



**HAL**  
open science

# Benchmarking Quantum Annealing Against “Hard” Instances of the Bipartite Matching Problem

Daniel Vert, Renaud Sirdey, Stéphane Louise

► **To cite this version:**

Daniel Vert, Renaud Sirdey, Stéphane Louise. Benchmarking Quantum Annealing Against “Hard” Instances of the Bipartite Matching Problem. SN Computer Science, 2021, 2 (2), pp.106. 10.1007/s42979-021-00483-1 . hal-04462577

**HAL Id: hal-04462577**

**<https://hal.science/hal-04462577>**

Submitted on 23 Apr 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License



# Benchmarking Quantum Annealing Against “Hard” Instances of the Bipartite Matching Problem

Daniel Vert<sup>1</sup> · Renaud Sirdey<sup>1</sup> · Stéphane Louise<sup>1</sup>

Received: 12 November 2020 / Accepted: 21 January 2021 / Published online: 20 February 2021  
© The Author(s) 2021

## Abstract

This paper experimentally investigates the behavior of analog quantum computers as commercialized by D-Wave when confronted to instances of the maximum cardinality matching problem which is specifically designed to be hard to solve by means of simulated annealing. We benchmark a D-Wave “Washington” (2X) with 1098 operational qubits on various sizes of such instances and observe that for all but the most trivially small of these it fails to obtain an optimal solution. Thus, our results suggest that quantum annealing, at least as implemented in a D-Wave device, falls in the same pitfalls as simulated annealing and hence provides additional evidences suggesting that there exist polynomial-time problems that such a machine cannot solve efficiently to optimality. Additionally, we investigate the extent to which the qubits interconnection topologies explains these latter experimental results. In particular, we provide evidences that the sparsity of these topologies which, as such, lead to QUBO problems of artificially inflated sizes can partly explain the aforementioned disappointing observations. Therefore, this paper hints that denser interconnection topologies are necessary to unleash the potential of the quantum annealing approach.

**Keywords** Quantum computing · Quantum annealing · Bipartite matching

## Introduction

From a practical view, the emergence of quantum computers able to compete with the performance of the most powerful conventional computers remains highly speculative in the foreseeable future. Indeed, although quantum computing devices are scaling up to the point of achieving the so-called milestone of quantum supremacy [23], these intermediate-scale devices, referred to as NISQ [22], will not be able to run mainstream quantum algorithms such as Grover, Shor

and their many variants at significant practical scales. Yet there are other breeds of machines in the quantum computing landscape, in particular, the so-called analog quantum computers for which there exists a family of actual processor series developed and sold by the Canadian company D-Wave as first concrete realizations of these kinds of quantum computers. These machines implement a noisy version of the Quantum Adiabatic Algorithm introduced by Farhi et al. in 2001 [13]. From an abstract point of view, such a machine may be seen as an oracle specialized in the resolution of an NP-hard optimization problem<sup>1</sup> (of the spin-glass type) with an algorithm that can be compared to a quantum version of the usual simulated annealing, and hence can display quantum speed-up in some cases at least. In this context, as it is considered unlikely that any presently known quantum computing paradigm will lead to efficient algorithms for solving NP-hard problems, determining whether or not quantum adiabatic computing yields an advantage over classical computing is most likely an ill-posed question given

---

This article is part of the topical collection “Quantum Computing: Circuits Systems Automation and Applications” guest edited by Himanshu Thapliyal and Travis S. Humble.

---

✉ Stéphane Louise  
stephane.louise@cea.fr

Daniel Vert  
daniel.vert2@cea.fr

Renaud Sirdey  
renaud.sirdey@cea.fr

<sup>1</sup> CEA, LIST, Université Paris-Saclay, 91129 Palaiseau, France

<sup>1</sup> Strictly speaking, to the best of the authors’ knowledge, although the general problem is NP-hard, the complexity status of the more specialized instances constrained by the qubit interconnection topology of these machines remains open.

present knowledge. Yet, as a quantum analogue of simulated annealing, attempting to demonstrate a quantum advantage of adiabatic algorithms over simulated annealing appears to be a better-posed question. At the time of writing, this problem is the focus of a lot of works which, despite claims of exponential speedups in specific cases [12] (which also lead to the development of the promising Simulated Quantum Annealing classical metaheuristic [10]), hint towards a logarithmic decay requirement of the temperature-analog of QA but with smaller constants involved [25] leading to only an  $O(1)$  advantage of QA over SA in the general case. Such an advantage has furthermore recently been experimentally demonstrated by Albash and Lidar [1]. The present paper contributes to the study of the QA vs SA issue by experimentally confronting a D-Wave quantum annealer to the pathological instances of the maximum cardinality matching problem proposed in the late 80's [26] to show that simulated annealing was indeed unable to solve certain polynomial problems in polynomial time. Demonstrating an ability to solve these instances to optimality on a quantum annealer would certainly hint towards a worst-case quantum annealing advantage over simulated annealing whereas failure to do so would tend to demonstrate that quantum annealing remains subject to the same pitfalls as simulated annealing and is, therefore, unable to solve certain polynomial problems efficiently. To do so, the present paper experimentally benchmarks a D-Wave “Washington” (2X) with 1098 operational qubits on various sizes of such pathologic instances of the maximum cardinality matching problem and observes that for all but the most trivially small of these it fails to obtain an optimal solution. This thus provides negative evidences towards the existence of a *worst-case* advantage of quantum annealing over classical annealing. As a by-product, our study also provides feedback on using a D-Wave annealer in particular with respect to the size of problems that can be mapped on such a device due to the various constraints of the system. In addition, we investigate to what extent the qubits interconnection topology influences these results. To do so, we study how *simulated annealing* is able to solve our hard instances of the bipartite matching problem when they are embedded in the so-called Chimera and Pegasus topologies [11] used in present-day D-Wave machines. These results show that when solved without taking any topology into account (or, equivalently, when assuming a fully connected network of qubits), the instance sizes we are able to map on a D-Wave remains solvable by simulated annealing. In other words, the regime in which those instances becomes (asymptotically) hard for simulated annealing is not yet accessible to these machines, due to the need for one-to-many variables to qubits assignments. Then, when simulated annealing is used to solve these (artificially) larger QUBO instances resulting from mapping the original problems onto the qubits interconnection topology, it performs no better than a quantum annealer. This,

therefore, hints that the constraints imposed by presently used qubits interconnects tend to counter-productively obfuscate the optimization problem enough to prevent both classical and quantum annealing from performing well. This paper is organized as follows. After a brief reminder of lessons learnt from simulated annealing history (“[Lessons from simulated annealing history](#)” section), “[Quantum annealing and its D-wave implementation](#)” section provides some background on quantum annealing, the D-Wave devices and their limitations. “[Solving maximum cardinality matching on a quantum annealer](#)” section surveys the maximum cardinality matching problem, introduces the  $G_n$  graph family underlying our pathologic instances and subsequently details how we build the QUBO instances to be mapped on the D-Wave from those instances. Then, “[Experimental results](#)” section extensively details our experimental setup and experimentations and “[Discussion and perspectives](#)” section concludes the paper with a discussion of the results and a number of perspectives to follow up on this work.

