frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 05 June 2024
DOI 10.3389/fcomp.2024.1286057

Check for updates

OPEN ACCESS

EDITED BY
David Esteban Bernal Neira,
Purdue University, United States

REVIEWED BY
Luis Zuluaga,
Lehigh University, United States
Marcos César de Oliveira,
State University of Campinas, Brazil

*CORRESPONDENCE
Stéphane Louise
✉ Stephane.LOUISE@cea.fr
Kristel Michielsen
✉ k.michielsen@fz-juelich.de

†These authors share first authorship

RECEIVED 30 August 2023
ACCEPTED 14 May 2024
PUBLISHED 05 June 2024

CITATION
Vert D, Willsch M, Yenilen B, Sirdey R, Louise S and Michielsen K (2024) Benchmarking quantum annealing with maximum cardinality matching problems.
Front. Comput. Sci. 6:1286057.
doi: 10.3389/fcomp.2024.1286057

COPYRIGHT
© 2024 Vert, Willsch, Yenilen, Sirdey, Louise and Michielsen. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Benchmarking quantum annealing with maximum cardinality matching problems

Daniel Vert[1,2†], Madita Willsch[3,4†], Berat Yenilen[3,5], Renaud Sirdey[1,2], Stéphane Louise[1,2*] and Kristel Michielsen[3,4,5*]

[1]Université Paris-Saclay Commission for Atomic Energy (CEA), Laboratory of Integration of Systems and Technologies (LIST), Palaiseau, France, [2]AIDAS, Palaiseau, France, [3]Jülich Supercomputing Centre, Forschungszentrum Jülich, Jülich, Germany, [4]University in German Rheinisch-Westfälische Technische Hochschule Aachen (RWTH), Jülich, Germany, [5]RWTH Aachen University, Aachen, Germany

We benchmark Quantum Annealing (QA) vs. Simulated Annealing (SA) with a focus on the impact of the embedding of problems onto the different topologies of the D-Wave quantum annealers. The series of problems we study are especially designed instances of the maximum cardinality matching problem that are easy to solve classically but difficult for SA and, as found experimentally, not easy for QA either. In addition to using several D-Wave processors, we simulate the QA process by numerically solving the time-dependent Schrödinger equation. We find that the embedded problems can be significantly more difficult than the unembedded problems, and some parameters, such as the chain strength, can be very impactful for finding the optimal solution. Thus, finding a good embedding and optimal parameter values can improve the results considerably. Interestingly, we find that although SA succeeds for the unembedded problems, the SA results obtained for the embedded version scale quite poorly in comparison with what we can achieve on the D-Wave quantum annealers.

KEYWORDS

quantum annealing, simulated annealing, benchmarking, maximum cardinality matching problem, minor embedding

1 Introduction

In theory, quantum computing has the potential to yield significant, potentially even exponential, speed-up compared with the best known algorithms of traditional computing. Whether quantum computing can meet these high expectations is not yet clear. The current technological state of the art only allows for rather modest results: Even though some computing advantage has been shown on some corner cases (Arute et al., 2019; King et al., 2021), no computing advantage that is relevant for practical applications has been shown so far, and although successful application of quantum error correction has been reported (Ryan-Anderson et al., 2022; Takeda et al., 2022; Acharya et al., 2023), the threshold for fault-tolerant quantum computing has not yet been reached.

However, especially when focusing on the field of optimization, there exist quantum processors that do not rely on quantum gates but on Quantum Annealing (QA)—the natural time evolution of a "programmable" quantum system—to find a solution of an optimization problem. The Canadian company D-Wave Systems Inc. builds and commercializes such quantum processors comprising over 5,000 qubits. As these types of quantum computers act as optimizing black boxes, an important aspect from a user perspective is to characterize the quality of the outcome. Furthermore, such benchmarking

activities are of high interest for commercial and scientific prospects. There are several aspects of benchmarking:

(i) One aspect is the investigation of whether or not quantum effects are at play in the annealing process of the D-Wave processors. Studies with this aim were conducted by comparing D-Wave data with (simulated) QA and classical models often including, among others, Simulated Annealing (SA). Such studies either showed that previously considered classical models cannot reproduce the D-Wave results, for instance for freeze-out time vs. temperature (Johnson et al., 2011), for success probability distributions over many random spin glass instances (Boixo et al., 2014), and for the non-uniform probability of degenerate ground states of specially crafted problem instances where the probability of one ground state is suppressed in the quantum case and enhanced for SA (Boixo et al., 2013; Albash et al., 2015) or introduced another classical model such as noisy Boltzmann distributions (Chancellor et al., 2022), classical spin dynamics (Smolin and Smith, 2013), and spin-vector Monte Carlo (SVMC, also called SSSV model—from Shin Smith Smolin Vazirani) (Shin et al., 2014), which shows agreement with the studied D-Wave data.

(ii) A second aspect is the comparison between different generations of processors to judge possible improvements. Previous studies compared D-Wave 2000Q and Advantage system (Calaza et al., 2021; McLeod and Sasdelli, 2022; Willsch et al., 2022a), D-Wave 2000Q, Advantage system and Advantage2 prototype (Pelofske, 2023), and D-Wave Two, D-Wave 2X, D-Wave 2000Q, and Advantage system (Pokharel et al., 2021).

(iii) Another aspect is the search for quantum speedup (Rønnow et al., 2014) and investigations of the performance of quantum processors in comparison to classical algorithms. Studies were performed for academic instances such as random spin glasses (Rønnow et al., 2014), specially crafted problems with or without planted solutions (Hen et al., 2015; King et al., 2015; Albash and Lidar, 2018; Vert et al., 2020; McLeod and Sasdelli, 2022), a variety of problems with different level of difficulty (Jünger et al., 2021; McGeoch and Farre, 2023) and problems with industrial application such as the multi-car paint shop problem (Yarkoni et al., 2021), job shop scheduling problem (Carugno et al., 2022), and Earth-observation satellite mission planning problem (Stollenwerk et al., 2021). Studies benchmarking QA against classical algorithms comprise annealing-like algorithms such as SA (Rønnow et al., 2014; Hen et al., 2015; King et al., 2015; Albash and Lidar, 2018; Vert et al., 2020; Yarkoni et al., 2021; Carugno et al., 2022; McLeod and Sasdelli, 2022; Ceselli and Premoli, 2023; McGeoch and Farre, 2023), parallel tempering (McGeoch and Farre, 2023), simulated QA and SVMC (Hen et al., 2015; Albash and Lidar, 2018), and heuristic algorithms such as Tabu search (McGeoch and Wang, 2013; Yarkoni et al., 2021; Carugno et al., 2022), Hamze-de Freitas-Selby algorithm (Hen et al., 2015; King et al., 2015), or greedy algorithms (Yarkoni et al., 2021; Carugno et al., 2022; McGeoch and Farre, 2023), as well as exact solvers (McGeoch and Wang, 2013; Jünger et al., 2021; Stollenwerk et al., 2021; Ceselli and Premoli, 2023).

Benchmarking QA by comparing it to SA is a common approach since these two meta-heuristics share some similarities, SA being inspired by statistical physics and QA relying on quantum processes to achieve an optimization. In essence, SA (slowly) converges toward a Boltzmann-like distribution with a high probability of sampling low-cost solutions, whereas QA attempts to converge toward a quantum state which contains with high probability the low-energy states of an Hamiltonian. SA works well for a variety of problems, including NP-hard problems, to obtain a solution of reasonably good quality without spending too many computational resources.

In this study, we extend previous study on earlier generations of the D-Wave quantum processors in which QA was compared with SA on a specially crafted series of problems that are known to be asymptotically difficult for SA: the G_n series of the Maximum Cardinality Matching (MCM) problem. In Vert et al. (2020) and Vert et al. (2021), this problem was studied on the D-Wave 2X processor and showed how the sparse connectivity of this particular machine adds to the difficulty of solving this series of problem.

In McLeod and Sasdelli (2022), the same problem was studied, and it was pointed out that the G_n series may be exponentially difficult for the QA meta-heuristic as well, as preliminary results indicated that the spectral gap could decrease quickly in this series of problem. Indeed, the spectral gap is an important element in deciding if a given problem requires a long annealing time to reach the solution with a high probability.

In this study, we extend the study of the G_n series further:

- We added the results obtained from simulating the ideal QA process on conventional computers to assess how problem difficulty is affected by the embedding procedure even under ideal conditions.
- We provide some order of magnitude estimation to check that intrinsic precision limitations of the D-Wave processor are not the main cause of error up to the largest problem which was possible to map onto these processors.
- We compare the results including improved embeddings between several generations of the D-Wave quantum processing units (QPUs), which have different levels of connectivity, to investigate improvements between the different generations and assess the performance gain due to the higher connectivity.
- We show that the embedding is a sensitive parameter for the quality of the results obtained from the QPUs by studying the performance of SA using the same embedded instances on the D-Wave processors.

Using several approaches (ideal QA, QA on different generations of processors, and SA), we present a extensive study to demonstrate that the minor embedding required to map a problem instance onto the quantum processors can increase the difficulty of the problem significantly.

The outline of the study is as follows: First, in Section 2, we introduce the theoretical background and the applied methods relevant for the study. Second, we show and discuss the results of several experiments in Section 3 before concluding our study in Section 4.

2 Theoretical background and methods

In this section, we introduce basic concepts to understand the important ideas of this study and outline why a particular series of problems aimed at theoretically probing the worst-case complexity of SA which is relevant to test some aspects of QA. Therefore, in this section, we briefly introduce the key concepts of SA, the MCM problem, and QA with a focus on how it is implemented and what are the concrete limitations of its implementation on D-Wave quantum processors.

2.1 Simulated annealing

SA is a probabilistic meta-heuristic algorithm that is commonly used for solving optimization problems (Kirkpatrick et al., 1983). The name "annealing" comes from the annealing process in metallurgy, and the idea is strongly inspired by statistical physics.

In SA, an initial solution is randomly generated, and the algorithm gradually explores the search space by making small changes to the solution. The algorithm can even accept these changes in detrimental cases with a probability that decreases over time, i.e., with a parameter corresponding to temperature, similar to a cooling process. This allows the algorithm to escape local optima and search for better solutions.

SA has been used to solve a wide range of optimization problems, including the Traveling Salesman Problem (TSP), scheduling problems, and other NP-hard problems. It is particularly useful for problems where the search space is large and the objective function is noisy or difficult to evaluate.

One of the key advantages of SA is that it can find a good solution even if the objective function has multiple local optima. However, it can be slow to converge, and it may require a large number of iterations to find a good solution. Additionally, the performance of the algorithm can be sensitive to its parameters, such as the cooling schedule and the acceptance probability function.

