Fares and Walid Gomaa. 2015. Multi-Agent Reinforcement Learning Control for Ramp Metering. In *Progress in Systems Engineering*, Henry Selvaraj, Dawid Zydek, and Grzegorz Chmaj (Eds.). Vol. 366. Springer International Publishing, Cham, 167–173. https://doi.org/10.1007/978-3-319-08422-0_25
- [54] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. 2000. *Quantum Computation by Adiabatic Evolution*. Technical Report MIT-CTP-2936, preprint available as arXiv:quant-ph/0001106. Center for Theoretical Physics, Massachusetts Institute of Technology.
- [55] Sebastian Feld, Christoph Roch, Thomas Gabor, Christian Seidel, Florian Neukart, Isabella Galter, Wolfgang Mauerer, and Claudia Linnhoff-Popien. 2019. A Hybrid Solution Method for the Capacitated Vehicle Routing Problem Using a Quantum Annealer. *Frontiers in ICT* 6 (2019), 13. <https://doi.org/10.3389/fict.2019.00013>
- [56] Richard P. Feynman. 1982. Simulating Physics with Computers. *International Journal of Theoretical Physics* 21, 6 (1982), 467–488. <https://doi.org/10.1007/BF02650179>
- [57] Fred Glover, Zhipeng Lü, and Jin-Kao Hao. 2010. Diversification-Driven Tabu Search for Unconstrained Binary Quadratic Problems. *4OR* 8, 3 (Oct. 2010), 239–253. <https://doi.org/10.1007/s10288-009-0115-y>
- [58] Fred W. Glover and Manuel Laguna. 1997. *Tabu Search*. Springer US. <https://doi.org/10.1007/978-1-4615-6089-0>
- [59] John Golden and Daniel O'Malley. 2021. Reverse annealing for nonnegative/binary matrix factorization. *PLOS ONE* 16, 1 (Jan. 2021), e0244026. <https://doi.org/10.1371/journal.pone.0244026>
- [60] Timothy D. Goodrich, Blair D. Sullivan, and Travis S. Humble. 2018. Optimizing Adiabatic Quantum Program Compilation Using a Graph-Theoretic Framework. *Quantum Information Processing* 17, 5 (April 2018), 118. <https://doi.org/10.1007/s11128-018-1863-4>
- [61] Erica Grant, Travis S. Humble, and Benjamin Stump. 2021. Benchmarking Quantum Annealing Controls with Portfolio Optimization. *Physical Review Applied* 15, 1 (Jan. 2021), 014012. <https://doi.org/10.1103/PhysRevApplied.15.014012>
- [62] Lov K. Grover. 1996. A Fast Quantum Mechanical Algorithm for Database Search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing (Philadelphia, Pennsylvania, USA) (STOC '96)*. Association for Computing Machinery, New York, NY, USA, 212–219. <https://doi.org/10.1145/237814.237866>
- [63] Lalit Gupta and Itay Hen. 2020. Elucidating the Interplay between Non-Stoquasticity and the Sign Problem. *Advanced Quantum Technologies* 3, 1 (2020), 1900108. <https://doi.org/10.1002/qute.201900108> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/qute.201900108> preprint at arXiv:1910.13867.
- [64] Gurobi Optimization, LLC. 2021. Gurobi Optimizer Reference Manual. <https://www.gurobi.com>
- [65] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Gael Varoquaux, Travis Vaught, and Jarrod Millman (Eds.). 11–15.
- [66] Nikolaus Hansen. 2006. The CMA Evolution Strategy: A Comparing Review. In *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*, Jose A. Lozano, Pedro Larrañaga, Iñaki Inza, and Endika Bengoetxea (Eds.). Springer, Berlin, Heidelberg, 75–102. https://doi.org/10.1007/3-540-32494-1_4
- [67] David Harris and Sarah Harris. 2012. *Digital Design and Computer Architecture*. Elsevier. <https://doi.org/10.1016/B978-0-12-370497-9.X5000-8>
- [68] R. Harris, M. W. Johnson, T. Lanting, A. J. Berkley, J. Johansson, P. Bunyk, E. Tolkacheva, E. Ladizinsky, N. Ladizinsky, T. Oh, F. Cioata, I. Perminov, P. Spear, C. Enderud, C. Rich, S. Uchaikin, M. C. Thom, E. M. Chapple, J. Wang, B. Wilson, M. H. S. Amin, N. Dickson, K. Karimi, B. Macready, C. J. S. Truncik, and G. Rose. 2010. Experimental Investigation of an Eight-Qubit Unit Cell in a Superconducting Optimization Processor. *Physical Review B* 82, 2 (2010), 024511. <https://doi.org/10.1103/PhysRevB.82.024511>
- [69] R. Harris, Y. Sato, A. J. Berkley, M. Reis, F. Altomare, M. H. Amin, K. Boothby, P. Bunyk, C. Deng, C. Enderud, S. Huang, E. Hoskinson, M. W. Johnson, E. Ladizinsky, N. Ladizinsky, T. Lanting, R. Li, T. Medina, R. Molavi, R. Neufeld, T. Oh, I. Pavlov, I. Perminov, G. Poulin-Lamarre, C. Rich, A. Smirnov, L. Swenson, N. Tsai, M. Volkmann, J. Whittaker, and J. Yao. 2018. Phase transitions in a programmable quantum spin glass simulator. *Science* 361, 6398 (2018), 162–165.