## Lessons from Simulated Annealing History

On top of the formal analogies between simulated and quantum annealing, there also appears to be an analogy between the latter present state of art and that of simulated annealing when it was first introduced. So it might be useful to recall a few facts on SA. Indeed, simulated annealing was introduced in the mid-80's [9, 19] and its countless practical successes quickly established it as a mainstream method for approximately solving computationally-hard combinatorial optimization problems. Thus, the theoretical computer science community investigated in great depth its convergence properties in an attempt to understand the worst-case behavior of the method. With that respect, these pieces of work, which were performed in the late 80's and early 90's, lead to the following insights. First, when it comes to solving combinatorial optimization problems to optimality, it is necessary (and sufficient) to use a logarithmic cooling schedule [15, 16, 21] leading to an exponential-time convergence in the worst-case (an unsurprising fact since it is known that  $P \neq NP$  in the oracle setting [3]). Second, particular instances of combinatorial problems have been designed to specifically require an exponential number of iterations to reach an optimal solution for example on the (NP-hard) 3-coloring problem [21] and, more importantly for this paper, on the (polynomial) maximum cardinality matching problem [26]. Lastly, another line of works, still active today, investigated the asymptotic behavior of hard combinatorial problems [8, 18, 27] showing that the cost ratio between best and worst-cost solutions to random instances tends (quite quickly) to 1 as the instance size tends to infinity. These latter results provided clues as to why simple heuristics such

as simulated annealing appear to work quite well on large instances as well as to why branch-and-bound type exact resolution methods tend to suffer from a trailing effect (i.e. find optimal or near-optimal solutions relatively quickly but fail to prove their optimality in reasonable time). Despite these results now being quite well established, they can also, as illustrated in this paper, contribute to the ongoing effort to better understand and benchmark quantum adiabatic algorithms [13] and especially the machines that now implements it to determine whether or not they provide a quantum advantage over some classes of classical computations.

## Quantum Annealing and Its D-wave Implementation

### The Generalized Ising Problem and QUBO

D-Wave systems are based on a quantum annealing process<sup>2</sup> whose goal is to minimize the Generalized Ising Hamiltonian from Eq. (1):

$$\mathcal{H}(\mathbf{h}, \mathbf{J}, \boldsymbol{\sigma}) = \sum_i h_i \sigma_i + \sum_{i < j} J_{ij} \sigma_i \sigma_j, \tag{1}$$

where the external field  $\mathbf{h}$  and spin coupling interactions matrix  $\mathbf{J}$  are given, and the vector of spin (or qubit) values  $\boldsymbol{\sigma} / \forall i, \sigma_i \in \{-1, 1\}$  is the variable for which the energy of the system is minimized. Historically speaking, the Ising Hamiltonian corresponds to the case where only the closest neighbouring spins are allowed to interact (i.e.  $J_{ij} \neq 0 \iff$  nodes  $i$  and  $j$  are conterminous). The generalized Ising problem, for which any pair of spins in the system are allowed to interact, is easily transformed into a well known 0–1 optimization problem called QUBO (for Quadratic Unconstrained Binary Optimization) which objective function is given by:

$$O(\mathbf{Q}, \mathbf{x}) = \sum_i Q_{ii} x_i + \sum_{i < j} Q_{ij} x_i x_j, \tag{2}$$

in which the matrix  $\mathbf{Q}$  is constant and the goal of the optimization is to find the vector of binary variables  $\forall i, x_i \in \{0, 1\}$  that either minimizes or maximizes the objective function  $O(\mathbf{Q}, \mathbf{x})$  from Eq. 2. For the minimization problem (but only up to a change of sign for the maximization problem), it is trivial that the generalized Ising problem and the QUBO problem are equivalent given  $\forall i, Q_{ii} = h_i, \forall i, j / i \neq j, Q_{ij} = J_{ij}$  and  $\forall i, \sigma_i = 2x_i - 1$ .

Hence, if quantum annealing can reach a configuration of minimum energy, then the associated state vector solves the equivalent QUBO problem at the same time. As the behavior of each qubit in a quantum annealer allows them to be in a superposition state (a combination of the states “−1” and “+1”) until they relax to either one of these eigen-states, it is thought that quantum mechanical phenomena—e.g., quantum tunneling—can help reaching the minimum energy configuration, or at least a close approximation of it, in more cases than with Simulated Annealing (SA). Indeed, when SA only relies on (simulated) temperatures to pass over barriers of potential, in Quantum Annealing, quantum phenomena can help because tunneling is more efficient to pass energy barriers even in the case where the temperature is low. Therefore, this technique is a promising heuristic approach to “quickly” find acceptable solutions for certain classes of complex NP-Hard problems that are easily mapped to these machines, such as optimization, machine learning, or operational research problems.

The physical principle upon which the computation process of D-Wave machines [17] occurs is given by a time-dependent Hamiltonian as given in Eq. (3).