As SA has been around for so long, there is no need to further introduce the general method but rather to specify the key free parameter choices for reproducibility. In our case, we have used a standard cooling schedule of the form $T_{k+1} = 0.95T_k$ starting at $T_0 = |c_0|$ (c_0 is the high cost of the initial random solution) and stopping when $T < 10^{-3}$. The key parameter of our implementation, however, is the number of iterations of the Metropolis algorithm running for each k at constant temperature which we set to n and n^2 (where n denotes the number of variables in the QUBO). For n iterations per plateau of temperature, the algorithm is very fast but the Metropolis algorithm has less iterations to reach its stationary distribution, and hence, the algorithm is expected to provide lower quality results. On the other end of the spectrum, n^2 iterations per plateau means that one can expect high-quality results, but the computation time is then much more important.

2.2 Quantum annealing principles and D-wave processors

Like SA, QA is a meta-heuristic, but instead of being inspired by statistical physics, QA lays its base on quantum physics. While first intended as a variation of SA where thermal fluctuations are replaced by quantum fluctuations (Finnila et al., 1994; Kadowaki and Nishimori, 1998), in case of a closed system, QA can also be understood by the adiabatic theorem of quantum mechanics (Farhi et al., 2000; Albash and Lidar, 2018). More accurately, a specific utilization of the adiabatic theorem applies: If a quantum system is prepared in an initial state that is the ground state of a Hamiltonian, and if the Hamiltonian is slowly and continuously changed over time, the system will remain in the instantaneous ground state of the Hamiltonian throughout the evolution. The time of the total process from the initial (or "driver") Hamiltonian to the final one is known as the *annealing time* t_A.

Let $\mathcal{H}_0$ denote the initial (driver) Hamiltonian and $\mathcal{H}_f$ the final one. For $A(s)$ and $B(s)$, respectively decreasing and increasing functions of $s = t/t_A$ with support in the interval $[0, 1]$ so that $A(0) \gg B(0)$ and $A(1) \ll B(1)$,

$$\mathcal{H}(t) = A(t/t_A)\mathcal{H}_0 + B(t/t_A)\mathcal{H}_f$$

interpolates between $\mathcal{H}_0$ and $\mathcal{H}_f$. A usual illustration is given by using a linear ramp:

$$\mathcal{H}(t) = \left(1 - \frac{t}{t_A}\right)\mathcal{H}_0 + \frac{t}{t_A}\mathcal{H}_f$$

In QA, the system is initialized in the ground state of a simple Hamiltonian $\mathcal{H}_0$, and the Hamiltonian is slowly changed to $\mathcal{H}_f$ that encodes the cost function of interest. In theory, the adiabatic theorem ensures that the system remains in the ground state throughout this process, allowing the cost function to be effectively minimized.

In the case of D-Wave quantum annealers, $\mathcal{H}_0$ and $\mathcal{H}_f$ are given as follows:

$$\mathcal{H}_0 = -\sum_{i=1}^{N} \sigma_i^x, \tag{1}$$

$$\mathcal{H}_f = \sum_{i=1}^{N} h_i \sigma_i^z + \sum_{(i,j)} J_{i,j} \sigma_i^z \sigma_j^z, \tag{2}$$

where σ_i^x and σ_i^z are the Pauli x- and z operators, N is the number of qubits, and the summation index (i, j) indicates that the sum is over pairs of qubits that are connected on the hardware. The initial quantum state $|\varphi_0\rangle = \bigotimes_i |+\rangle$ is the uniform superposition of all basis states in the z-basis or equivalently the state with all spins aligned with the x-axis. The final solution is a vector of spins $\mathbf{s}$ with $s_i \in \{-1, 1\}$ aligned with the z-axis.

The Ising Hamiltonian Equation (2) is formally and bijectively a Quadratic Unconstrained Binary Optimization (QUBO) problem formulation. Therefore, finding the energy of the fundamental state of $\mathcal{H}_f$ is equivalent to solving a QUBO problem, which is in the general case an NP-hard problem. Since NP-complete problems are polynomially reducible to one another (Garey and Johnson, 1979), many NP-hard problems such as the TSP or many logistics

and other important optimization problems can be expressed as a QUBO formulation. However, this is easier said than done as finding quadratic reformulations allowing faster resolution for some NP-hard problems is still an active domain of research in the Operations Research community (see Anthony et al., 2017 for a recent reference). A few important examples are further presented in Lucas (2014).

A QUBO problem is defined as follows:

$$\min_{\mathbf{x}\in\{0,1\}^N} (\mathcal{C}(\mathbf{x}) = \mathbf{x}^t Q \mathbf{x}),$$

where $\mathcal{C}(\mathbf{x})$ is the cost function to be minimized, $\mathbf{x}$ is a binary vector with $x_i \in \{0, 1\}$, and Q is a matrix in $\mathcal{M}_{N\times N}(\mathbb{R})$. It is worth noting that Q is often expressed as an upper triangular matrix, although it is not mandatory. Note that constrained quadratic binary problems can also be transformed to QUBO if constraints are incorporated into the cost function as soft constraints (Lucas, 2014).

2.2.1 The embedding problem on D-Wave machines

Coupling coefficients of the Ising problem can only be set to values different from zero if the physical connection between the qubits exists. The connectivity between qubits on D-Wave quantum processors is given by a sparse graph which is different for the three generations of devices that we benchmark in our present study. The architecture of the connectivity graph in the DW_2000Q Quantum Processing Unit (QPU) is called "Chimera." The average number of connections per qubit is equal to 6, and the number of qubits of DW_2000Q processors is slightly more than 2,000. The topology on the current D-Wave Advantage series is called "Pegasus," and it has an average of 15 connections per qubit. Advantage processors have more than 5,000 qubits (the exact number varies between processors). The latest generation topology, which will be available in the future Advantage2 QPU and is already available in a prototype processor, is called "Zephyr" and has an average of 20 connections per qubit. While the Advantage2 will have more than 7,000 qubits, the prototype only comprises approximately 500 qubits.

As the number of connections per qubit is low compared with the number of available qubits in any D-Wave QPU (e.g., 15 connections for over 5,000 possible destination qubits), the graph defined by the connections of a given Ising or QUBO problem is rarely isomorphic to (a subgraph of) the hardware topology. To circumvent the connectivity problem of the QPU, it is often required to combine several physical qubits to form a single logical qubit with an effectively higher connectivity to represent a given logical variable of the problem (Choi, 2008, 2010). These sets of physical qubits, also called a qubit chain, have to act like a single qubit to produce valid outputs. To achieve this, a strong ferromagnetic coupling is applied between them. Such a procedure in which multiple qubits of the QPU are used to constitute a logical variable of the target problem is called minor embedding. The D-Wave Ocean SDK (D-Wave Systems Inc, 2023a) provides a heuristic algorithm to generate a minor embedding for a given Ising (or QUBO) problem. The coupling strength of the ferromagnetic coupling between the physical qubits of a qubit chain is called chain strength, and this hyperparameter can be set by the user.

The value of the chain strength has to be chosen carefully (Choi, 2008; Raymond et al., 2020). If the chain strength is chosen too weak, low-energy solutions of the embedded problem might favor solutions where qubit chains are broken, i.e., physical qubits that represent the same logical variable have different values. In such a case, post-processing has to be applied to obtain a valid logical variable from the physical qubits, but the resulting bitstring does not necessarily have to be a low-energy solution of the original problem. If the chain strength is chosen too strong, all qubit chains act as one logical variable but their actual value might be random. This is due to the limited range and precision of the h_i and J_{ij} values. The problem has to be rescaled too much so that the h_i and J_{ij} values become too small (i.e., below the precision limit) so that all valid solutions get too close and, thus, difficult to separate, leading to random outcomes. Moreover, in principle, the chain strength should be optimized to find the sweet spot (Grant and Humble, 2022). The default procedure "uniform torque compensation" (D-Wave Systems Inc, 2023b) provided in the Ocean SDK may (see Chen et al., 2021) or may not (see Carugno et al., 2022) work well enough.

Here, we use the relative chain strength r, i.e., we scan the chain strength by setting it to $s_c = rm$ in relation to the maximum value $m = \max(\max_i |h_i|, \max_{ij} |J_{ij}|)$, occurring in the particular problem instance.

2.2.2 Searching for better embeddings

Finding an optimal embedding generally is computationally intractable in the worst-case: When both the application graph *and* the hardware topology are part of its input, the problem is NP-hard [even when the set of topologies is limited to subgraphs of the Chimera or Pegasus graphs (Lobe and Lutz, 2021)]; when the hardware topology is fixed, the problem becomes polynomial but the large constants hidden in the big-O do not lead to practical algorithms (Roberston and Seymour, 1995). Therefore, generating an optimal embedding would annihilate any possible quantum advantage a QPU would provide. Nonetheless, quickly finding a good-enough embedding is required to try and solve any given problem on a D-Wave quantum annealer. Finding a good embedding can be advantageous because it reduces the complexity of the problem to solve on the QPU and may reduce the number of utilized qubits (Gilbert and Rodriguez, 2023) while taking potentially more time on the classical computer to be generated.

From our experiments, we find that the embeddings generated by the default heuristic provided in D-Wave's Ocean library can be quite far from optimal. Nonetheless, it has the advantage of ease of use so that it does not overly complicate the procedure of solving a problem on the QPU.

We try and probe what kind of gain can be achieved by looking at better embedding results. We note that we do not aim for a highly optimized embedding here but for a reasonably good one to assess the performance improvement compared with a randomly picked embedding of potentially poor or average quality. The methodology we utilize here is to use a large number of different seeds for the Ocean's embedding procedure to generate many different embeddings (between 100 for the larger instances and up to 500 for the small ones) and select the best outcome of this process. For

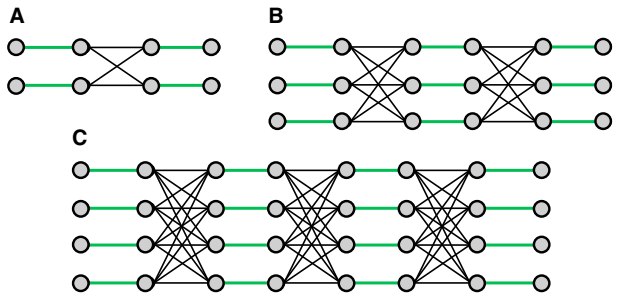

FIGURE 1
An illustration of the **(A)** G_1, **(B)** G_2, **(C)** G_3 maximum cardinality matching problems. The maximum matching is reached when all edges are selected in the sparse parts of the graph (in green).

all except the smallest problem size, we utilized the most obvious measure to estimate the quality of the embedding, namely, the number of qubits required by the embedding. Another option would be to also try to deduce the lengths of qubit chains. Trying to avoid particularly noisy parts of the QPU could be an additional consideration. For the smallest problem size, we could not apply either of the first two measures as all embeddings require eight physical qubits (on Pegasus and Zephyr topologies), which equal the number of the logical variables (i.e., no minor embedding is required). We show and discuss the results obtained from the QPU with these improved embeddings in Section 3.2.2 below. Of course, this modus operandi would not be considered for any consistent quantum advantage as such because of the amount of required pre-processing. However, used for benchmarking purposes, it can show the influence of the quality of the embedding.