<https://doi.org/10.1126/science.aat2025>

- [70] Andreas Hartmann and Wolfgang Lechner. 2019. Rapid counter-diabatic sweeps in lattice gauge adiabatic quantum computing. 21, 4 (apr 2019), 043025. <https://doi.org/10.1088/1367-2630/ab14a0>
- [71] Philipp Hauke, Helmut G. Katzgraber, Wolfgang Lechner, Hidetoshi Nishimori, and William D. Oliver. 2020. Perspectives of Quantum Annealing: Methods and Implementations. *Reports on Progress in Physics* 83 (March 2020), 054401. <https://doi.org/10.1088/1361-6633/ab85b8> arXiv:1903.06559 [quant-ph]
- [72] Maritza Hernandez and Maliheh Aramon. 2017. Enhancing Quantum Annealing Performance for the Molecular Similarity Problem. *Quantum Information Processing* 16, 5 (April 2017), 133. <https://doi.org/10.1007/s11128-017-1586-y>
- [73] Ting-Jui Hsu, Fengping Jin, Christian Seidel, Florian Neukart, Hans De Raedt, and Kristel Michielsens. 2018. Quantum Annealing with Anneal Path Control: Application to 2-SAT Problems with Known Energy Landscapes. *arXiv:1810.00194 [quant-ph]* (Oct. 2018). arXiv:1810.00194 [quant-ph]
- [74] Kazuki Ikeda, Yuma Nakamura, and Travis S. Humble. 2019. Application of Quantum Annealing to Nurse Scheduling Problem. *Scientific Reports* 9, 1 (Sept. 2019), 12837. <https://doi.org/10.1038/s41598-019-49172-3>
- [75] Daisuke Inoue, Akihisa Okada, Tadayoshi Matsumori, Kazuyuki Aihara, and Hiroaki Yoshida. 2021. Traffic Signal Optimization on a Square Lattice with Quantum Annealing. *Scientific Reports* 11, 1 (Dec. 2021), 3303. <https://doi.org/10.1038/s41598-021-82740-0> arXiv:2003.07527
- [76] Takahiro Ishihara and Steffen Limmer. 2020. Optimizing the Hyperparameters of a Mixed Integer Linear Programming Solver to Speed Up Electric Vehicle Charging Control. In *Applications of Evolutionary Computation 2020*. Springer, LNCS, LNAI, LNBI, 37–53. https://doi.org/10.1007/978-3-030-43722-0_3
- [77] Hiroshi Ishikawa. 2011. Transformation of General Binary MRF Minimization to the First-Order Case. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 6 (2011), 1234–1249. <https://doi.org/10.1109/TPAMI.2010.91>
- [78] A.S. Jain and S. Meeran. 1999. Deterministic job-shop scheduling: Past, present and future. *European Journal of Operational Research* 113, 2 (1999), 390–434. [https://doi.org/10.1016/S0377-2217\(98\)00113-1](https://doi.org/10.1016/S0377-2217(98)00113-1)
- [79] Adam Janiak, Wladyslaw Adam Janiak, Radoslaw Rudek, and Agnieszka Wielgus. 2009. Solution Algorithms for the Makespan Minimization Problem with the General Learning Model. *Computers & Industrial Engineering* 56, 4 (May 2009), 1301–1308. <https://doi.org/10.1016/j.cie.2008.07.019>
- [80] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. S. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose. 2011. Quantum Annealing with Manufactured Spins. *Nature* 473, 7346 (2011), 194–198. <https://doi.org/10.1038/nature10012>
- [81] J. A. Jones and M. Mosca. 1998. Implementation of a quantum algorithm on a nuclear magnetic resonance quantum computer. *The Journal of Chemical Physics* 109, 5 (Aug. 1998), 1648–1653. <https://doi.org/10.1063/1.476739>
- [82] S. Karimi and P. Ronagh. 2019. Practical integer-to-binary mapping for quantum annealers. *Quantum Inf Process* 18, 94 (2019). <https://doi.org/10.1007/s11128-019-2213-x> preprint arxiv:1706.01945.
- [83] Valentin Kasper, Daniel González-Cuadra, Apoorva Hegde, Andy Xia, Alexandre Dauphin, Felix Huber, Eberhard Tiemann, Maciej Lewenstein, Fred Jendrzejewski, and Philipp Hauke. 2021. Universal quantum computation and quantum error correction with ultracold atomic mixtures. *Quantum Science and Technology* 7, 1 (Nov 2021), 015008. <https://doi.org/10.1088/2058-9565/ac2d39>
- [84] Helmut G. Katzgraber, Firas Hamze, Zheng Zhu, Andrew J. Ochoa, and H. Munoz-Bauza. 2015. Seeking Quantum Speedup Through Spin Glasses: The Good, the Bad, and the Ugly. *Phys. Rev. X* 5 (Sep 2015), 031026. Issue 3. <https://doi.org/10.1103/PhysRevX.5.031026>
- [85] Andrew D. King, Juan Carrasquilla, Jack Raymond, Isil Ozfidan, Evgeny Andriyash, Andrew Berkley, Mauricio Reis, Trevor Lanting, Richard Harris, Fabio Altomare, Kelly Boothby, Paul I. Bunyk, Colin Enderud, Alexandre Fréchet, Emile Hoskinson, Nicolas Ladizinsky, Travis Oh, Gabriel Poulin-Lamarre, Christopher Rich, Yuki Sato, Anatoly Yu. Smirnov, Loren J. Swenson, Mark H. Volkmann, Jed Whittaker, Jason Yao, Eric Ladizinsky, Mark W. Johnson, Jeremy Hilton, and Mohammad H. Amin. 2018. Observation of topological phenomena in a programmable lattice of 1,800 qubits. *Nature* 560, 7719 (2018), 456–460. <https://doi.org/10.1038/s41586-018-0410-x>