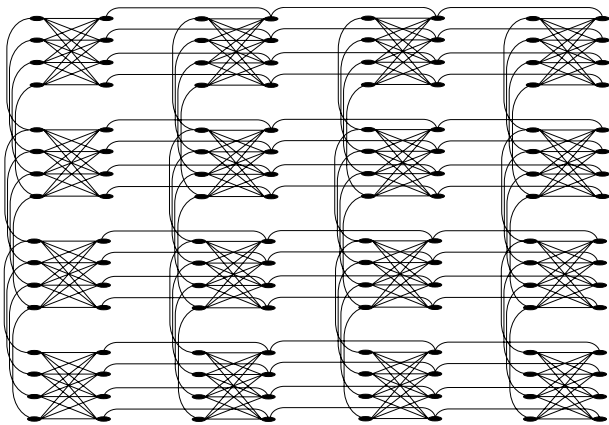
$$\mathcal{H}(t) = A(t)\mathcal{H}_0 + B(t)\mathcal{H}_p \tag{3}$$

The functions  $A(t)$  and  $B(t)$  must satisfy  $B(t = 0) = 0$  and  $A(t = \tau) = 0$  so that, when the state evolution  $t = 0$  changes to  $t = \tau$ , the Hamiltonian  $H(t)$  is the quantum annealing process that lead to the final form of the Hamiltonian which is the objective Ising problem that requires to be minimized. Thus, the fundamental state  $\mathcal{H}(0) = \mathcal{H}_0$  evolves to a state  $\mathcal{H}(\tau) = \mathcal{H}_p$ , the measurements made at time  $\tau$  give us low energy states of the Ising Hamiltonian (Eq. 1). The adiabatic theorem states that if the time evolution is slow enough (i.e.  $\tau$  is large enough), and supposing the coherence domain is large enough then the optimal (global) solution  $\epsilon(\boldsymbol{\sigma})$  of the system can be obtained with a high probability. By using

- $\mathcal{H}_0 = \sum_i \sigma_i^x$  gives the quantum effects,
- $\mathcal{H}_p = \sum_i h_i \sigma_i^z + \sum_{(ij)} J_{ij} \sigma_i^z \sigma_j^z$  is given to encode the problem of the Ising instance in the final state.

As the process of adiabatic annealing transitions the system from a constant coupling with a superposition of spins because the initial Hamiltonian is based on Eigen-vectors of operator  $\hat{\sigma}^x$  (on the  $x$ -axis) whilst the momentum of spin on  $\mathcal{H}_p$  is an Eigen-state of  $\hat{\sigma}^z$  (on the  $z$ -axis) for which Eigen-states of  $\hat{\sigma}^x$  are superposition states, the adiabatic theorem allows transitioning from the initial ferromagnetic state on axis  $x$  to an eigen-state of the Hamiltonian of Eq. 1 on axis  $z$  and hopefully to the lowest energy of it.

<sup>2</sup> A combinatorial optimization technique functionally similar to conventional (simulated) annealing but which, instead of applying thermal fluctuations, uses quantum phenomena to search the solution space more efficiently [14].



**Fig. 1** Representation of a Chimera graph with  $4 \times 4$  unit cells, each a small  $2 \times 4$  bipartite graph, for 128 physical qubits. The links represent all the inter-spin coupling  $J_{ij}$  that can be different from 0

$$\min \epsilon(\sigma) = \min \left\{ \sum_i h_i \sigma_i + \sum_{i,j} J_{ij} \sigma_i \sigma_j \right\}. \quad (4)$$

### D-wave Limitations

Nonetheless, it is worth noting, that in the case of the current architectures of the D-Wave annealing devices, the freedom to choose the  $J_{ij}$  coupling constants is severely restrained by the hardware qubit-interconnection topology. In particular, this so-called *Chimera* topology is sparse, with a number of inter-spin couplings limited to a maximum of 6 per qubit (or spin variable). Figure 1 illustrates an instance of the Chimera graph for 128 qubits,  $T = (N_T, E_T)$ , where nodes  $N_T$  are qubits and represent problem variables with programmable weights ( $h_i$ ), and edges  $E_T$  are associated to the couplings  $J_{ij}$  between qubits ( $J_{ij} \neq 0 \implies (i, j) \in E_T$ ). As such, if the graph induced by the nonzero couplings is not isomorphic to the Chimera graph, which is the case most usually, then one must resort to several palliatives among which the duplication of logical qubits onto several physical qubits is the least disruptive one if the corresponding expanded problem can still fit on the target device.

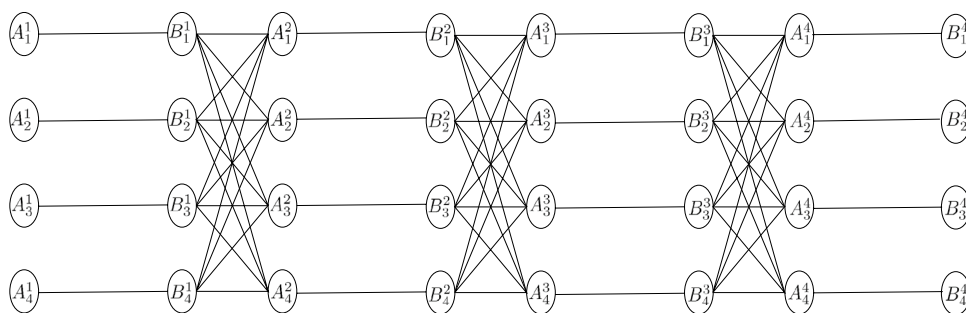
Then, a D-Wave annealer minimizes the energy from the Hamiltonian of Eq. (1) by associating weights ( $h_i$ ) with qubit spins ( $\sigma_i$ ) and couplings ( $J_{ij}$ ) with couplers between the spins of the two connected qubits ( $\sigma_i$  and  $\sigma_j$ ). As an example, the D-Wave 2X system we used has 1098 operational qubits and 3049 operational couplers. As said previously, a number of constraints have an impact on the practical efficiency of this type of machines. In [5], the authors highlight four factors: the precision/control error which is limited by the

parameters  $\mathbf{h}$  and  $\mathbf{J}$  which value ranges are also limited<sup>3</sup>, the low connectivity<sup>4</sup> in  $T$ , and the in fine small number of useful qubits once the topological constraints are accounted for. In [4], the authors show that using large energy gaps in the Ising representation of the model one wants to optimize can greatly mitigate some of the intrinsic limitations of the hardware like precision of the coupling values and noises in the spin measurements. They also suggest using ferromagnetic Ising coupling between qubits (i.e., making qubit duplication) to mitigate the issues with the limited connectivity of the Chimera graph. All these suggestions can be considered good practices (which we did our best to follow) when trying to use the D-Wave machine to solve real Ising or QUBO problems with higher probabilities of outputting the best solution despite hardware and architecture limitations. A last point to take into consideration is that real qubits may be biased due to hardware defects, and this also should be taken into consideration when conducting a series of computing jobs on the D-Wave computers. As described in Sect. 5, the state-of-the-art recommendation is simply to change several times the target Ising problem by randomly choosing 10% of the variables and make a variable transformation from  $x$  to  $y = 1 - x$ . As this is not yet automatically done in the D-Wave tools, this is part of the pre-processing of one problem resolution onto a D-Wave computer.