2.3 The maximum cardinality matching problem

Given an undirected graph (V, E) where V is a set of vertices linked by a set of edges $E \subset V \times V$, the MCM problem is a combinatorial optimization problem aimed at finding the highest number of edges $|\tilde{E}|, \tilde{E} \subseteq E$, such that each vertex V is linked by at most one edge $e \in \tilde{E}$. While being a combinatorial problem, this problem is polynomial.

To demonstrate explicitly the slow convergence of SA in some cases, in 1988, Sasaki and Hajek (1988) devised a series of special variants of this class of problem called the G_n series. This particular series of problem instances is trivial to solve, but it is demonstrated to be exponentially hard to solve by utilizing SA.

The simplest problem of the series is the G_1 problem with only eight edges and one densely connected subgraph. Each increment in the series adds one new line of vertices and edges and one new densely connected subgraph. Therefore, G_2 adds 10 vertices and 19 edges resulting in 18 vertices and 27 edges. Only nine of these edges constitute the optimal solution. To summarize, for each instance of the G_n series, there are $2(n+1)^2$ vertices and $(n+1)^3$ edges but only $(n+1)^2$ edges in the optimal subgraph. Thus, the probability of selecting an adequate edge in the random selection operated by SA vanishes quickly. This provides a hint as to why this series of problem instances is exponentially difficult to solve for SA. Figure 1 provides an illustration of the G_1, G_2, and G_3 problems.

2.3.1 QUBO formulation

The MCM problem is a problem with constraints, and therefore, the MCM problem and the G_n series have to be transformed to a QUBO problem that can be executed on D-Wave machines. There are three steps required for this transformation. The first step is to associate relevant aspects of the original problem with binary variables, the second step is to find the cost function and constraints separately, and the third step is to add the constraints as soft constraints by incorporating them as penalties in the cost function.

The first two steps are usually closely related and also performed when the problem is solved classically with linear programming. The obvious choice for the binary variables, as we aim to maximize the number of edges in the graph, is to associate the binary variable x_e with the fact that a given edge $e \in E$ is selected for the matching ($x_e = 1$) or not ($x_e = 0$). Then, the cost function to be maximized would be $\mathcal{C}_0(\mathbf{x}) = \sum_{e \in E} x_e$, where E, as already defined, is the set of the edges of the G_n graph. The constraints are such that, for each vertex v, at most one of its associated edges is selected: $\sum_{e \in \Gamma(v)} x_e \leq 1$, where $\Gamma(v) \subseteq E$ denotes the set of edges associated to vertex v. Under the additional assumption that any maximum matching is a perfect one (which is the case for the G_n family of graphs), the penalty terms added to the cost function are then $\mathcal{C}_1 = \lambda \sum_{v \in V} \left(1 - \sum_{e \in \Gamma(v)} x_e\right)^2$ with $\lambda > 0$ and which are zero if and only if all vertices are associated with one and only one selected edge.

By constructing the total cost function $\mathcal{C}_t(\mathbf{x}) = -\mathcal{C}_0(\mathbf{x}) + \mathcal{C}_1(\mathbf{x})$ so that the cost is minimized when the solution is found, the following is obtained for the coefficients of the Q matrix:

$$q_{ee'} = \begin{cases} -1 - 2\lambda & \text{if } e = e' \\ 2\lambda & \text{if } e \neq e' \wedge \exists v \in V : (e \in \Gamma(v) \wedge e' \in \Gamma(v)) \\ 0 & \text{else}. \end{cases}$$

By choosing $\lambda = |E|$ as $|E| \geq \sum_{e \in E} x_e$, we ensure that the configuration with the minimal cost satisfies the soft constraints.

As an illustration, Figure 1 shows the graph for G_1, and the associated Q_{G_1} matrix is given by:

$$Q_{G_1} = \begin{bmatrix} -17 & 0 & 16 & 16 & 0 & 0 & 0 & 0 \\ & -17 & 0 & 0 & 16 & 16 & 0 & 0 \\ & & -17 & 16 & 16 & 0 & 16 & 0 \\ & & & -17 & 0 & 16 & 0 & 16 \\ & & & & -17 & 16 & 16 & 0 \\ & & & & & -17 & 0 & 16 \\ & & & & & & -17 & 0 \\ & & & & & & & -17 \end{bmatrix}.$$

It is easy to verify that the cost of the optimal solution is $\mathcal{C}_{G_n}(\mathbf{x}_{opt}) = -(n+1)^2 - 2(n+1)^5$, where $-2n(n+1)^5$ is an offset introduced by neglecting the constant term of $\mathcal{C}_1$ which is irrelevant to the optimization.

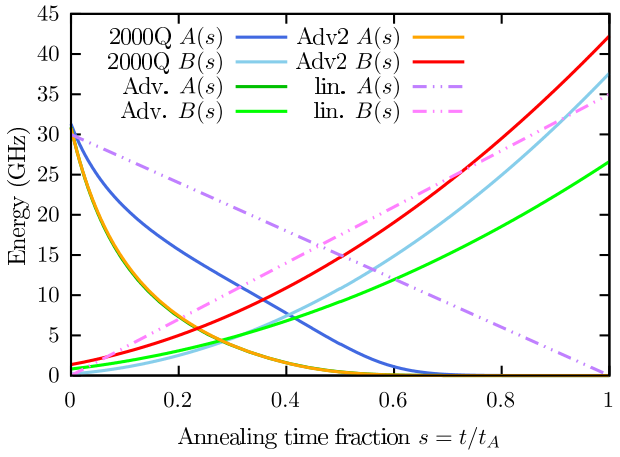

FIGURE 2
Annealing schedules of the DW_2000Q_6 (light and dark blue, dash-dotted), Advantage_system4.1 (light and dark green, solid), and Advantage2_prototype1.1 (red and orange, dashed) QPUs as well as a linear annealing schedule (pink and purple, dot-dash-dotted).

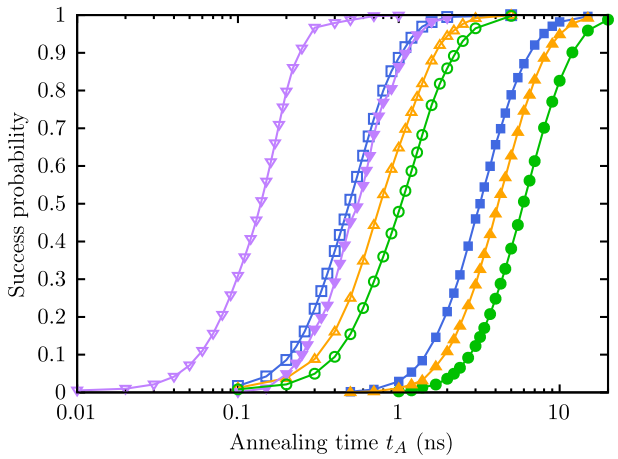

FIGURE 3
Success probability for the G_1 graph (open markers) and the G_2 graph (filled markers) as a function of annealing time for the DW_2000Q_6 (blue squares), Advantage_system4.1 (green circles), Advantage2_prototype1.1 (orange up-triangles), and linear (purple down-triangles) annealing schedules. No embedding is used for these simulation results.

2.4 Principles of ideal quantum annealing simulation

We simulate the (ideal) QA process by solving the time-dependent Schrödinger equation ($\hbar = 1$)

$$i\frac{\partial}{\partial t}|\psi(t)\rangle = \mathcal{H}(t)|\psi(t)\rangle$$

numerically, where $|\psi(t)\rangle$ denotes the state vector. For this, we use the Suzuki-Trotter product-formula algorithm (Trotter, 1959; Suzuki, 1976, 1985; De Raedt, 1987; Huyghebaert and De Raedt, 1990). This method yields a full state vector simulation of an ideal, closed system, i.e., we have direct access to the theoretical success probability, and we do not need to acquire an estimate from sampling. For the decomposition of the Hamiltonian, we use $\mathcal{H}(t) = A(t/t_A)\mathcal{H}_0 + B(t/t_A)\mathcal{H}_f$, with $A(s = t/t_A)$ and $B(s = t/t_A)$ either linear annealing schedules or the annealing functions of the particular D-Wave processor that we simulate. All annealing schedules are shown in Figure 2, and $\mathcal{H}_0$ and $\mathcal{H}_f$ are given in Equations (1, 2), respectively.

Since the memory requirement to store the full state vector grows exponentially, specifically with 2^N with N number of qubits, we have to use supercomputers with distributed memory to simulate systems with more than 30 qubits. The communication via the Message Passing Interface (MPI) follows the same scheme that is used for the gate-based quantum computer simulator JUQCS (De Raedt et al., 2007, 2019; Willsch et al., 2022b). The code utilizes OpenACC and CUDA-aware MPI to be run on GPUs. To speed up the simulation, we also run the 27-qubit cases on 4 GPUs (on one node).

The ideal QA simulation can be used to compare aspects of the D-Wave processors which are inaccessible on the real devices. For instance, we can compare the different annealing schedules of the processors without the need to apply minor embedding since in the simulation, we have all-to-all connectivity between the qubits.

3 Results and discussion

We present and discuss our results from QA simulation and QA on three generations of D-Wave processors and compare them with the results from SA. Note that most of the raw data utilized here and program to generate them are available in Benchmarking QA with MCM and problems (2023).

3.1 (Ideal) quantum annealing simulation

3.1.1 Influence of the annealing schedule

The shape of an annealing schedule influences the performance of a QA device and optimizing the annealing schedule is an active field of research (Farhi et al., 2002; Morita, 2007; Zeng et al., 2016; Brady et al., 2021; Mehta et al., 2021; Susa and Nishimori, 2021; Venuti et al., 2021; Chen et al., 2022; Hegde et al., 2022, 2023). The D-Wave annealing schedules are partly dictated by the hardware as the functions A and B cannot be chosen completely independently (Harris et al., 2010). We illustrate the influence of the annealing schedule on the difficulty of the problem exemplarily for the G_1 and G_2 graphs.

Figure 3 shows the success probability as a function of annealing time for four different annealing schedules. Three annealing schedules are taken from the DW_2000Q_6, Advantage_system_4.1, and Advantage2_prototype1.1 QPUs, the fourth is a linear one, and they are shown in Figure 2.