- [86] James King, Masoud Mohseni, William Bernoudy, Alexandre Fréchet, Hossein Sadeghi, Sergei V. Isakov, Hartmut Neven, and Mohammad H. Amin. 2019. Quantum-Assisted Genetic Algorithm. arXiv:arXiv:1907.00707
- [87] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by Simulated Annealing. *Science* 220, 4598 (1983), 671–680. <https://doi.org/10.1126/science.220.4598.671>
- [88] Koki Kitai, Jiang Guo, Shenghong Ju, Shu Tanaka, Koji Tsuda, Junichiro Shiomi, and Ryo Tamura. 2020. Designing metamaterials with quantum annealing and factorization machines. *Phys. Rev. Research* 2 (Mar 2020), 013319. Issue 1. <https://doi.org/10.1103/PhysRevResearch.2.013319>
- [89] Christine Klymko, Blair D. Sullivan, and Travis S. Humble. 2012. Adiabatic Quantum Programming: Minor Embedding With Hard Faults. *arXiv:1210.8395 [quant-ph]* (Nov. 2012). arXiv:1210.8395 [quant-ph]

- [90] Mario S. Könz, Wolfgang Lechner, Helmut G. Katzgraber, and Matthias Troyer. 2021. Embedding Overhead Scaling of Optimization Problems in Quantum Annealing. *PRX Quantum* 2 (Nov 2021), 040322. Issue 4. <https://doi.org/10.1103/PRXQuantum.2.040322>
- [91] Bernhard Korte and Jens Vygen. 2012. *Combinatorial Optimization: Theory and Algorithms*. Springer.
- [92] Wen-Yang Ku and J. Christopher Beck. 2016. Mixed Integer Programming models for job shop scheduling: A computational analysis. *Computers and Operations Research* 73 (2016), 165–173. <https://doi.org/10.1016/j.cor.2016.04.006>
- [93] Vaibhaw Kumar, Gideon Bass, Casey Tomlin, and Joseph Dulny III. 2018. Quantum annealing for combinatorial clustering. *Quantum Inf Process* 17 (2018), 39. <https://doi.org/10.1007/s11128-017-1809-2>
- [94] Eil Kwon and Yorgos J. Stephanedes. 1994. Comparative evaluation of adaptive and neural-network exit demand prediction for freeway control. *Transportation Research Record* 1446 (Oct. 1994), 66–76.
- [95] T. Lanting, A. J. Przybysz, A. Yu. Smirnov, F. M. Spedalieri, M. H. Amin, A. J. Berkley, R. Harris, F. Altomare, S. Boixo, P. Bunyk, N. Dickson, C. Enderud, J. P. Hilton, E. Hoskinson, M. W. Johnson, E. Ladizinsky, N. Ladizinsky, R. Neufeld, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, S. Uchaikin, A. B. Wilson, and G. Rose. 2014. Entanglement in a Quantum Annealing Processor. *Phys. Rev. X* 4 (May 2014), 021041. Issue 2. <https://doi.org/10.1103/PhysRevX.4.021041>
- [96] Wolfgang Lechner, Philipp Hauke, and Peter Zoller. 2015. A quantum annealing architecture with all-to-all connectivity from local interactions. *Science Advances* 1, 9 (2015), e1500838. <https://doi.org/10.1126/sciadv.1500838> arXiv:<https://www.science.org/doi/pdf/10.1126/sciadv.1500838>
- [97] Martin Leib, Peter Zoller, and Wolfgang Lechner. 2016. A transmon quantum annealer: decomposing many-body Ising constraints into pair interactions. 1, 1 (dec 2016), 015008. <https://doi.org/10.1088/2058-9565/1/1/015008>
- [98] Richard Y. Li, Rosa Di Felice, Remo Rohs, and Daniel A. Lidar. 2018. Quantum Annealing versus Classical Machine Learning Applied to a Simplified Computational Biology Problem. *npj Quantum Information* 4, 1 (Feb. 2018), 1–10. <https://doi.org/10.1038/s41534-018-0060-8>
- [99] Chih-Jen Lin. 2007. Projected Gradient Methods for Nonnegative Matrix Factorization. *Neural Computation* 19, 10 (Oct. 2007), 2756–2779. <https://doi.org/10.1162/neco.2007.19.10.2756>
- [100] Andrew Lucas. [n. d.]. Hard combinatorial problems and minor embeddings on lattice graphs. arXiv:1812.01789 ([n. d.]).
- [101] Andrew Lucas. 2014. Ising Formulations of Many NP Problems. *Frontiers in Physics* 2 (2014). <https://doi.org/10.3389/fphy.2014.00005> arXiv:1302.5843
- [102] David Madigan and Jeremy York. 1995. Bayesian Graphical Models for Discrete Data. *International Statistical Review* 63 (1995), 215–232.
- [103] Harry Markowitz. 1952. Portfolio Selection. *The Journal of Finance* 7, 1 (1952), 77–91. <https://doi.org/10.1111/j.1540-6261.1952.tb01525.x>
- [104] Paul Martin and David B. Shmoys. 1996. A New Approach to Computing Optimal Schedules for the Job-Shop Scheduling Problem. In *Integer Programming and Combinatorial Optimization (Lecture Notes in Computer Science)*, William H. Cunningham, S. Thomas McCormick, and Maurice Queyranne (Eds.). Springer, Berlin, Heidelberg, 389–403. https://doi.org/10.1007/3-540-61310-2_29
- [105] Milad Marvian, Daniel A. Lidar, and Itay Hen. 2019. On the computational complexity of curing non-stoquastic Hamiltonians. *Nature Communications* 10, 1 (05 Apr 2019), 1571. <https://doi.org/10.1038/s41467-019-09501-6>
- [106] Daniel Marx. 2004. Graph colouring problems and their applications in scheduling. *Period. Polytech. Electr. Eng.* 48 (2004), 11–16.