Thus, pre-processing algorithms are required to adapt the graph of a problem to the hardware. Pure quantum approaches are limited by the number of variables (duplication included) that can be mapped on the hardware. Larger graphs require the development of hybrid approaches (both classical and quantum) or the reformulation of the problem to adapt to the architecture. For example, for a  $128 \times 128$  matrix size, the number of possible coefficients  $J_{ij}$  is 8128 in the worst-case, while the Chimera graph which associates 128 qubits ( $4 \times 4$  unit cells) has only 318 couplers. The topology, therefore, accounts only for  $\sim 4\%$  of the total number of couplings required to map a  $128 \times 128$  matrix in the worst case. Although preliminary studies (e.g., [28]) have shown that it is possible to obtain solutions close to known minimums for  $\mathbf{Q}$  matrices with densities higher than those permitted by the Chimera graph by eliminating some coefficients, they have also shown that doing so isomorphically to the Chimera topology is difficult. It follows that solving large and dense QUBO instances requires nontrivial pre and postprocessing as well as a possibly large number of invocations of the quantum annealer.

<sup>3</sup> The range of  $h_i \in [-2, +2]$  and  $J_{i,j} \in [-1, +1]$  is a limitation for all values of the variables to be included in the graph. If the values of  $h_i$  and  $J_{i,j}$  are outside their respective ranges, then they are unavailable and not mapped

<sup>4</sup> If the problems to be solved do not match the structure of the  $T$  graph architecture, then they cannot be mapped and resolved directly.



**Fig. 2**  $G_3$  is a quite simple instance of the maximum cardinality matching problem. While it is not a natural QUBO problem, it is transformable into a QUBO problem by introducing additional

weights so that invalid solutions would not be optimal. Here the optimal solution is easy: select all the edges in the sparse areas of the  $G_n$  graph

Additionally, the next generation of systems that D-Wave is starting to release at the time of writing reaches above 5000 qubits interconnected by the so-called *Pegasus* topology [7, 11]. The *Pegasus* topology admits the *Chimera* topology as a subgraph but reaches up to a maximum degree of 15 to be compared to the low degree 6 maximum of the *Chimera* one. Although *Pegasus*-based machines are commercialized just now, the D-Wave software toolchain already supports this new interconnect which allows to perform preliminary experiments at least in terms of problem mapping (as we do in Sect. 5.2).

### Solving Maximum Cardinality Matching on a Quantum Annealer

#### Maximum Cardinality Matching and the $G_n$ Graph Family

Given an (undirected) graph  $G = (V, E)$ , the maximum matching problem asks for  $M \subseteq E$  such that  $\forall e, e' \in M, e \neq e'$  we have that  $e \cap e' = \emptyset$  and such that  $|M|$  is maximum. The maximum matching problem is a well-known polynomial problem dealt with in almost every textbook on combinatorial optimization (e.g., [20]), yet the algorithm for solving it in general graphs, Edmond’s algorithm, is a non-trivial masterpiece of algorithmics. Additionally, when  $G$  is bipartite i.e. when there exists two collectively exhaustive and mutually exclusive subsets of  $E, A$  and  $B$ , such that no edge has both its vertices in  $A$  or in  $B$ , the problem becomes a special case of the maximum flow problem and can be dealt with several simpler algorithms [20].

It is, therefore, very interesting that such a seemingly powerful method as simulated annealing can be deceived by special instances of this latter easier problem. Indeed, in a landmark 1988 paper [26], Sasaki and Hajek, have considered the following family of special instances of the

bipartite matching problem. Let  $G_n$  denote the (undirected) graph with vertices  $\bigcup_{i=0}^n A^{(i)} \cup \bigcup_{i=0}^n B^{(i)}$  where each of the  $A^{(i)}$ ’s and  $B^{(i)}$ ’s have cardinality  $n + 1$  (vertex numbering goes from 0 to  $n$ ), where vertex  $A_j^{(i)}$  is connected to vertex  $B_j^{(i)}$  and where vertex  $B_j^{(i)}$  is connected to all vertices in  $A^{(i+1)}$  (for  $i \in \{0, \dots, n\}$  and  $j \in \{0, \dots, n\}$ ). These graphs are clearly bipartite has neither two vertices in  $\bigcup_{i=0}^n A^{(i)}$  nor two vertices in  $\bigcup_{i=0}^n B^{(i)}$  are connected. These graphs therefore exhibit a sequential structure which alternates between sparsely and densely connected subsets of vertices, as illustrated on Fig. 2 for  $G_3$ .

As a special case of the bipartite matching problem, the maximum cardinality matching over  $G_n$  can be solved by any algorithm solving the former. Yet, it is even easier as one can easily convince oneself that a maximum matching in  $G_n$  is obtained by simply selecting all the edges connecting vertices in  $A^{(i)}$  to vertices in  $B^{(i)}$  (for  $i \in \{0, \dots, n\}$ ), i.e. all the edges in the sparsely connected subsets of vertices, and that is the only way to do so. This, therefore, leads to a maximum matching of cardinality  $(n + 1)^2$ .

Hence, we have a straightforward special case of a polynomial problem, yet the seminal result of Sasaki and Hajek states that the mathematical expectation of the number of iterations required by a large class of (classical) annealing-type algorithms to reach a maximum matching on  $G_n$  is in  $O(\exp(n))$ . The  $G_n$  family therefore provides an interesting playground to study how quantum annealing behaves on problems that are hard for simulated annealing. This is what we do, experimentally, in the following.