Since the results are obtained from (ideal) simulation of the QA process, no embedding is required regardless of the annealing schedule. Thus, differences in success probability are only due to the different annealing schedules. We find that using the linear annealing schedule requires a much shorter annealing time to reach the same success probabilities than using the D-Wave annealing

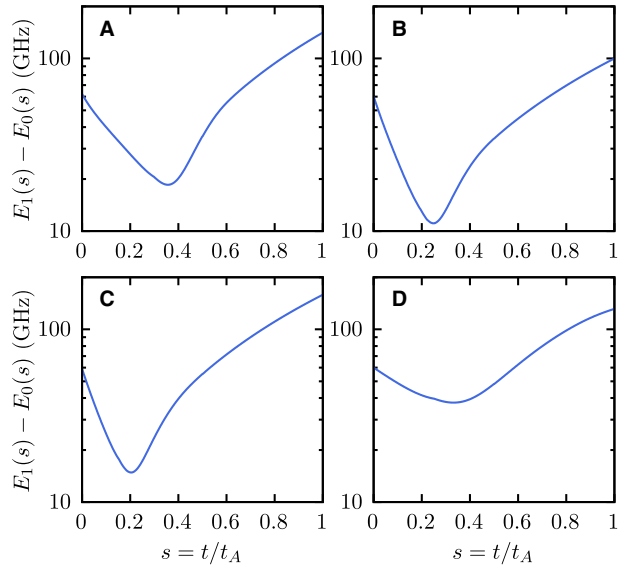

FIGURE 4
Energy $E_1(s) - E_0(s)$ between the ground state and the first excited state for the G_1 graph as a function of the anneal fraction s. **(A)** The DW_2000Q_6 annealing schedule is used. **(B)** The Advantage_system4.1 annealing schedule is used. **(C)** The Advantage2_prototype1.1 annealing schedule is used. **(D)** The linear annealing schedule is used.

schedules, which also show different performances: For the G_1 and G_2 graphs, using the DW_2000Q_6 annealing schedule performs better than using the Advantage2_prototype1.1 annealing schedule which performs better than the Advantage_system_4.1 annealing schedule. More subtle is the observation that by going from the G_1 to the G_2 problem, the required annealing time to reach a fixed success probability increases more for the DW_2000Q_6 annealing schedule than for the others.

The influence of the annealing schedule onto the difficulty of the problem can also be observed in the energy spectrum: The energy gap between the ground state and the first excited state can differ quite significantly between different annealing schedules. This is presented for the G_1 problem in Figure 4. The different panels show the energy gap during the annealing process for the DW_2000Q_6, Advantage_system4.1, Advantage2_prototype1.1, and linear annealing schedules for the same problem instance. Obviously, the energy gap between the ground state and the excited state is much smaller for the Advantage_system4.1 annealing schedule, which means the G_1 problem is then harder to solve, resulting in the longer annealing times required to achieve the same success probabilities as with the other annealing schedules. This might be different for other problems but especially for small and/or sparse problems, it was observed that DW_2000Q processors could achieve a better performance than Advantage processors (Calaza et al., 2021; Willsch et al., 2022a).

3.1.2 Influence of the embedding

To study the influence of the embedding on the performance, we consider three different embeddings for the G_1 problem. To exclude the influence of the annealing schedule, we only use the DW_2000Q_6 annealing schedule for all embeddings. We perform a scan of the relative chain strength and the annealing time.

The results are shown in Figure 5. On the left, we show the results of a scan of the relative chain strength for three different embeddings which require a different number of chains of different lengths. The embeddings onto their respective topologies are shown in Supplementary material. We normalize the problem to keep the values for h in the range $[-2, 2]$ and the values for J in the range $[-1, 1]$ as is the standard on DW_2000Q QPUs. Depending on the embedding, the maximal relative chain strength $r_{\text{max}}^{\text{scale}}$ which does not require further (compared with the unembedded problem) rescaling of the h and J parameters can be different. The minimal relative chain strength $r_{\text{min}}^{\text{sol}}$ for which the solution state is also the ground state depends on the embedding too. The relation between these two chain strengths is different for all three cases that we consider. Thus, the curves exhibit different features. For the embedding onto the Chimera topology with 4 chains of length 2, which we label by (i), $r_{\text{min}}^{\text{sol}} > r_{\text{max}}^{\text{scale}}$, the maximum success probability is reached for relative chain strengths larger than $r_{\text{min}}^{\text{sol}}$ and $r_{\text{max}}^{\text{scale}}$. For the embedding onto the modified Chimera topology with 2 chains of length 2, which we label by (ii), the solution state is always the ground state and the success probability is maximal for relative chain strengths smaller than $r_{\text{max}}^{\text{scale}}$. For the embedding onto the square grid with 2 chains of length 3, which we label by (iii) $r_{\text{max}}^{\text{scale}} > r_{\text{min}}^{\text{sol}}$, the maximum lies between $r_{\text{max}}^{\text{scale}}$ and $r_{\text{min}}^{\text{sol}}$. For embeddings (i) and (iii), there is a minimal relative chain strength one has to choose to achieve success probabilities significantly larger than zero. Due to the rescaling of the h and J parameters, one cannot choose an arbitrarily large relative chain strength as the success probability also drops for large values.

The right column in Figure 5 shows the success probability as a function of annealing time for different relative chain strengths. We consider for the success probability only the solution state without broken chains. This corresponds to discarding all samples with broken chains. Here, we include chain break fixes by majority vote (labeled by "mv" in the legend), i.e., we assign to the logical variable the value of the majority of the qubits in a chain. If the numbers in a chain is equal, we multiply the probability of this particular state by 0.5 for each such chain. For the first and second embeddings, the improvement by including chain break fixes is marginal. For embedding (iii), however, the chain break fix by majority vote can improve the performance a lot so that the success probability after majority vote gets close to the success probability without embedding. This might be surprising at first as the chain length is largest for this case (3 compared with 2 for the other 2 embeddings). However, if only one qubit is flipped in a chain of length 3, majority vote fixes it with 100%, while for the chains of length 2, the probability to fix the chain correctly is only 50%.

3.2 The G_n series on D-Wave quantum processors

The first instances of the G_n series were run on several generations of the D-Wave quantum annealer. Here, we present our results obtained with the default embeddings and the improved

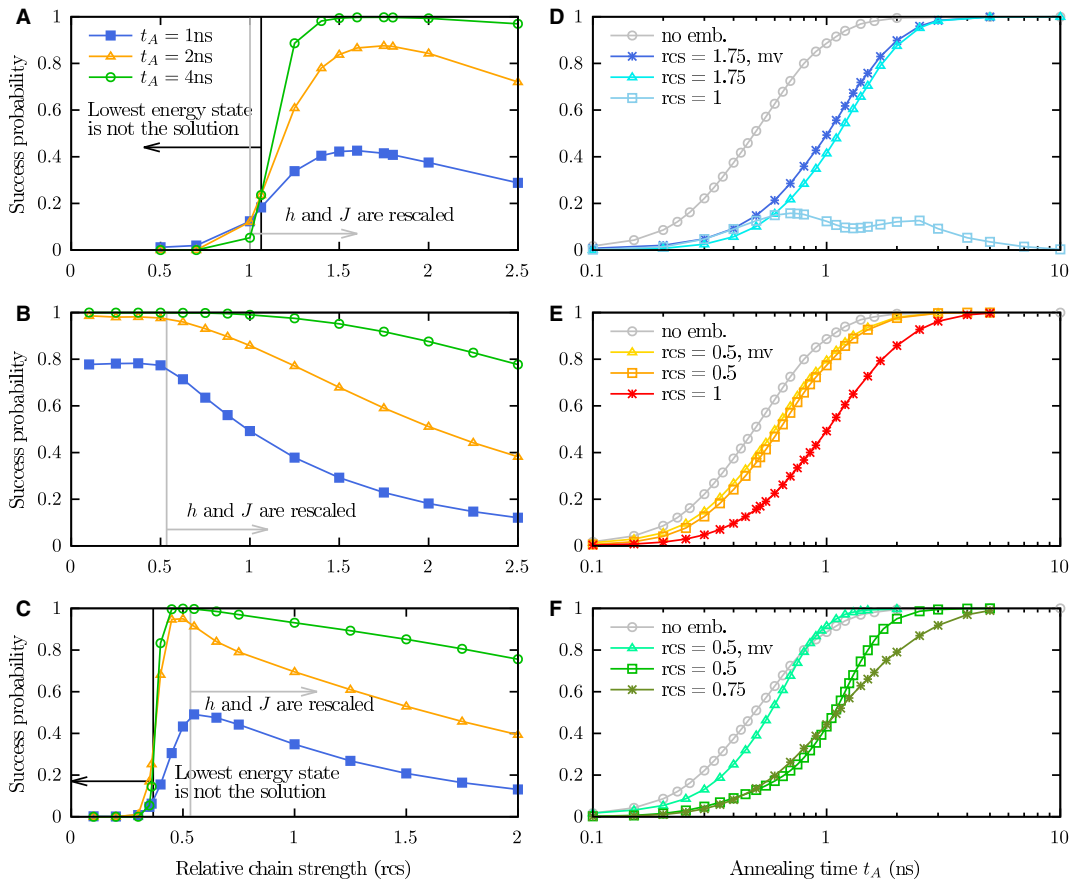

FIGURE 5
Success probability for the G_1 graph **(A–C)** as a function of the relative chain strength for different annealing times indicated in the legend and **(D–F)** as a function of annealing time for different chain strengths indicated in the legend. The DW_2000Q_6 annealing schedule is used, and the rows show different embeddings. Only the solution state is counted as success; the results with broken chains are not considered unless indicated by "mv" which means chains are fixed by majority vote. In case of the chains of length 2, the chance to fix a chain is 50%. The success probability for the unembedded case is shown in gray for comparison. **(A, D)**: Embedding onto the Chimera topology with 4 chains of length 2. **(B, E)**: Embedding onto a modified Chimera topology with additional couplers yielding 2 chains of length 2. **(C, F)**: Embedding onto a square grid with 2 chains of length 3.

ones. For reference, Table 1 shows the number of variables and the optimal energy for the graphs G_1 to G_7.

TABLE 1 Number of variables and optimal energies for G_1 up to G_7.

	# var	Opt
G_1	8	−68
G_2	27	−495
G_3	64	−2064
G_4	125	−6275
G_5	216	−15588
G_6	343	−33663
G_7	512	−65600

3.2.1 Results with default embedding procedures

D-Wave's Ocean SDK (D-Wave Systems Inc, 2023a) provides a heuristic procedure to generate minor embeddings of the problem graph onto the hardware's topology. We use 5–10 different seeds for this heuristic embedding procedure to avoid obtaining an unluckily bad embedding. Since the chain strength can have a strong influence on the success rate, we scan the relative chain strength and the annealing time as well to achieve high success rates on the different processors. The chosen relative chain strength and annealing time values are shown in the Supplementary material.