- [107] Jarrod R McClean, Nicholas C Rubin, Kevin J Sung, Ian D Kivlichan, Xavier Bonet-Monroig, Yudong Cao, Chengyu Dai, E Schuyler Fried, Craig Gidney, Brendan Gimby, Pranav Gokhale, Thomas Häner, Tarini Hardikar, Vojtěch Havlíček, Oscar Higgott, Cupjin Huang, Josh Izaac, Zhang Jiang, Xinle Liu, Sam McArdle, Matthew Neeley, Thomas O’Brien, Bryan O’Gorman, Isil Ozfidan, Maxwell D Radin, Jhonathan Romero, Nicolas P D Sawaya, Bruno Senjean, Kanav Setia, Sukin Sim, Damian S Steiger, Mark Steudtner, Qiming Sun, Wei Sun, Daochen Wang, Fang Zhang, and Ryan Babbush. 2020. OpenFermion: The Electronic Structure Package for Quantum Computers. *Quantum Science and Technology* 5 (2020), 034014.
- [108] Barry M. McCoy and Tai Tsun Wu. 2013. *The Two-Dimensional Ising Model*. Harvard University Press.
- [109] Catherine C. McGeoch. 2014. Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice. *Synthesis Lectures on Quantum Computing* 5, 2 (July 2014), 1–93. <https://doi.org/10.2200/S00585ED1V01Y201407QMC008>
- [110] Cristian Micheletti, Philipp Hauke, and Pietro Faccioli. 2021. Polymer Physics by Quantum Computing. *Phys. Rev. Lett.* 127 (Aug 2021), 080501. Issue 8. <https://doi.org/10.1103/PhysRevLett.127.080501>
- [111] Nimish Mishra, Manik Kapil, Hemant Rakesh, Amit Anand, Nilima Mishra, Aakash Warke, Soumya Sarkar, Sanchayan Dutta, Sabhyata Gupta, Aditya Prasad Dash, Rakshit Gharat, Yagnik Chatterjee, Shuvarati Roy, Shivam Raj, Valay Kumar Jain, Shreeram Bagaria, Smit Chaudhary, Vishwanath Singh, Rituparna Maji, Priyanka Dalei, Bikash K. Behera, Sabyasachi Mukhopadhyay, and Prasanta K. Panigrahi. 2021. Quantum Machine Learning: A Review and

- Current Status. *Data Management, Analytics and Innovation. Advances in Intelligent Systems and Computing* 1175 (2021), 101–145. https://doi.org/10.1007/978-981-15-5619-7_8
- [112] Gary J. Mooney, Gregory A. L. White, Charles D. Hill, and Lloyd C. L. Hollenberg. 2021. Whole-Device Entanglement in a 65-Qubit Superconducting Quantum Computer. *Advanced Quantum Technologies* 4, 10 (2021), 2100061. <https://doi.org/10.1002/qute.202100061> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/qute.202100061>
- [113] Satoshi Morita and Hidetoshi Nishimori. 2008. Mathematical foundation of quantum annealing. *J. Math. Phys.* 49, 12 (2008), 125210. <https://doi.org/10.1063/1.2995837> arXiv:<https://doi.org/10.1063/1.2995837> preprint arXiv:0806.1859.
- [114] Alex Mott, Joshua Job, Jean-Roch Vlimant, Daniel Lidar, and Maria Spiropulu. 2017. Solving a Higgs optimization problem with quantum annealing for machine learning. *Nature* 555 (2017), 375–379. <https://doi.org/10.1038/nature24047>
- [115] Samuel Mugel, Carlos Kuchkovsky, Escolastico Sanchez, Samuel Fernandez-Lorenzo, Jorge Luis-Hita, Enrique Lizaso, and Roman Orus. 2020. Dynamic Portfolio Optimization with Real Datasets Using Quantum Processors and Quantum-Inspired Tensor Networks. *arXiv:2007.00017 [quant-ph]* (2020).
- [116] Vikram Khipple Mulligan, Hans Melo, Haley Irene Merritt, Stewart Slocum, Brian D. Weitzner, Andrew M. Watkins, P. Douglas Renfrew, Craig Pelissier, Paramjit S. Arora, and Richard Bonneau. 2020. Designing Peptides on a Quantum Computer. , 752485 pages. <https://doi.org/10.1101/752485>
- [117] Siddharth Muthukrishnan, Tameem Albash, and Daniel A. Lidar. 2016. Tunneling and Speedup in Quantum Optimization for Permutation-Symmetric Problems. *Physical Review X* 6, 3 (July 2016), 031010. <https://doi.org/10.1103/PhysRevX.6.031010> arXiv:1511.03910
- [118] Rajdeep Kumar Nath, Himanshu Thapliyal, and Travis S. Humble. 2021. A Review of Machine Learning Classification Using Quantum Annealing for Real-World Applications. *arXiv:2106.02964 [quant-ph]* (June 2021). arXiv:2106.02964 [quant-ph]
- [119] Florian Neukart, Gabriele Compostella, Christian Seidel, David von Dollen, Sheir Yarkoni, and Bob Parney. 2017. Traffic Flow Optimization Using a Quantum Annealer. *Frontiers in ICT* 4 (2017), 29. <https://doi.org/10.3389/fict.2017.00029>
- [120] Florian Neukart, David Von Dollen, and Christian Seidel. 2018. Quantum-Assisted Cluster Analysis on a Quantum Annealing Device. *Frontiers in Physics* 6 (2018), 55. <https://doi.org/10.3389/fphy.2018.00055>
- [121] Florian Neukart, David Von Dollen, Christian Seidel, and Gabriele Compostella. 2018. Quantum-Enhanced Reinforcement Learning for Finite-Episode Games with Discrete State Spaces. *Frontiers in Physics* 5 (2018), 71. <https://doi.org/10.3389/fphy.2017.00071>
- [122] Hartmut Neven, Vasil Denchev, Marshall Drew-Brook, Jiayong Zhang, William Macready, and Geordie Rose. 2009. NIPS 2009 Demonstration: Binary Classification using Hardware Implementation of Quantum Annealing. (2009), 1–17.