#### QUBO Instances

In order for our results to be fully reproducible we hereafter describe how we converted instances of the maximum matching problem into instances of the Quadratic Unconstrained Boolean Optimization (QUBO) problem which

D-Wave machines require as input. Let  $G = (V, E)$  denote the (undirected) graph for which a maximum matching is desired. We denote  $x_e \in \{0, 1\}$ , for  $e \in E$ , the variable which indicates whether  $e$  is in the matching. Hence we have to maximize  $\sum_{e \in E} x_e$  subject to the constraints that each vertex  $v$  is covered at most once, i.e.  $\forall v \in V$ ,

$$\sum_{e \in \Gamma(v)} x_e \leq 1, \tag{5}$$

where  $\Gamma(v)$ , in standard graph theory notations, denotes the set of edges which have  $v$  as an endpoint. In order to turn this into a QUBO problem we have to move the above constraints into the economic function, for example in maximizing,

$$\sum_{e \in E} x_e - \lambda \sum_{v \in V} \left( 1 - \sum_{e \in \Gamma(v)} x_e \right)^2,$$

which, after rearrangements, leads to the following economic function,

$$\sum_{e \in E} x_e + \sum_{v \in V} \sum_{e \in \Gamma(v)} 2\lambda x_e - \sum_{v \in V} \sum_{e \in \Gamma(v)} \sum_{e' \in \Gamma(v)} \lambda x_e x_{e'}$$

Yet we have to reorganize a little to build a proper QUBO matrix. Let  $e = (v, w)$ , variable  $x_e$  has coefficient 1 in the first term,  $2\lambda$  in the second term (for  $v$ ) then  $2\lambda$  again in the second term (for  $w$ ) then  $-\lambda$  in the third term (for  $v$  and  $e' = e$ ) and another  $-\lambda$  again in the third term (for  $w$  and  $e' = e$ ). Hence, the diagonal terms of the QUBO matrix are,

$$Q_{ee} = 1 + 4\lambda - 2\lambda = 1 + 2\lambda.$$

Then, if two distinct edges  $e$  and  $e'$  share a common vertex, the product of variables  $x_e x_{e'}$  has coefficient  $-\lambda$ , in the third term, when  $v$  corresponds to the vertex shared by the two edges, and this is so twice. So, for  $e \neq e'$ ,

$$Q_{ee'} = \begin{cases} -2\lambda & \text{if } e \cap e' \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

Taking  $\lambda = \frac{E}{5}$ , for example for  $G_1$ , we thus obtain an 8 variables QUBO (the corresponding matrix is given in [29]) for which a maximum matching has cost 68, the second-best solutions has cost 53 and the worst one (which consist in selecting all edges) has cost -56.

<sup>5</sup> As,  $E/5$  is clearly an upper bound for the cost of any matching, any solution which violates at least one of the constraints (5) cannot be optimal.

**Table 1** Number of qubits required to handle the QUBO instances associated to  $G_1, G_2, G_3$  and  $G_4$ . See text

	#var.	#qubits	Average dup.	Max. dup.
$G_1$	8	16	2.0	6
$G_2$	27	100	3.7	6
$G_3$	64	431	6.7	18
$G_4$	125	951	7.6	18

## Experimental Results

### Concrete Implementation on a D-Wave

In this section, we detail the steps that we have followed to concretely map and solve the QUBO instances associated to  $G_n, n \in \{1, 2, 3, 4\}$ , on a DW2X operated by the University of South California. Unfortunately (yet unsurprisingly), the QUBO matrices defined in the previous section are not directly mappable on the Chimera interconnection topology and, thus, we need to resort to qubit duplication i.e., use *several* physical qubits to represent *one* problem variable (or “logical qubit”). Fortunately, the D-Wave software pipeline automates this duplication process. Yet, this need for duplication (or equivalently the sparsity of the Chimera interconnection topology) severely limits the size of the instances we were able to map on the device and we had to stop at  $G_4$  which 125 variables required using 951 of the 1098 available qubits. Table 1 provides the number of qubits required for each of our four instances. For  $G_1, G_2$  the maximum duplication is 6 qubits and for  $G_3, G_4$  it is 18 qubits.

Eventually, qubit duplication leads to an expanded QUBO with more variables and an economic function which includes an additional set of penalty constraints to favor solutions in which qubits representing the same variable indeed end up with the same value. More precisely, each pair of distinct qubits  $q$  and  $q'$  (associated to the same QUBO variable) adds a penalty term of the form  $\varphi q(1 - q')$ . Where the penalty constant  $\varphi$  is (user) chosen as minus the cost of the worst possible solution to the initial QUBO which is obtained for a vector filled with ones (i.e., a solution that selects all edges of the graph and which therefore maximizes the highly-penalized violations of the cardinality constraints). This, therefore, guarantees that a solution which violates at least one of these consistency constraints cannot be optimal (please note that we have switched from a maximization problem in Sect. 4.2 to a minimization problem as required by the machine). Lastly, as qubit duplication leads to an expanded QUBO which support graph is trivially isomorphic to the Chimera topology, it can be mapped on the device after a renormalization of its coefficients to ensure that the diagonal terms of  $Q$  are in  $[-2, 2]$  and the others in  $[-1, 1]$ .

**Table 2** Experimental results summary without (top) and with (bottom) majority voting to fix qubit duplication issues on  $G_1, G_2, G_3, G_4$ . See text

	Opt.	Best	Worst	Mean	Median	Stdev
$G_1$	-68	-68	-9	-66.8	-68	4.6
$G_2$	-495	-495	-29	-398.2	-388	48.1
$G_3$	-2064	-1810	-505	-1454.8	-1548	157.7
$G_4$	-6275	-5527	-2507	-4609.9	-4675	346.5
$G_1$	-68	-68	-37	-66.8	-68	4.2
$G_2$	-495	-495	-277	-400.4	-388	44.6
$G_3$	-2064	-1810	-911	-1496.5	-1550	111.8
$G_4$	-6275	-5527	-3030	-4579.2	-4527	314.1

## Results Summary

This section reports on the experiments we have been able to perform on instances of the previous QUBO problems. As already emphasized, due to the sparsity of the qubit interconnection topology, our QUBO instances were not directly mappable on the D-Wave machine and we had to resort to qubit duplications (whereby one problem variable is represented by several qubits on the D-Wave, bound together to end up with the same value at the end of the annealing process). This need for qubit duplication limited us to  $G_4$  which, with 125 binary variables, already leads to a combinatorial problem of non trivial size. Yet, to solve it, we had to mobilize about 87% of the 1098 qubits of the machine. The results below have been obtained by running 10,000 times the quantum annealer with a 20  $\mu$ s annealing time (although we also experimented with 200 and 2000  $\mu$ s, which did not appear to affect the results significantly).