The results are shown in Table 2 for the processors Advantage_system4.1, Advantage_system5.2, Advantage_system6.1, DW_2000Q_6, and Advantage2_prototype1.1. We obtain the optimal solution up to G_3 on the Advantage processors. For DW_2000Q_6, we obtained the solution for G_1 and G_2. The required number of physical qubits is much higher on this processor due to the sparser connectivity of the Chimera topology, and since this processor has approximately 2,000 qubits, we are able to find an embedding only up to G_4 while we can embed problems until G_7 onto the Pegasus topology of the Advantage_system processors with approximately 5,000 qubits. The Advantage2_prototype1.1 with its Zephyr topology has an even higher connectivity than the Advantage_system processors, but we can only find an embedding up to G_4 due to the limited number of qubits which is approximately 500.

TABLE 2 Results obtained on various D-Wave processors for the G_n series.

	#qbs	#Opt	Best	Worst	Mean	Median	Stdev
G_1	8	1,000	−68	−68	−68	−68	0
	8	1,000	−68	−68	−68	−68	0
	8	993	−68	−52	−67	−68	1.3
	8	1000	−68	−68	−68	−68	0
	13	**998**	**−68**	**−51**	**−68**	**−68**	**0.7**
G_2	42	840	−495	−335	−486	−495	21.3
	41	951	−495	−388	−492	−495	12.5
	43	863	−495	−387	−487	−495	19.4
	37	912	−495	−388	−490	−495	15.4
	84	**849**	**−495**	**−387**	**−488**	**−495**	**19.1**
G_3	144	9	−2,064	−1,171	−1,736	−1,683	140.9
	144	15	−2,064	−1,298	−1,802	−1,808	125.8
	150	27	−2,064	−1,423	−1,773	−1,808	127.7
	121	110	−2,064	−1,551	−1,884	−1,936	104.6
	315	**0**	**−1,937**	**−1,295**	**−1,608**	**−1,554**	**98.5**
G_4	427	0	−5,527	−3,526	−4,766	−4,776	308.5
	424	0	−5,526	−3,775	−4,947	−5,024	279.2
	396	0	−5,527	−3,278	−4,514	−4,527	358.7
	322	0	−5,775	−4,030	−5,042	−5,027	227.5
	917	**0**	**−5277**	**−3774**	**−4785**	**−4775**	**232.0**
G_5	923	0	−12,994	−8,676	−11,122	−11,266	733.4
	952	0	−13,428	−9,114	−11,779	−11,702	615.9
	987	0	−12,997	−7,817	−11,396	−11,271	686.9
G_6	2,018	0	−27,487	−18,574	−23,302	−23,374	1,463.9
	1,853	0	−27,489	−19,943	−24,008	−24,058	1,309.0
	1,880	0	−27,488	−17,883	−23,136	−23,373	1,528.0
G_7	3,573	0	−51,266	−31,819	−41,678	−42,048	2,807.0
	3,562	0	−52,284	−36,921	−44,852	−45,113	2,413.0
	3,410	0	−50,239	−34,883	−43,029	−43,072	2,827.0

For each graph, rows show the results on Advantage_System4.1, Advantage_System5.2, and Advantage_System6.1 (until G_7) and Advantage2_prototype1.1 and DW_2000Q_6 (until G_4, resp. gray background and gray background with bold font). The columns show the graph, the number of physical qubits (after minor embedding), the number of optimal solutions found, the best, worst, mean and median energies, and the standard deviation of the energies obtained. The sample size is 1,000, and the used annealing times and relative chain strengths are given in the Supplementary material.

Taking into account the results on the older generation D-Wave 2X processor which can be found in the study mentioned in the reference (Vert et al., 2021), we find significant improvement from D-Wave 2X (5.1%) to D-Wave 2000Q (84.9%) for the G_2 graph. For the G_1 graph, all processors perform quite well, and there is no significant difference in success rates between the different generations. Similarly, there is not much difference in the success rates between the D-Wave 2000Q and newer generations for the G_2 graph, although the embedding onto the DW_2000Q_6 requires approximately twice as many physical qubits as embeddings onto the newer processors. For the G_3 graph, neither D-Wave 2X nor D-Wave 2000Q returned the optimal solution in our case, but the "best" energy returned by the D-Wave 2000Q is lower than the D-Wave 2X. We note that in the study mentioned in the reference (McLeod and Sasdelli, 2022), D-Wave 2000Q was able to return the optimal solution at least once. Advantage_system and Advantage2 prototype processors were able to sample the optimal solution several times, and with improved embedding (see below), Advantage2 prototype even reached a success rate of approximately 50%. For G_4, none of the tested processors returned the optimal solution, and surprisingly, D-Wave 2X and the Advantage_system processors all achieved (almost) the same "best" energy (unless an improved embedding was used on Advantage_system4.1). McLeod and Sasdelli (2022) also found the same "best" energy on the

DW_2000Q processor, and a slightly lower "best" energy on Advantage_system4.1 is very close to the one which we obtained with the Advantage2 prototype processor. For graphs G_5 and higher, only embeddings onto the Advantage_system processors were possible, and the results are compatible with each other. We note that the values of the energies and the standard deviation are stretched due to the choice of $\lambda = |E| = (n+1)^3$. Division by λ (as is also done to fit the values into the parameter ranges of the QPUs) would give more compact values and especially standard deviations, which are more comparable among the different instances. However, we decided not to divide by λ to keep the easy comparison to the previous studies in Vert et al. (2021); McLeod and Sasdelli (2022).

3.2.2 Results with improved embeddings

Table 3 shows the results of the Advantage_system4.1 QPU from G_2 to G_7 for a random embedding and the improved embedding (cf. Section 2.2.2). Table 2 shows the results of the Advantage2_prototype1.1. The annealing time was set to 500 μs, and the chain strength values are shown in Supplementary material.

The Advantage2 prototype is a preview of what will be available for the future Advantage2 full system. Its main limitation, when compared with the planned release QPU, is the number of available qubits which is only approximately 500 compared with the over 7,000 that are planned for the release-ready QPU. This limitation means we can only test the chip up to G_4.

We find that usually the solution quality is comparable; however, there are cases (for instance G_3) where the improved embedding also leads to a significant improvement in the solution quality (increased probability of finding the solution). The number of utilized qubits for an improved embedding is typically approximately 7% to 17% lower than the random one.

At first glance of Table 4, we can observe the number of qubits required for the random embedding comes close to the number of qubits required for the improved embedding of the Advantage_system4.1. As can be expected, the results become much better than what can be achieved on systems 2X and 2000Q.

The fact that the Pegasus architecture provides more connections means that any heuristic for generating a better embedding has more options and therefore the chance of finding a good embedding is much higher. This can decrease the complexity of finding a better heuristic for this particular QPU, hence greatly improving the overall quality of the solutions.

3.3 Comparison of SA, ideal QA, and QA on D-Wave processors

As the G_n series was especially crafted to probe the SA meta-heuristic, it is also a potential good candidate to probe the QA-based meta-heuristics. In this study, we did so with an ideal QA process simulated from the Schrödinger equation on a standard supercomputer, and we also completed a full set of experiments on several generations of D-Wave's quantum annealers. The latter requires to take into account several parameters that can have significant impacts on the results such as relative chain strength, annealing time, and the particular embedding.

As stated in the study mentioned in Vert et al. (2021), finding the optimal solution without the embedding for the G_n series up to G_7 is not difficult for SA with a relatively low number of iterations. Nonetheless, taking the embedding into account, D-Wave QPUs compare very favorably to SA. When using the same embedding of the Chimera topology for the cases up to G_4 also for SA, SA yields similar results as the 2X QPU only when utilizing $\alpha \text{card}(E)^2$ annealing steps (α being chosen around 1,000) per plateau of temperature, which is considered a costly but usually accurate parametrization of SA. Table 5 shows the results of SA on the embeddings onto the Advantage_system4.1 QPU. As can be observed, the D-Wave results remain mostly comparable to the same high-quality SA heuristic with $\alpha \text{card}(E)^2$ annealing steps per plateau of temperature. Nonetheless, it is worth noticing that obtaining these SA results with G_7 costed more than 5 days of computing time on a AMD 7,702 P, 2 GHz with 64 cores (C++ optimized SA code running in parallel with several instances per core to obtain a statistical significance). In this regard, we can conclude that even by taking the pre-processing into account, the Advantage_system4.1 QPU compares well with SA on the embedded versions of the G_n series.

Table 6 shows the same results with the embedding of the Advantage2 prototype. In this case, we can too strong observe that the Advantage2 prototype outperforms any reasonably parameterized SA metaheuristic on embedded problems both in quality and in processing time (which may also be transferable to better energy performance).

We compare the results from ideal QA obtained by simulation to the results obtained on the Advantage2_prototype1.1 processor. In Figure 6, we show the success probability obtained by ideal QA for the G_2 problem instance.

Three cases are shown: The unembedded instance with the maximum h-range on the Advantage processors ($h_{\max} = 4$) indicated by orange asterisks, the unembedded instance with maximum h-range on D-Wave 2,000Q processors ($h_{\max} = 2$) indicated by blue squares, and the embedded instance requiring 33 qubits using a relative chain strength of 0.35 with the maximum h-range of the Advantage processors indicated by red triangles. The latter two are actually quite close, suggesting that the embedding (with reasonably good chain strength, cf. Figure 7) has similar effects than a rescaling of the problem parameters h_i and J_{ij}. This is not totally surprising since a (required minimum) chain strength may force the problem parameters to be rescaled (cf. Figure 5A).

Figure 8 shows the success probability obtained by ideal QA for the embedded case in comparison to the results obtained on the Advantage2_prototype1.1 with the same embedding. Obviously, the time scales are very different. While in the ideal case approximately 20 ns is sufficient to reach a success probability of approximately 1, the shortest time possible on the D-Wave processor is 1 μs where the success rate only reaches approximately 0.9. For an annealing time of $\approx$ 35 μs, the success rate exceeds 0.98. From this observation, we conclude that for this problem instance, the D-Wave processor works in the quasistatic regime (Amin, 2015) as the ideal simulation shows that the coherent regime would have to be at approximately 1–20 ns. We note that the time to simulate the ideal QA process requires much more time (on the

TABLE 3 Results for the G_n series obtained on the Advantage_system4.1 processor for a random embedding (G_n^{rand}) and the improved embedding (G_n^{imp}).