- [123] Hartmut Neven, Vasil S. Denchev, Geordie Rose, and William G. Macready. 2009. Training a Large Scale Classifier with the Quantum Adiabatic Algorithm. *arXiv:0912.0779* (2009).
- [124] Hidetoshi Nishimori and Kabuki Takada. 2017. Exponential Enhancement of the Efficiency of Quantum Annealing by Non-Stoquastic Hamiltonians. *Frontiers in ICT* 4 (2017), 2. <https://doi.org/10.3389/fict.2017.00002>
- [125] Masayuki Ohzeki. 2020. Breaking limitation of quantum annealer in solving optimization problems under constraints. *Scientific Reports volume* 10 (2020), 3126. <https://doi.org/10.1038/s41598-020-60022-5>
- [126] Masayuki Ohzeki, Akira Miki, Masamichi J. Miyama, and Masayoshi Terabe. 2019. Control of Automated Guided Vehicles Without Collision by Quantum Annealer and Digital Devices. *Frontiers in Computer Science* 1 (2019). <https://doi.org/10.3389/fcomp.2019.00009>
- [127] Shuntaro Okada and Shinichiro Ohzeki, Masayuki open-and Taguchi. 2019. Improving solutions by embedding larger subproblems in a D-Wave quantum annealer. *Scientific Reports* 9, 1 (14 Feb 2019), 2098. <https://doi.org/10.1038/s41598-018-38388-4>
- [128] Daniel O'Malley, Velimir V. Vesselinov, Boian S. Alexandrov, and Ludmil B. Alexandrov. 2018. Nonnegative/Binary matrix factorization with a D-Wave quantum annealer. *PLOS ONE* 13, 12 (Dec. 2018), e0206653. <https://doi.org/10.1371/journal.pone.0206653>
- [129] Román Orús, Samuel Mugel, and Enrique Lizaso. 2019. Quantum computing for finance: Overview and prospects. *Reviews in Physics* 4 (2019), 100028. <https://doi.org/10.1016/j.revip.2019.100028>
- [130] J. Ossorio-Castillo and F. Pena-Brage. 2021. Optimization of a refinery scheduling process with column generation and a quantum annealer. *Optimization and Engineering* (2021). <https://doi.org/10.1007/s11081-021-09662-8>
- [131] Daniele Ottaviani and Alfonso Amendola. 2018. Low Rank Non-Negative Matrix Factorization with D-Wave 2000Q. arXiv:arXiv:1808.08721
- [132] Daniele Ottaviani and Alfonso Amendola. 2018. Low Rank Non-Negative Matrix Factorization with D-Wave 2000Q. arXiv:arXiv:1808.08721
- [133] I. Ozfidan, C. Deng, A.Y. Smirnov, T. Lanting, R. Harris, L. Swenson, J. Whittaker, F. Altomare, M. Babcock, C. Baron, A.J. Berkley, K. Boothby, H. Christiani, P. Bunyk, C. Enderud, B. Evert, M. Hager, A. Hajda, J. Hilton, S. Huang, E.

- Hoskinson, M.W. Johnson, K. Jooya, E. Ladizinsky, N. Ladizinsky, R. Li, A. MacDonald, D. Marsden, G. Marsden, T. Medina, R. Molavi, R. Neufeld, M. Nissen, M. Norouzpour, T. Oh, I. Pavlov, I. Perminov, G. Poulin-Lamarre, M. Reis, T. Prescott, C. Rich, Y. Sato, G. Sterling, N. Tsai, M. Volkmann, W. Wilkinson, J. Yao, and M.H. Amin. 2020. Demonstration of a Nonstoquastic Hamiltonian in Coupled Superconducting Flux Qubits. *Phys. Rev. Applied* 13 (Mar 2020), 034037. Issue 3. <https://doi.org/10.1103/PhysRevApplied.13.034037>
- [134] Alejandro Perdomo-Ortiz, Neil Dickson, Marshall Drew-Brook, Geordie Rose, and Alán Aspuru-Guzik. 2012. Finding Low-Energy Conformations of Lattice Protein Models by Quantum Annealing. *Scientific Reports* 2, 1 (Aug. 2012), 571. <https://doi.org/10.1038/srep00571>
- [135] Frank Phillipson and Harshil Singh Bhatia. 2020. Portfolio Optimisation Using the D-Wave Quantum Annealer. *arXiv:2012.01121 [quant-ph, q-fin]* (Nov. 2020). arXiv:2012.01121 [quant-ph, q-fin]
- [136] I. Pogorelov, T. Feldker, Ch. D. Marciniak, L. Postler, G. Jacob, O. Krieglsteiner, V. Podlesnic, M. Meth, V. Negnevitsky, M. Stadler, B. Höfer, C. Wächter, K. Lakhmanskiy, R. Blatt, P. Schindler, and T. Monz. 2021. Compact Ion-Trap Quantum Computing Demonstrator. *PRX Quantum* 2 (Jun 2021), 020343. Issue 2. <https://doi.org/10.1103/PRXQuantum.2.020343>
- [137] Matias Péres, Germán Ruiz, Sergio Nesmachnow, and Ana C. Olivera. 2018. Multiobjective evolutionary optimization of traffic flow and pollution in Montevideo, Uruguay. *Applied Soft Computing* 70 (2018), 472–485. <https://doi.org/10.1016/j.asoc.2018.05.044>
- [138] Kristen L. Pudenz, Tameem Albash, and Daniel A. Lidar. 2014. Error-corrected quantum annealing with hundreds of qubits. *Nature Communications* 5, 1 (06 Feb 2014), 3243. <https://doi.org/10.1038/ncomms4243>
- [139] Jack Raymond, Sheir Yarkoni, and Evgeny Andriyash. 2016. Global Warming: Temperature Estimation in Annealers. *Frontiers in ICT* 3 (2016), 23. <https://doi.org/10.3389/fict.2016.00023>
- [140] Mohsen Razavy. 2003. *Quantum Theory Of Tunneling*. World Scientific Publishing.