Additionally, to improve the quality of the results obtained in our experiments, we used different gauges (spin-reversal transformations). The principle of a gauge is to apply a Boolean inversion transformation to operators  $\sigma_i$  in our Hamiltonian (in QUBO terms, after qubit duplication, this just means replacing some variable  $x_i$  by  $1 - y_i$ , with  $y_i = 1 - x_i$  and updating the final QUBO matrix accordingly). This transformation has the particularity of not changing the optimal solution of the problem and of limiting the effect of local biases of the qubits, as well as machine accuracy errors [6]. Following common practices (e.g., [2]), we randomly selected 10% of the physical qubits used as gauges for each  $G_n$  instance that we mapped to the D-Wave. This preprocessing does indeed improve the results obtained, but not widely so: for example, on  $G_4$ , it leads to a 2.5% improvement on the mean solution cost outputted by the D-Wave and only a 1.2% improvement on the mean solution cost after correction of the duplication inconsistencies by majority voting. Our overall results are given in Table 2.

## Instances Solutions

$G_1$ . This instance leads to a graph with 8 vertices, 8 edges and then (before duplication) to a QUBO with 8 variables and 12 nonzero nondiagonal coefficients<sup>6</sup>; 16 qubits are then finally required. Over 10,000 runs, the optimal solution (with a cost of -68) was obtained 9265 times (with correction 9284 times). Interestingly, the worst solution obtained (with a cost of -9) violates duplication consistency as all the 6 qubits representing variable 6 do not have the same value (4 of them are 0, so in that particular case, rounding the solution by means of majority voting gives the optimal solution).

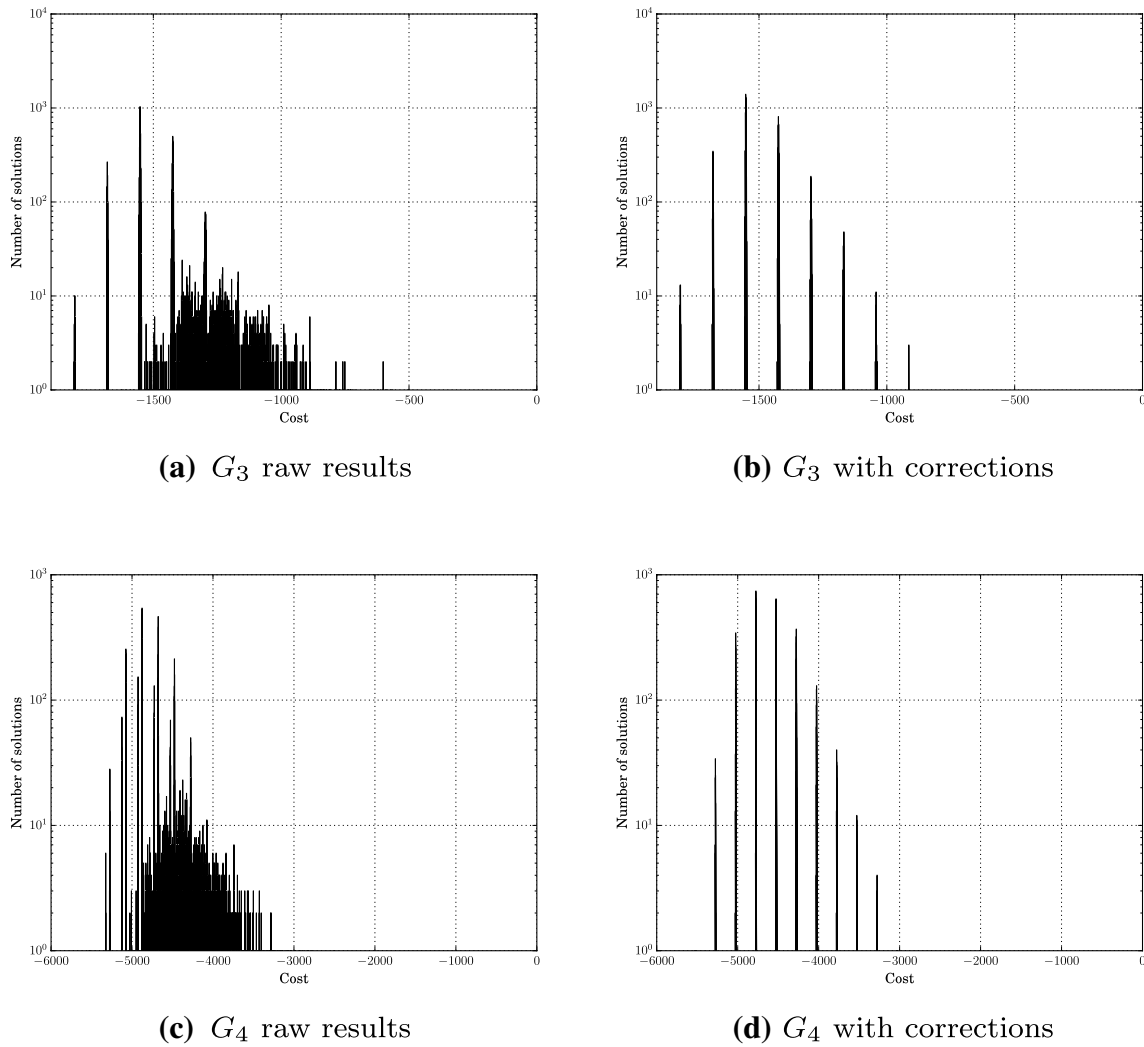
$G_2$ . This instance leads to a graph with 18 vertices, 27 edges and then to a QUBO with 27 variables and 72 nonzero nondiagonal coefficients. Overall, 100 qubits are required. Over 10,000 runs the optimal solution (with cost -495) was obtained only 510 times (i.e., a 6% hitting probability). Although the best solution obtained is optimal, the median solution (with cost -388) does not lead to a valid matching since four vertices are covered 3 times<sup>7</sup>. As for  $G_1$ , we also observe that the worst solution (with cost -277) has duplication consistency issues. Fixing these issues by means of majority voting results only in a marginal left shift of the average solution cost from -398.2 to -400.4, the median being unchanged.

$G_3$ . This instance leads to a graph with 32 vertices, 64 edges and then to a QUBO with 64 variables and 240 nonzero nondiagonal coefficients. Post-duplication, 431 qubits were required (39% of the machine capacity). Over 10,000 runs the optimal solution was never obtained. For  $G_3$ , the optimum value is -2064, thus the best solution obtained (with cost -1810) is around 15% far-off (the median cost of -1548 is 25% far-off). Furthermore,

<sup>6</sup> In the Chimera topology the diagonal coefficient are not constraining as there is no limitation on the qubits autocouplings.