	#qbs	#opt	best	worst	av. best	av. worst	av. mean	av. std
G_2^{rand}	44	981.3 ± 10.2	-495	-388	-495.0 ± 0.0	-434.1 ± 16.9	-494.0 ± 0.6	7.0 ± 2.0
G_2^{imp}	38	975.1 ± 6.0	-495	-387	-495.0 ± 0.0	-428.9 ± 21.6	-493.7 ± 0.3	8.4 ± 1.0
G_3^{rand}	142	61.8 ± 23.0	$-2,064$	$-1,172$	$-2,064.0 \pm 0.0$	$-1,422.8 \pm 53.7$	$-1,825.6 \pm 19.0$	122.8 ± 3.6
G_3^{imp}	129	167.5 ± 27.9	$-2,064$	$-1,298$	$-2,064.0 \pm 0.0$	$-1,547.2 \pm 61.8$	$-1,906.0 \pm 10.4$	101.0 ± 2.5
G_4^{rand}	438	0.0 ± 0.0	$-5,776$	$-3,533$	$-5,546.7 \pm 67.3$	$-3,992.9 \pm 172.0$	$-4,965.1 \pm 25.7$	229.2 ± 6.6
G_4^{imp}	360	0.0 ± 0.0	$-6,025$	$-3,776$	$-5,811.2 \pm 86.2$	$-3,983.2 \pm 169.9$	$-5,076.1 \pm 32.4$	279.3 ± 7.9
G_5^{rand}	935	0.0 ± 0.0	$-13,861$	$-8,250$	$-13,257.3 \pm 227.5$	$-9,379.8 \pm 309.2$	$-11,668.2 \pm 74.5$	620.0 ± 16.9
G_5^{imp}	870	0.0 ± 0.0	$-13,862$	$-9,113$	$-13,490.8 \pm 149.1$	$-10,019.1 \pm 289.1$	$-12,217.1 \pm 53.8$	555.5 ± 16.5
G_6^{rand}	1,971	0.0 ± 0.0	$-28,865$	$-17,208$	$-28,191.0 \pm 422.0$	$-19,825.0 \pm 652.5$	$-24,574.9 \pm 146.8$	$13,08.6 \pm 37.2$
G_6^{imp}	1,801	0.0 ± 0.0	$-28,865$	$-19,266$	$-28,739.1 \pm 262.6$	$-20,949.8 \pm 630.9$	$-25,596.2 \pm 174.1$	$1,209.7 \pm 43.1$
G_7^{rand}	3,635	0.0 ± 0.0	$-55,361$	$-32,847$	$-53,907.7 \pm 649.7$	$-38,367.4 \pm 1,661.8$	$-47,159.0 \pm 657.4$	2397.5 ± 109.8
G_7^{imp}	3,293	0.0 ± 0.0	$-55,360$	$-35,910$	$-53,478.4 \pm 747.6$	$-38,018.6 \pm 1,145.5$	$-46,677.0 \pm 384.6$	2366.4 ± 59.6

The columns show the graph with the corresponding embedding, the number of (physical) qubits, the average number of optimal solutions found, the best, worst, average best, average worst, and average mean energies, and the average standard deviation of the energies obtained. The sample size is 1,000, averages are over 50 repetitions and given with standard deviation. Best and worst energies are over all samples of all repetitions.

TABLE 4 Results for the G_n series obtained on the Advantage2_prototype1.1 processor for a random embedding (G_n^{rand}) and the improved embedding (G_n^{imp}).

	#qbs	#opt	best	worst	av. best	av. worst	av. mean	av. std
G_2^{rand}	37	996.0 ± 3.3	-495	-440	-495.0 ± 0.0	-442.6 ± 7.5	-494.8 ± 0.2	3.1 ± 1.3
G_2^{opt}	33	999.9 ± 0.2	-495	-442	-495.0 ± 0.0	-491.8 ± 12.6	-495.0 ± 0.0	0.1 ± 0.4
G_3^{rand}	126	278.4 ± 110.1	$-2,064$	$-1,555$	$-2,064.0 \pm 0.0$	$-1,669.4 \pm 45.5$	$-1,952.2 \pm 20.0$	80.9 ± 6.9
G_3^{opt}	112	724.6 ± 103.9	$-2,064$	$-1,554$	$-2,064.0 \pm 0.0$	$-1,750.3 \pm 77.2$	$-2,027.5 \pm 14.0$	58.7 ± 7.1
G_4^{rand}	333	0.0 ± 0.0	$-5,777$	$-4,278$	$-5,687.1 \pm 119.3$	$-4,474.5 \pm 102.8$	$-5,185.6 \pm 35.1$	188.2 ± 11.8
G_4^{opt}	307	0.0 ± 0.0	$-6,026$	$-4,529$	$-5,881.5 \pm 122.8$	$-4,773.4 \pm 92.6$	$-5,429.7 \pm 36.4$	171.4 ± 12.5

The columns show the graph with the corresponding embedding, the number of (physical) qubits, the average number of optimal solutions found, the best, worst, average best, average worst, and average mean energies, and the average standard deviation of the energies obtained. The sample size is 1,000, averages are over 50 repetitions and given with standard deviation. Best and worst energies are over all samples of all repetitions.

TABLE 5 SA results using the embedding graphs onto the Pegasus topology of the Advantage_system4.1 QPU.

	#iter	#Opt	Best	Worst	Mean	Median	Stdev
G_1	n	1,000	-68	-68	-68	-68	0
	n^2	1,000	-68	-68	-68	-68	0
G_2	n	187	-495	-334	-435	-442	38.8
	n^2	937	-495	-389	-491	-495	13.2
G_3	n	0	$-1,682$	-787	$-1,336$	$-1,300$	148.8
	n^2	0	$-1,937$	$-1,299$	$-1,626$	$-1,680$	107.2
G_4	n	0	$-5,601$	$-4,026$	$-4,849$	$-4,844$	225.0
	n^2	0	$-6,054$	$-5,048$	$-5,585$	$-5,570$	185.6
G_5	n	0	$-13,367$	$-9,355$	$-11,395$	$-11,404$	633.8
	n^2	0	$-14,949$	$-13,125$	$-13,813$	$-13,775$	286.9
G_6	n	0	$-27,019$	$-18,867$	$-23,091$	$-23,131$	1,370.4
	n^2	0	$-33,245$	$-30,524$	$-32,112$	$-32,119$	470.4
G_7	n	0	$-37,073$	$-11,152$	$-24,083$	$-24,187$	4,137.3
	n^2	0	$-58,859$	$-52,187$	$-56,192$	$-56,204$	999.3

The columns show the graph, the number of iterations, the number of optimal solutions found, the best, worst, mean and median energies, and the standard deviation of the energies obtained. The sample size is 1,000.

TABLE 6 SA results using the embedding graphs onto the Zephyr topology of the Advantage2_prototype1.1 QPU.

	#iter	#Opt	Best	Worst	Mean	Median	Stdev
G_1	n	1000	−68	−68	−68	−68	0
	n^2	1,000	−68	−68	−68	−68	0
G_2	n	840	−495	−389	−486	−495	19.6
	n^2	997	−495	−442	−495	−495	2.9
G_3	n	0	−1,937	−1,299	−1,628	−1,681	94.2
	n^2	118	−2,064	−1,681	−1,889	−1,937	98.5
G_4	n	0	−5,528	−3,843	−4,761	−4,778	259.5
	n^2	0	−6,026	−4,979	−5,433	−5,526	190.2

The columns show the graph, the number of iterations, the number of optimal solutions found, the best, worst, mean and median energies, and the standard deviation of the energies obtained. The sample size is 1,000.

order of minutes to hours depending on system size and simulated annealing time).

3.4 Effects of systematic errors in the QPU

When a given problem is assigned to a D-Wave QPU, there is a limited precision and several sources of potential noise. While some of the sources of errors and imprecisions are compensated for by the electronic controllers and the base software, some of these so-called ICEs (Integrated Control Errors) are intrinsic, e.g., the precision of the Digital to Analog (DAC) converters to control the couplers. As a consequence, any user of the QPU can not expect an infinite precision of the given values of the coupling. The D-Wave documentation points at an expected precision for coupling values between 0.1% and 2%.

When dividing by 2λ, the non-zero off-diagonal elements of the QUBO matrix take the value $q_{ij} = 1$ and the diagonal elements $q_{ii} = -1 - 1/2\lambda$, which is between -1 and -2 for $\lambda \geq 0.5$, and thus does not require further rescaling.

Nonetheless, the important element to consider is the associated Ising model as this is what is actually mapped to the QPU after renormalizing the values in the interval $[-1, 1]$ for J_{ij} and $[-2, 2]$ (on D-Wave 2,000Q and older systems) or $[-4, 4]$ for h_i (on Advantage and Advantage2 systems). Going from the QUBO formulation to the Ising model (where we use the convention $x_i = (1 - s_i)/2$, cf. also Section 2.2) yields non-zero $J_{ij} = \lambda/2$ (it is non-zero for all adjacent edges j of a given edge i) and $h_i \in \{(n\lambda - \lambda - 1)/2, n\lambda - 1/2\}$ for the outermost edges and all the others, respectively. The dependence on n of the h_i arises from the number of edges connected to the vertices which also grows with n. Rescaling by $2/n$ yields $h_i \in \{\lambda - (\lambda + 1)/n, 2\lambda - 1/n\}$ and $J_{ij} = \lambda/n$. As we can observe, the coupling matrix J tends toward 0 in an inverse law of n independent of the choice of λ as choosing a λ growing with n enforces a rescaling by $1/\lambda$ due to the h_i.

Nonetheless, the coupling should be an order of magnitude above the noise and biases of ICE up to $n \approx 20$ considering a 0.5% accuracy or better on the coupling. Therefore, this particular point would not constitute a large contributor to the limitation of the D-Wave QPUs in the near term future. It can thus be considered safe to assume that the limitations of D-Wave compared with ideal QA are due to the other limitations of the QPU in our experiments.

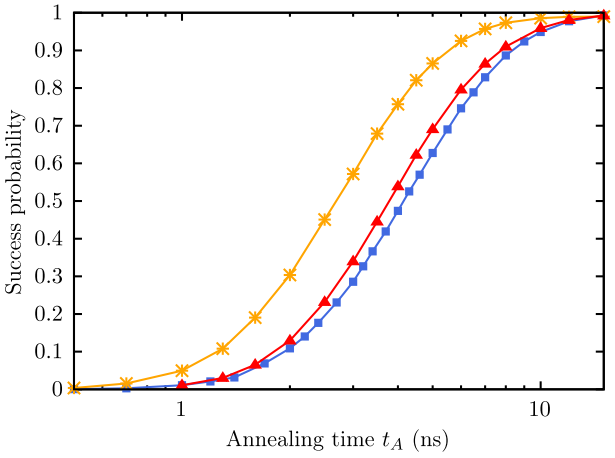

FIGURE 6
Success probability obtained from simulation for the G_2 graph as a function of annealing time for an embedding onto the Zephyr topology (rcs = 0.35, red triangles) with the Advantage2_prototype1.1 annealing schedule in comparison to direct embedding with the allowed range on Advantage processors ($-4 \leq h_i \leq 4$, orange asterisks) and direct embedding with the reduced range available on DW2000Q QPUs ($-2 \leq h_i \leq 2$, blue squares).