- [141] Matthew Reagor, Christopher B. Osborn, Nikolas Tezak, Alexa Staley, Guenevere Prawiroatmodjo, Michael Scheer, Nasser Alidoust, Eyob A. Sete, Nicolas Didier, Marcus P. da Silva, Ezer Acala, Joel Angeles, Andrew Bestwick, Maxwell Block, Benjamin Bloom, Adam Bradley, Catvu Bui, Shane Caldwell, Lauren Capelluto, Rick Chilcott, Jeff Cordova, Genya Crossman, Michael Curtis, Saniya Deshpande, Tristan El Bouayadi, Daniel Girshovich, Sabrina Hong, Alex Hudson, Peter Karalekas, Kat Kuang, Michael Lenihan, Riccardo Manenti, Thomas Manning, Jays Marshall, Yuvraj Mohan, William O'Brien, Johannes Otterbach, Alexander Papageorge, Jean-Philip Paquette, Michael Pelstring, Anthony Polloreno, Vijay Rawat, Colm A. Ryan, Russ Renzas, Nick Rubin, Damon Russel, Michael Rust, Diego Scarabelli, Michael Selvanayagam, Rodney Sinclair, Robert Smith, Mark Suska, Ting-Wai To, Mehrnoosh Vahidpour, Nagesh Vodrahalli, Tyler Whyland, Kamal Yadav, William Zeng, and Chad T. Rigetti. 2018. Demonstration of universal parametric entangling gates on a multi-qubit lattice. *Science Advances* 4, 2 (2018), eaao3603. <https://doi.org/10.1126/sciadv.aao3603> arXiv:<https://www.science.org/doi/pdf/10.1126/sciadv.aao3603>
- [142] K. Rezaee, B. Abdulhai, and Hossam Abdelgawad. 2012. Application of Reinforcement Learning with Continuous State Space to Ramp Metering in Real-World Conditions. *2012 15th International IEEE Conference on Intelligent Transportation Systems* (2012). <https://doi.org/10.1109/ITSC.2012.6338837>
- [143] Eleanor G. Rieffel, Davide Venturelli, Bryan O'Gorman, Minh B. Do, Elicia M. Prystay, and Vadim N. Smelyanskiy. 2015. A Case Study in Programming a Quantum Annealer for Hard Operational Planning Problems. *Quantum Information Processing* 14, 1 (Jan. 2015), 1–36. <https://doi.org/10.1007/s11128-014-0892-x>
- [144] N. Robertson and P. D. Seymour. 1995. Graph Minors. XIII. The Disjoint Paths Problem. *Journal of Combinatorial Theory, Series B* 63, 1 (Jan. 1995), 65–110. <https://doi.org/10.1006/jctb.1995.1006>
- [145] Troels F. Rønnow, Zhihui Wang, Joshua Job, Sergio Boixo, Sergei V. Isakov, David Wecker, John M. Martinis, Daniel A. Lidar, and Matthias Troyer. 2014. Defining and detecting quantum speedup. *Science* 345, 6195 (2014), 420–424. <https://doi.org/10.1126/science.1252319> arXiv:<https://science.sciencemag.org/content/345/6195/420.full.pdf>
- [146] G. Rosenberg, P. Haghnegahdar, P. Goddard, P. Carr, K. Wu, and M. L. de Prado. 2016. Solving the Optimal Trading Trajectory Problem Using a Quantum Annealer. *IEEE Journal of Selected Topics in Signal Processing* 10, 6 (2016), 1053–1060. <https://doi.org/10.1109/JSTSP.2016.2574703>
- [147] Gili Rosenberg, Poya Haghnegahdar, Phil Goddard, Peter Carr, Kesheng Wu, and Marcos López De Prado. 2016. Solving the Optimal Trading Trajectory Problem Using a Quantum Annealer. *IEEE Journal on Selected Topics in Signal Processing* 10, 6 (Sept. 2016), 1053–1060. <https://doi.org/10.1109/JSTSP.2016.2574703>
- [148] Pascal Scholl, Michael Schuler, Hannah J. Williams, Alexander A. Eberharter, Daniel Barredo, Kai-Niklas Schymik, Vincent Lienhard, Louis-Paul Henry, Thomas C. Lang, Thierry Lahaye, Andreas M. Läuchli, and Antoine Browaeys. 2021. Quantum simulation of 2D antiferromagnets with hundreds of Rydberg atoms. *Nature* 595, 7866 (01 Jul 2021), 233–238. <https://doi.org/10.1038/s41586-021-03585-1>