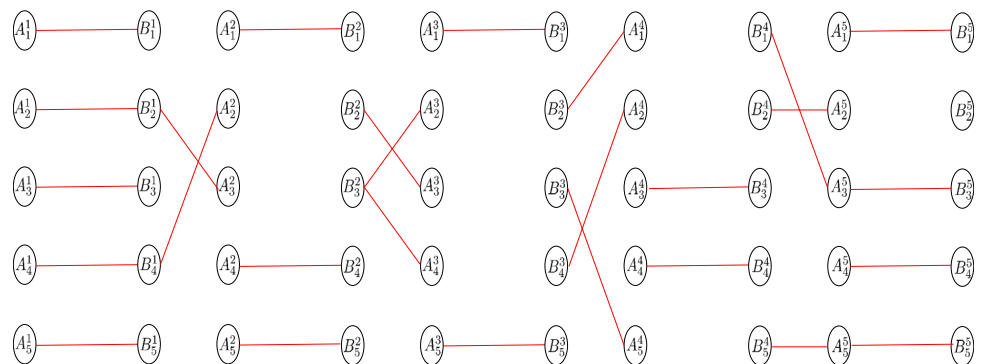
<sup>7</sup> Fixing this would require a postprocessing step to produce valid matchings. Of course, this is of no relevance for a polynomial problem, but such a postprocessing would thus be required when operationally using a D-Wave for solving non artificial problems.





**Fig. 3** Histograms on the left represent the economic function over 10000 annealing runs on  $G_3$  and  $G_4$ . Histograms on the right represent the economic function over 10,000 annealing runs on  $G_3$  and  $G_4$  (with duplication inconsistencies fixed by majority voting).

**Fig. 4** Graphic representation of the best solution obtained for  $G_4$ . See text



neither the best nor the median solution lead to valid matchings since in both, some vertices are covered several times. We also observe that the worst solution has

duplication consistency issues. Figure 3a shows the (renormalized) histogram of the economic function as outputted by the D-Wave for the 10,000 annealing runs we

**Table 3** Experimental results obtained when solving the raw QUBO instances for  $G_1$ ,  $G_2$ ,  $G_3$  and  $G_4$  by means of simulated annealing for several numbers of iterations per plateau of temperature

		Opt.	Best	Worst	Mean	Median	Stdev
$G_1$	$n$	-68	-68	-68	-68	-68	0
	$n^{1.5}$	-68	-68	-68	-68	-68	0
	$n^2$	-68	-68	-68	-68	-68	0
$G_2$	$n$	-495	-495	-495	-495	-495	0
	$n^{1.5}$	-495	-495	-495	-495	-495	0
	$n^2$	-495	-495	-495	-495	-495	0
$G_3$	$n$	-2064	-2064	-1810	-2004.7	-2064	79.9
	$n^{1.5}$	-2064	-2064	-2064	-2064	-2064	0
	$n^2$	-2064	-2064	-2064	-2064	-2064	0
$G_4$	$n$	-6275	-6275	-5528	-5785.3	-5777	178.9
	$n^{1.5}$	-6275	-6275	-6026	-6241.8	-6275	86.1
	$n^2$	-6275	-6275	-6275	-6275	-6275	0

performed. Additionally, since some of these solutions are inconsistent with respect to duplication, Fig. 3b shows the histogram of the economic function for the solutions in which duplication inconsistencies were fixed by majority voting (thus left shifting the average cost from  $-1454.8$  to  $-1496.5$  and the median cost from  $-1548$  to  $-1550$  which is marginal).

$G_4$ . This instance leads to a graph with 50 vertices, 125 edges and then to a QUBO with 125 variables and 600 nonzero non-diagonal coefficients. Post-duplication, 951 qubits were required (i.e., 87% of the machine capacity). Over 10,000 runs the optimal solution was never obtained. Still, Fig. 4 provides a graphic representation of the best solutions obtained, with cost  $-5527$  (median and worst solutions obtained respectively had costs  $-4675$  and  $-2507$ ). For  $G_4$ , the optimum value is  $-6075$ , thus the best solution obtained is around 10% far-off (a better ratio than for  $G_3$ ) and median cost 25%. Furthermore, neither the best nor the median solution lead to valid matches since in both, some vertices are covered several times. We also observe that the worst solution (as well as many others) has duplication consistency issues. Figure 3c shows the (renormalized) histogram of the economic function as outputted by the D-Wave for the 10000 annealing runs we performed. Additionally, since some of these solutions are inconsistent with respect to duplication, Fig. 3d shows the histogram of the economic function for the solutions in which duplication inconsistencies were fixed by majority voting (resulting, in this case, in a slight right shift of the average solution cost from  $-4609.9$  to  $-4579.2$  and of the median cost from  $-4675$  to  $-4527$  which is also marginal<sup>8</sup>).

<sup>8</sup> This slight right shift may seem counter-intuitive but the (highly nonlinear) majority voting operator does not provide any guarantee with respect to objective function improvement. It just guarantees that duplication issues are fixed, which is necessary (yet not sufficient) for a solution to be feasible.

## Resolution by Simulated Annealing

Simulated annealing was introduced in the mid-80's [9, 19] and its countless practical successes quickly established it as a mainstream method for approximately solving computationally-hard combinatorial optimization problems. As simulated annealing has been around for so long, there is no need to introduce the general method but rather to specify the key free parameter choices. 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 a constant temperature which we set to  $n$ ,  $n^{1.5}$  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.