4 Conclusion and outlook

In summary, we have benchmarked three generations of D-Wave quantum processors by studying the performance when using different embeddings and by comparing to simulated annealing and ideal quantum annealing (i.e., solving the Schrödinger equation numerically) using a particular series (the G_n series) of maximum cardinality matching problems. From ideal quantum annealing with all-to-all connectivity, we find that for the problems without minor embedding (at least for the smaller instances), the annealing schedule of the DW_2000Q performs better, but this improvement is compensated by the decrease in

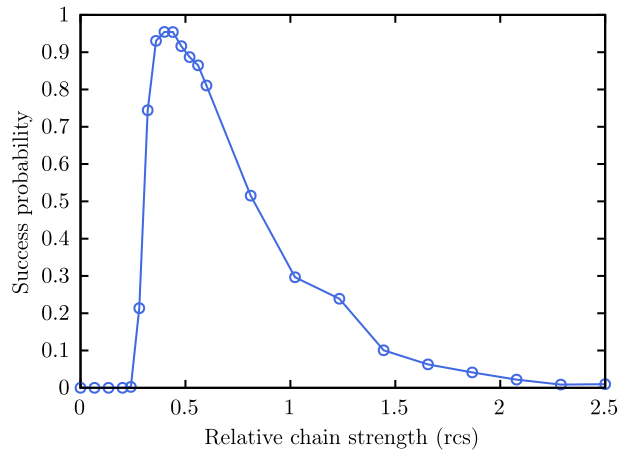

FIGURE 7
Relative chain strength scan for G_2 with a 33-qubit embedding on Advantage2_prototype1.1. with annealing time 1.0 μs and sample size 10,000.

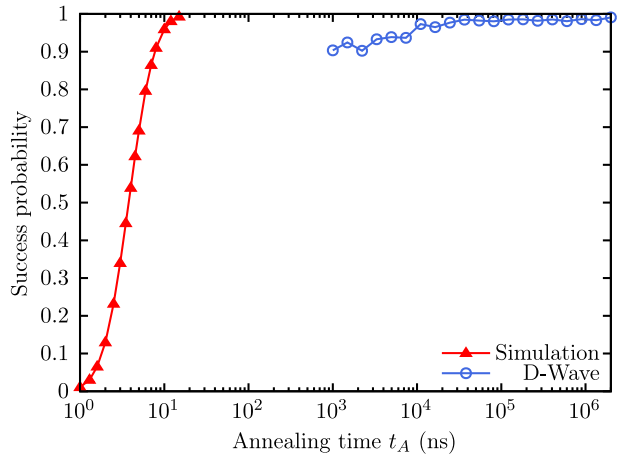

FIGURE 8
Success probability for the G_2 graph as a function of annealing time for a 33-qubit embedding onto the Zephyr topology obtained on the D-Wave quantum processor Advantage2_prototype1.1 (blue circles) and by simulation using the same annealing schedule (red triangles). The relative chain strength was set to 0.35. The sample size on the D-Wave processor is 10,000.

performance due to the required embedding onto the Chimera topology. Results for the G_1 and G_2 graphs are comparable between DW_2000Q_6 and Advantage processors, which is reasonable given the investigation using ideal QA. Although embeddings onto the Pegasus topology are possible with less than 2,000 qubits ($\approx$ the size of the DW_2000Q processors) up to G_6, we could not find an embedding onto DW_2000Q_6 for G_5. Moreover, not only the size of the processor but also the connectivity between the qubits are important to map larger problems onto the QPU's hardware graph.

For the Advantage_system4.1 and Advantage2_prototype1.1 processors, we explicitly compared the performance with different embeddings (a random one and the one with the least number of qubits found).

For most of the embeddings, we found little variation in the performance (either in the success rate if the optimal solution was found or in the minimal energy obtained). Only in a few cases, we found a significant difference suggesting that the utilized embedding might have been an unluckily bad one. Thus, trying more than a single embbedding can be useful, but trying to find an embedding with a particularly low number of qubits does not seem to be worth the effort.

The comparison to SA underlines the drop in performance when minor embedding has to be applied. While the unembedded problems are easy to solve with SA, especially the larger ones (G_3 or larger) become increasingly difficult to solve with SA if the embedding is taken into account. The performance when utilizing the embedding onto the Zephyr topology is worse than what we obtained from the Advantage2_prototype1.1 processor even for n^2 iterations per plateau of temperature. When using the embedding onto the Pegasus topology, the performance of SA is comparable (sometimes better sometimes worse) to the performance of the Advantage_system4.1 processor. However, the actual computing time required to obtain the SA results was significantly larger than what was used on the QPUs.

In conclusion, we find that the main bottleneck is the minor embedding required to solve an arbitrary problem on a QPU. The annealing schedule does have an influence but at least in the current study, it was found that it could only compensate for the larger embeddings required for DW_2000Q compared with Advantage_system in the cases of the smaller instances.

Possible future directions to continue the present study could include the benchmark of other embedding algorithms (such as Lucas, 2019; Zbinden et al., 2020) or a more in-depth investigation of annealing schedule variations such as pausing and quenching or reverse annealing in the spirit of Ikeda et al. (2019); Marshall et al. (2019); Venturelli and Kondratyev (2019); Gonzalez Izquierdo et al. (2021); Grant et al. (2021); Carugno et al. (2022). Another interesting path to include in these studies would be to consider the general QUBO formulation studied by McLeod and Sasdelli (2022) and different values for λ in the soft constraint to investigate whether all three algorithms (SA, ideal QA, and QA on the D-Wave) show the same behavior in performance change.

Data availability statement

The program code to generate the problem instances and to perform the experiments on the QPUs is publicly available in the accompanying repository (https://jugit.fz-juelich.de/qip/benchmarking-qa-with-mcm-problems). Additional information is given in the Supplementary material. Further inquiries can be directed to the corresponding authors.

Author contributions

DV: Investigation, Software, Writing – original draft. MW: Investigation, Software, Writing – original draft. BY: Investigation, Software, Writing – review & editing. RS: Writing – review & editing. SL: Software, Supervision, Writing – original draft. KM: Supervision, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. MW acknowledges support from the project Jülich UNified Infrastructure for Quantum computing (JUNIQ) that has received funding from the German Federal Ministry of Education and Research (BMBF) and the Ministry of Culture and Science of the State of North Rhine-Westphalia. BY acknowledges funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy — Cluster of Excellence Matter and Light for Quantum Computing (ML4Q) EXC 2004/1 — 390534769. The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC). The authors gratefully acknowledge the Jülich Supercomputing Centre (https://www.fzjuelich.de/ias/jsc) for funding this project by providing computing time on the D-Wave Advantage™ System JUPSI through the Jülich UNified Infrastructure for Quantum computing (JUNIQ). The work presented in this paper has been also supported by AIDAS - AI, Data Analytics and Scalable Simulation - which is a Joint Virtual Laboratory gathering the Forschungszentrum Jülich (FZJ) and the French Alternative Energies and Atomic Energy Commission (CEA). Open Access publication funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) — 491111487.

Conflict of interest

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

Publisher's note

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

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2024.1286057/full#supplementary-material

References

Acharya, R., Aleiner, I., Allen, R., Andersen, T. I., Ansmann, M., Arute, F., et al. (2023). Suppressing quantum errors by scaling a surface code logical qubit. *Nature* 614, 676–681. doi: 10.1038/s41586-022-05434-1

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

Albash, T., Vinci, W., Mishra, A., Warburton, P. A., and Lidar, D. A. (2015). Consistency tests of classical and quantum models for a quantum annealer. *Phys. Rev. A* 91:042314. doi: 10.1103/PhysRevA.91.042314

Amin, M. H. (2015). Searching for quantum speedup in quasistatic quantum annealers. *Phys. Rev. A* 92:052323. doi: 10.1103/PhysRevA.92.052323

Anthony, M., Boros, E., Crama, Y., and Gruber, A. (2017). Quadratic reformulations of nonlinear binary optimization problems. *Mathem. Progr.* 162, 115–144. doi: 10.1007/s10107-016-1032-4

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

Benchmarking QA with MCM and problems (2023). *Accompanying repository*. Available online at: https://jugit.fz-juelich.de/qip/benchmarking-qa-with-mcm-problems

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

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

Brady, L. T., Baldwin, C. L., Bapat, A., Kharkov, Y., and Gorshkov, A. V. (2021). Optimal protocols in quantum annealing and quantum approximate optimization algorithm problems. *Phys. Rev. Lett.* 126:070505. doi: 10.1103/PhysRevLett.126.070505

Calaza, C. D. G., Willsch, D., and Michielsen, K. (2021). Garden optimization problems for benchmarking quantum annealers. *Quant. Inf. Proc.* 20:305. doi: 10.1007/s11128-021-03226-6

Carugno, C., Dacrema, M. F., and Cremonesi, P. (2022). Evaluating the job shop scheduling problem on a D-Wave quantum annealer. *Sci. Rep.* 12:6539. doi: 10.1038/s41598-022-10169-0

Ceselli, A., and Premoli, M. (2023). On good encodings for quantum annealer and digital optimization solvers. *Sci. Rep.* 13:5628. doi: 10.1038/s41598-023-32232-0

Chancellor, N., Crowley, P. J. D., Durić, T., Vinci, W., Amin, M. H., Green, A. G., et al. (2022). Error measurements for a quantum annealer using the one-dimensional Ising model with twisted boundaries. *NPJ Quant. Inf.* 8:73. doi: 10.1038/s41534-022-00580-w

Chen, J., Stollenwerk, T., and Chancellor, N. (2021). Performance of domain-wall encoding for quantum annealing. *IEEE Trans. Quant. Eng.* 2, 1–14. doi: 10.1109/TQE.2021.3094280

Chen, Y.-Q., Chen, Y., Lee, C.-K., Zhang, S., and Hsieh, C.-Y. (2022). Optimizing quantum annealing schedules with monte carlo tree search enhanced with neural networks. *Nat. Mach. Intell.* 4, 269–278. doi: 10.1038/s42256-022-00446-y

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

Choi, V. (2010). Minor-embedding in adiabatic quantum computation: II. Minor-universal graph design. *Quant. Inf. Proc.*10, 343–353. doi: 10.1007/s11128-010-0200-3

De Raedt, H. (1987). Product formula algorithms for solving the time dependent Schrödinger equation. *Comp. Phys. Rep.* 7:1. doi: 10.1016/0167-7977(87)90002-5

De Raedt, H., Jin, F., Willsch, D., Willsch, M., Yoshioka, N., Ito, N., et al. (2019). Massively parallel quantum computer simulator, eleven years later. *Comput. Phys. Commun.* 237, 47–61. doi: 10.1016/j.cpc.2018.11.005

De Raedt, K., Michielsen, K., De Raedt, H., Trieu, B., Arnold, G., Richter, M., et al. (2007). Massively parallel quantum computer simulator. *Comput. Phys. Commun.* 176, 121–136. doi: 10.1016/j.cpc.2006.08.007

D-Wave Systems Inc (2023a). *Ocean SDK*. Available online at: https://github.com/dwavesystems/dwave-ocean-sdk (accessed July 17, 2023).