- [149] David Sherrington and Scott Kirkpatrick. 1975. Solvable Model of a Spin-Glass. *Physical Review Letters* 35, 26 (Dec. 1975), 1792–1796. <https://doi.org/10.1103/PhysRevLett.35.1792>
- [150] Peter W. Shor. 1997. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* 26, 5 (Oct. 1997), 1484–1509. <https://doi.org/10.1137/s0097539795293172>

- [151] R. B. Stinchcombe. 1973. Ising Model in a Transverse Field. I. Basic Theory. *Journal of Physics C: Solid State Physics* 6, 15 (Aug. 1973), 2459–2483. <https://doi.org/10.1088/0022-3719/6/15/009>
- [152] Tobias Stollenwerk, Elisabeth Lobe, and Martin Jung. 2018. Flight Gate Assignment with a Quantum Annealer. (2018). arXiv:arXiv:1811.09465
- [153] Tobias Stollenwerk, Bryan O’Gorman, Davide Venturelli, Salvatore Mandra, Olga Rodionova, Hokkwan Ng, Banavar Sridhar, Eleanor Gilbert Rieffel, and Rupak Biswas. 2020. Quantum Annealing Applied to De-Conflicting Optimal Trajectories for Air Traffic Management. *IEEE Transactions on Intelligent Transportation Systems* 21, 1 (Jan. 2020), 285–297. <https://doi.org/10.1109/tits.2019.2891235>
- [154] Michael Streif, Florian Neukart, and Martin Leib. 2019. Solving Quantum Chemistry Problems with a D-Wave Quantum Annealer. *arXiv:1811.05256 [quant-ph]* (2019). arXiv:1811.05256 [quant-ph]
- [155] Yuki Susa, Yu Yamashiro, Masayuki Yamamoto, and Hidetoshi Nishimori. 2018. Exponential Speedup of Quantum Annealing by Inhomogeneous Driving of the Transverse Field. *Journal of the Physical Society of Japan* 87, 2 (Feb. 2018), 023002. <https://doi.org/10.7566/JPSJ.87.023002>
- [156] Robert H. Swendsen and Jian-Sheng Wang. 1986. Replica Monte Carlo Simulation of Spin-Glasses. *Physical Review Letters* 57, 21 (Nov. 1986), 2607–2609. <https://doi.org/10.1103/PhysRevLett.57.2607>
- [157] Kensuke Tamura, Tatsuhiko Shirai, Hosho Katsura, Shu Tanaka, and Nozomu Togawa. 2021. Performance Comparison of Typical Binary-Integer Encodings in an Ising Machine. *IEEE Access* 9 (2021), 81032–81039. <https://doi.org/10.1109/ACCESS.2021.3081685>
- [158] Shu Tanaka, B. K. Chakrabarti, and Ryo Tamura. 2017. *Quantum Spin Glasses, Annealing and Computation*. Cambridge University Press.
- [159] Olawale Titiloye and Alan Crispin. 2011. Graph Coloring with a Distributed Hybrid Quantum Annealing Algorithm. In *Agent and Multi-Agent Systems: Technologies and Applications*, James O’Shea, Ngoc Thanh Nguyen, Keeley Crockett, Robert J. Howlett, and Lakhmi C. Jain (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 553–562.
- [160] Olawale Titiloye and Alan Crispin. 2011. Quantum Annealing of the Graph Coloring Problem. *Discrete Optimization* 8, 2 (May 2011), 376–384. <https://doi.org/10.1016/j.disopt.2010.12.001>
- [161] Paolo Toth and Daniele Vigo (Eds.). 2002. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, Philadelphia.
- [162] Elisabetta Valiante, Maritza Hernandez, Amin Barzegar, and Helmut G. Katzgraber. 2021. Computational overhead of locality reduction in binary optimization problems. *Computer Physics Communications* 269 (2021), 108102. <https://doi.org/10.1016/j.cpc.2021.108102> preprint arXiv:2012.09681.
- [163] Dyon van Vreumingen, Florian Neukart, David Von Dollen, Carsten Othmer, Michael Hartmann, Arne-Christian Voigt, and Thomas Bäck. 2019. Quantum-assisted finite-element design optimization. *arXiv:1908.03947* (2019).