D-Wave Systems Inc (2023b). *Dwave-system Reference Documentation*. Available online at: https://docs.ocean.dwavesys.com/projects/system/en/latest/reference/generated/dwave.embedding.chain_strength.uniform_torque_compensation.html (accessed August 22, 2023).

Farhi, E., Goldstone, J., and Gutmann, S. (2002). Quantum adiabatic evolution algorithms with different paths. *arXiv preprint quant-ph/0208135*.

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

Finnila, A., Gomez, M., Sebenik, C., Stenson, C., and Doll, J. (1994). Quantum annealing: a new method for minimizing multidimensional functions. *Chem. Phys. Lett*. 219, 343–348. doi: 10.1016/0009-2614(94)00117-0

Garey, M. R., and Johnson, D. S. (1979). Computers and intractability: a guide to the theory of Np-completeness. *Siam Rev*. 24:90. doi: 10.1137/1024022

Gilbert, V., and Rodriguez, J. (2023). "Discussions about high-quality embedding on quantum annealers," in *Emerging Optimization Methods: From Metaheuristics to Quantum Approaches*.

Gonzalez Izquierdo, Z., Grabbe, S., Hadfield, S., Marshall, J., Wang, Z., and Rieffel, E. (2021). Ferromagnetically shifting the power of pausing. *Phys. Rev. Appl*. 15:044013. doi: 10.1103/PhysRevApplied.15.044013

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

Grant, E., Humble, T. S., and Stump, B. (2021). Benchmarking quantum annealing controls with portfolio optimization. *Phys. Rev. Appl*. 15:014012. doi: 10.1103/PhysRevApplied.15.014012

Harris, R., Johnson, M. W., Lanting, T., Berkley, A. J., Johansson, J., Bunyk, P., et al. (2010). Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor. *Phys. Rev. B* 82:024511. doi: 10.1103/PhysRevB.82.024511

Hegde, P. R., Passarelli, G., Cantele, G., and Lucignano, P. (2023). Deep learning optimal quantum annealing schedules for random Ising models. *New J. Phys*. 25:073013. doi: 10.1088/1367-2630/ace547

Hegde, P. R., Passarelli, G., Scocco, A., and Lucignano, P. (2022). Genetic optimization of quantum annealing. *Phys. Rev. A* 105:012612. doi: 10.1103/PhysRevA.105.012612

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

Huyghebaert, J., and De Raedt, H. (1990). Product formula methods for time-dependent Schrödinger problems. *J. Phys. A* 23, 5777–5793. doi: 10.1088/0305-4470/23/24/019

Ikeda, K., Nakamura, Y., and Humble, T. S. (2019). Application of quantum annealing to nurse scheduling problem. *Sci. Rep*. 9:12837. doi: 10.1038/s41598-019-49172-3

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

Jünger, M., Lobe, E., Mutzel, P., Reinelt, G., Rendl, F., Rinaldi, G., et al. (2021). Quantum annealing versus digital computing. *ACM J. Exper. Algor*. 26, 1–30. doi: 10.1145/3459606

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

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

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

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

Lobe, E., and Lutz, A. (2021). Minor embedding in broken Chimera and Pegasus graphs is NP-complete. *arXiv preprint arXiv:2110.08325*.

Lucas, A. (2014). Ising formulations of many NP problems. *Front. Phys*. 2:74887. doi: 10.3389/fphy.2014.00005

Lucas, A. (2019). Hard combinatorial problems and minor embeddings on lattice graphs. *Quant. Inf. Proc*. 18:702. doi: 10.1007/s11128-019-2323-5

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

McGeoch, C. C., and Farre, P. (2023). "Milestones on the quantum utility highway," in *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)* (IEEE), 393–399. doi: 10.1109/SEC54971.2022.00058

McGeoch, C. C., and Wang, C. (2013). "Experimental evaluation of an adiabatic quantum system for combinatorial optimization," in *Proceedings of the ACM International Conference on Computing Frontiers* (ACM). doi: 10.1145/2482767.2482797

McLeod, C. R., and Sasdelli, M. (2022). "Benchmarking D-Wave quantum annealers: spectral gap scaling of maximum cardinality matching problems," in *Computational Science-ICCS 2022*, eds. D. Groen, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot (Cham: Springer International Publishing), 150–163. doi: 10.1007/978-3-031-08760-8_13

Mehta, V., Jin, F., De Raedt, H., and Michielsen, K. (2021). Quantum annealing with trigger Hamiltonians: application to 2-satisfiability and nonstoquastic problems. *Phys. Rev. A* 104:032421. doi: 10.1103/PhysRevA.104.032421

Morita, S. (2007). Faster annealing schedules for quantum annealing. *J. Phys. Soc. Japan* 76:104001. doi: 10.1143/JPSJ.76.104001

Pelofske, E. (2023). Comparing three generations of D-Wave quantum annealers for minor embedded combinatorial optimization problems. *arXiv preprint arXiv:2301.03009*.

Pokharel, B., Izquierdo, Z. G., Lott, P. A., Strbac, E., Osiewalski, K., Papathanasiou, E., et al. (2021). Inter-generational comparison of quantum annealers in solving hard scheduling problems. *Quant. Inf. Proc*. 22:364. doi: 10.1007/s11128-023-04077-z

Raymond, J., Ndiaye, N., Rayaprolu, G., and King, A. D. (2020). "Improving performance of logical qubits by parameter tuning and topology compensation," in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 295–305. doi: 10.1109/QCE49297.2020.00044

Roberston, N., and Seymour, P. D. (1995). Graph minors. xiii. The disjoint paths problem. *J. Combin. Theory Series B* 63, 65–110. doi: 10.1006/jctb.1995.1006

Rønnow, T. F., Wang, Z., Job, J., Boixo, S., Isakov, S. V., Wecker, D., et al. (2014). Defining and detecting quantum speedup. *Science* 345, 420–424. doi: 10.1126/science.1252319

Ryan-Anderson, C., Brown, N. C., Allman, M. S., Arkin, B., Asa-Attuah, G., Baldwin, C., et al. (2022). Implementing fault-tolerant entangling gates on the five-qubit code and the color code. *arXiv preprint arXiv:2208.01863*.

Sasaki, G. H., and Hajek, B. (1988). The time complexity of maximum matching by simulated annealing. *J. ACM* 35, 387–403. doi: 10.1145/42282.46160

Shin, S. W., Smith, G., Smolin, J. A., and Vazirani, U. (2014). How "quantum" is the D-Wave machine? *arXiv preprint arXiv:1401.7087*.

Smolin, J. A., and Smith, G. (2013). Classical signature of quantum annealing. *Front. Phys*. 2:52. doi: 10.3389/fphy.2014.00052

Stollenwerk, T., Michaud, V., Lobe, E., Picard, M., Basermann, A., and Botter, T. (2021). Agile earth observation satellite scheduling with a quantum annealer. *IEEE Trans. Aerosp. Electr. Syst*. 57, 3520–3528. doi: 10.1109/TAES.2021.3088490

Susa, Y., and Nishimori, H. (2021). Variational optimization of the quantum annealing schedule for the Lechner-Hauke-Zoller scheme. *Phys. Rev. A* 103:022619. doi: 10.1103/PhysRevA.103.022619

Suzuki, M. (1976). Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. *Commun. Math. Phys*. 51, 83–190. doi: 10.1007/BF01609348

Suzuki, M. (1985). Decomposition formulas of exponential operators and Lie exponentials with some applications to quantum mechanics and statistical physics. *J. Math. Phys*. 26, 601–612. doi: 10.1063/1.526596

Takeda, K., Noiri, A., Nakajima, T., Kobayashi, T., and Tarucha, S. (2022). Quantum error correction with silicon spin qubits. *Nature* 608, 682–686. doi: 10.1038/s41586-022-04986-6

Trotter, H. F. (1959). On the product of semi-groups of operators. *Proc. Amer. Math. Soc*. 10, 545–551. doi: 10.1090/S0002-9939-1959-0108732-6

Venturelli, D., and Kondratyev, A. (2019). Reverse quantum annealing approach to portfolio optimization problems. *Quant. Mach. Intell*. 1, 17–30. doi: 10.1007/s42484-019-00001-w

Venuti, L. C., D'Alessandro, D., and Lidar, D. A. (2021). Optimal control for quantum optimization of closed and open systems. *Phys. Rev. Appl*. 16:054023. doi: 10.1103/PhysRevApplied.16.054023

Vert, D., Sirdey, R., and Louise, S. (2020). "Revisiting old combinatorial beasts in the quantum age: quantum annealing versus maximal matching," in *Computational Science-ICCS 2020*, eds. V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, et al. (Cham: Springer International Publishing), 473–487. doi: 10.1007/978-3-030-50433-5_37

Vert, D., Sirdey, R., and Louise, S. (2021). Benchmarking quantum annealing against "hard" instances of the bipartite matching problem. *SN Comput. Sci*. 2:106. doi: 10.1007/s42979-021-00483-1

Willsch, D., Willsch, M., Gonzalez Calaza, C. D., Jin, F., De Raedt, H., Svensson, M., et al. (2022a). Benchmarking Advantage and D-Wave 2000Q quantum annealers with exact cover problems. *Quant. Inf. Proc*. 21:141. doi: 10.1007/s11128-022-03476-y

Willsch, D., Willsch, M., Jin, F., Michielsen, K., and De Raedt, H. (2022b). GPU-accelerated simulations of quantum annealing and the quantum approximate optimization algorithm. *Comput. Phys. Commun*. 278:108411. doi: 10.1016/j.cpc.2022.108411

Yarkoni, S., Alekseyenko, A., Streif, M., Von Dollen, D., Neukart, F., and Bäck, T. (2021). "Multi-car paint shop optimization with quantum annealing," in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 35–41. doi: 10.1109/QCE52317.2021.00019

Zbinden, S., Bärtschi, A., Djidjev, H., and Eidenbenz, S. (2020). "Embedding algorithms for quantum annealers with Chimera and Pegasus

connection topologies," in *Lecture Notes in Computer Science* (Cham: Springer International Publishing), 187–206. doi: 10.1007/978-3-030-50743-5_10

Zeng, L., Zhang, J., and Sarovar, M. (2016). Schedule path optimization for adiabatic quantum computing and optimization. *J. Phys. A* 49:165305. doi: 10.1088/1751-8113/49/16/165305