- [164] Davide Venturelli and Alexei Kondratyev. 2019. Reverse quantum annealing approach to portfolio optimization problems. *Quantum Machine Intelligence* 1, 1-2 (2019), 17–30. <https://doi.org/10.1007/s42484-019-00001-w>
- [165] Davide Venturelli, Salvatore Mandra, Sergey Knysch, Bryan O’Gorman, Rupak Biswas, and Vadim Smelyanskiy. 2015. Quantum Optimization of Fully Connected Spin Glasses. *Phys. Rev. X* 5 (Sep 2015), 031040. Issue 3. <https://doi.org/10.1103/PhysRevX.5.031040>
- [166] Davide Venturelli, Dominic J. J. Marchand, and Galo Rojo. 2016. Quantum Annealing Implementation of Job-Shop Scheduling. *arXiv:1506.08479 [quant-ph]* (2016). arXiv:1506.08479 [quant-ph]
- [167] Z. Vernon, N. Quesada, M. Liscidini, B. Morrison, M. Menotti, K. Tan, and J.E. Sipe. 2019. Scalable Squeezed-Light Source for Continuous-Variable Quantum Sampling. *Phys. Rev. Applied* 12 (Dec 2019), 064024. Issue 6. <https://doi.org/10.1103/PhysRevApplied.12.064024>
- [168] Walter Vinci and Daniel A. Lidar. 2017. Non-stoquastic Hamiltonians in quantum annealing via geometric phases. *npj Quantum Information* 3, 1 (2017), 38. <https://doi.org/10.1038/s41534-017-0037-z> arXiv:1701.07494
- [169] Erwin Walraven, Matthijs T. J. Spaan, and Bram Bakker. 2016. Traffic Flow Optimization: A Reinforcement Learning Approach. *Engineering Applications of Artificial Intelligence* 52 (June 2016), 203–212. <https://doi.org/10.1016/j.engappai.2016.01.001>
- [170] Yang Wang, Xianli Zhang, Theodore A. Corcovilos, Aishwarya Kumar, and David S. Weiss. 2015. Coherent Addressing of Individual Neutral Atoms in a 3D Optical Lattice. *Phys. Rev. Lett.* 115 (Jul 2015), 043003. Issue 4. <https://doi.org/10.1103/PhysRevLett.115.043003>
- [171] W. Wei and Y. Zhang. 2002. FL-FN Based Traffic Signal Control. *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE’02. Proceedings (Cat. No.02CH37291)* (2002). <https://doi.org/10.1109/FUZZ.2002.1005004>
- [172] Dennis Willsch, Madita Willsch, Hans De Raedt, and Kristel Michielsen. 2020. Support Vector Machines on the D-Wave Quantum Annealer. *Computer Physics Communications* 248 (March 2020), 107006. <https://doi.org/10.1016/j.cpc.2019.107006> arXiv:1906.06283

- [173] Blake A. Wilson, Zhaxylyk A. Kudyshev, Alexander V. Kildishev, Sabre Kais, Vladimir M. Shalaev, and Alexandra Boltasseva. 2021. Machine Learning Framework for Quantum Sampling of Highly-Constrained, Continuous Optimization Problems. *arXiv:2105.02396* (2021).
- [174] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pinti, M. Chmielewski, C. Collins, K. M. Hudek, J. Mizrahi, J. D. Wong-Campos, S. Allen, J. Apisdorf, P. Solomon, M. Williams, A. M. Duocre, A. Blinov, S. M. Kreikemeier, V. Chaplin, M. Keesan, C. Monroe, and J. Kim. 2019. Benchmarking an 11-qubit quantum computer. *Nature Communications* 10 (2019), 5464.
- [175] Rongxin Xia, Teng Bian, and Sabre Kais. 2018. Electronic Structure Calculations and the Ising Hamiltonian. *The Journal of Physical Chemistry B* 122, 13 (2018), 3384–3395. <https://doi.org/10.1021/acs.jpbc.7b10371>
- [176] Sheir Yarkoni, Alex Alekseyenko, Michael Streif, David Von Dollen, Florian Neukart, and Thomas Bäck. 2021. Multi-car paint shop optimization with quantum annealing. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 35–41. <https://doi.org/10.1109/QCE52317.2021.00019>
- [177] Sheir Yarkoni, Andreas Huck, Hanno Schülldorf, Benjamin Speitkamp, Marc Shakory Tabrizi, Martin Leib, Thomas Bäck, and Florian Neukart. 2021. Solving the Shipment Rerouting Problem with Quantum Optimization Techniques. In *Computational Logistics*, Martijn Mes, Eduardo Lalla-Ruiz, and Stefan Voß (Eds.). Springer International Publishing, Cham, 502–517.
- [178] Sheir Yarkoni, Florian Neukart, Eliane Moreno Gomez Tagle, Nicole Magiera, Bharat Mehta, Kunal Hire, Swapnil Narkhede, and Martin Hofmann. 2020. Quantum Shuttle: Traffic Navigation with Quantum Computing. In *Proceedings of the 1st ACM SIGSOFT International Workshop on Architectures and Paradigms for Engineering Quantum Software (Virtual, USA) (APEQS 2020)*. Association for Computing Machinery, New York, NY, USA, 22–30. <https://doi.org/10.1145/3412451.3428500>
- [179] Sheir Yarkoni, Hao Wang, Aske Plaat, and Thomas Bäck. 2019. Boosting Quantum Annealing Performance Using Evolution Strategies for Annealing Offsets Tuning. In *Quantum Technology and Optimization Problems*, Sebastian Feld and Claudia Linnhoff-Popien (Eds.). Springer International Publishing, Cham, 157–168.
- [180] Sizhuo Yu and Tahar Nabil. 2021. Applying the Hubbard-Stratonovich Transformation to Solve Scheduling Problems Under Inequality Constraints With Quantum Annealing. *Frontiers in Physics* 9 (2021), 520. <https://doi.org/10.3389/fphy.2021.730685>
- [181] Alexandre M. Zagoskin, Evgeni Ilichev, Miroslav Grajcar, Joseph J. Betouras, and Franco Nori. 2014. How to Test the "Quantumness" of a Quantum Computer? *arXiv:1401.2870 [quant-ph]* (Jan. 2014). <https://doi.org/10.3389/fphy.2014.00033> arXiv:1401.2870 [quant-ph]
- [182] Arman Zaribafiyani, Dominic J. J. Marchand, and Seyed Saeed Changiz Rezaei. 2017. Systematic and deterministic graph minor embedding for Cartesian products of graphs. *Quantum Information Processing* 16, 5 (10 Apr 2017), 136. <https://doi.org/10.1007/s11128-017-1569-z>