Table 3 presents the results obtained when solving the raw QUBO for  $G_1$ ,  $G_2$ ,  $G_3$  and  $G_4$  with simulated annealing and several numbers of iterations for the Metropolis algorithm (over just 30 runs). Needless to emphasize that simulated annealing performs extremely well compared to the D-Wave with the worst solution of the former (yet over 30 runs) almost always beating the best solutions obtained by the D-Wave over 10,000 runs. Also, the fact that simulated annealing (even with only  $n$  iterations per plateau) finds the optimal solution with high probability, suggests that the instance size up to  $G_4$  are too small to reach the (asymptotic) exponential number of iterations regime of Sasaki & Hajek theorem i.e., these instances are small enough to remain relatively easy for classical annealing (although we

**Table 4** Experimental results obtained when solving the expanded QUBO instances for  $G_4$  on both the Chimera (top rows) and Pegasus (bottom rows) topologies by means of simulated annealing (30 runs)

		Opt.	Best	Worst	Mean	Median
$G_4$	$n$	-6275	-2213	3662	1453.9	1401.0
(Chim.)	$n^{1.5}$	-6275	-4526	-2654	-3585.6	-3699.8
	$n^2$	-6275	-5028	-4027	-4473.1	-4527.0
D-Wave		-6275	-5025	-3551	-4447.7	-4525
$G_4$	$n$	-6275	-3930	-785	-2609.3	-2708.5
(Peg.)	$n^{1.5}$	-6275	-5028	-3580	-4305.5	-4281.0
	$n^2$	-6275	-5278	-4530	-5035.9	-5028.0

for several numbers of iterations per plateau of temperature. Note that the “D-Wave” line results from the random selection of 30 outputs from the 10000 runs that lead to Table 2

can observe a shift in the number of iterations per plateau to achieve optimality with almost certainty, e.g., for  $G_4$  this occurs only for  $n^2$  iterations per plateau). Yet, as shown in the previous section, quantum annealing was not able to solve them satisfactorily (for  $G_3$  and  $G_4$ ). Also, note that computing time is not issue when solving these instances: simulated annealing runs natively in less than 5 s ( $G_4$  with  $n^2$  iterations per plateau) on an average laptop PC with only moderately optimized code.

### Studying the Topology Bias

Let us emphasize that this comparison between our simulated annealing and the results obtained on the D-Wave 2X is perfectly fair as we compare the optimization capabilities of two devices coming with their operational constraints. Yet, it should also be emphasized that, for example on  $G_4$ , simulated annealing solved a 125 variables QUBO problems while quantum annealing had to solve an (artificially) much larger 958 variables QUBO. So, although the larger QUBO is equivalent to the smaller one, it is worth investigating whether or not these expanded QUBO are somehow harder to solve by simulated annealing.

To do so, we have considered the QUBO instances obtained after mapping the original QUBO for  $G_4$ <sup>9</sup> on both the Chimera and Pegasus topologies and attempted to solve them, this time, by simulated annealing.

Thus, Table 4 provides the results obtained when solving the expanded QUBO instances for  $G_4$  on both the Chimera and Pegasus topologies by means of classical annealing (also considering several numbers of iterations per plateau, as in the previous section). This time, the results obtained on the D-Wave are competitive with those obtained by simulated annealing which means that the expanded instances are much harder to solve (by simulated annealing) than the raw

ones, despite them being equivalent and despite the larger number of iterations per plateau (since there are more variables in the QUBO) i.e., the additional computing time, invested to solve them. In addition, probably not unsurprisingly, the denser Pegasus topology leads to smaller expanded QUBO instances than the Chimera one and provides better results (with simulated rather than quantum annealing as the first machines with that topology are just being released). Yet, although this topology is better, the results obtained remain very far from those obtained by simulated annealing, in “Resolution by simulated annealing” section, on the raw non-expanded QUBO instances. In terms of “computing” time, however, the D-Wave is several orders of magnitude faster. Indeed, when it takes less than a second to perform 10000 quantum annealing runs, solving the expanded ( $\approx 1000$  variables)  $G_4$  QUBO by simulated annealing (with  $n^2$  iterations per plateau) now takes several minutes on an average laptop computer.<sup>10</sup>

So, as a more general conclusion to this section, it appears that the D-Wave machine is competitive with a (heavy weight) simulated annealing algorithm with  $n^2$  iterations per plateau in terms of optimization quality and inherently several orders of magnitude faster. However (and that is a “big” however), it also appears that having to embed QUBO instances in either the Chimera or the Pegasus topologies tend to produce larger obfuscated QUBO which are much harder to solve by simulated annealing. This therefore hints that this is also counterproductive for quantum annealing and that these qubits interconnect topologies should be blamed, at least in part, for the relatively disappointing results reported in “Results summary” section.

<sup>9</sup> We limited ourselves to  $G_4$  as it is the larger instance we have been able to solve on the D-Wave 2X we had access to.

<sup>10</sup> And simulated annealing being an inherently sequential algorithm, it does not parallelize well.

## Discussion and Perspectives

In this paper, our primary goal was to provide a study on the behavior of an existing quantum annealer when confronted to old combinatorial beasts known to defeat classical annealing. At the very least, our study demonstrates that these special instances of the maximum (bipartite) matching problem are not at all straightforward to solve on a quantum annealer and, as such, are worth being included in a standard benchmark of problems for these emerging systems. Furthermore, as this latter problem is polynomial (and the specific instances considered in this paper even have straightforward optimal solutions), it allows to precisely quantify the quality of the solutions obtained by the quantum annealer in terms of distance to optimality. There also are a number of lessons learnt. First, the need for qubit duplication severely limits the size of the problem which can be mapped on the device leading to a ratio between 5 and 10 qubits for 1 problem variable. Yet, a  $\approx 1000$  qubits D-Wave can tackle combinatorial problems with a few hundred variables, a size which is clearly nontrivial. Also, the need to embed problem constraints (e.g., in our case, matching constraints requiring that each vertex is covered at most once) in the economic function, even with carefully chosen penalty constants, often lead to invalid solutions. This is true both in terms of qubits duplication consistency issues (i.e., qubits representing the same problem variable having different values) as well as for problem-specific constraints. This means that operationally using a quantum annealer requires one or more post-processing steps (e.g., solving qubit duplication inconsistencies by majority voting), including problem-specific ones (e.g., turning invalid matchings to valid ones).

Of course, the fact that, in our experiments, the D-Wave failed to find optimal solutions for nontrivial instance sizes, does not rule out the existence of an advantage of quantum annealing *as implemented in D-Wave systems* over classical annealing (the existence of which, as previously emphasized, as already been established on specially designed problems [1]). However, our results tends to rule out (or confirm) the absence of an exponential advantage in the general case of quantum over classical annealing. In addition, since the present study takes a worst-case (instances) point of view, it does not at all imply that D-Wave machines cannot be practically useful, and, indeed, its capacity to anneal in a few tens of  $\mu\text{s}$  makes it inherently very fast compared to software implementations of classical annealing. Stated otherwise, in the line of [24], the present study provides additional experimental evidences that there are (even non *NP*-hard) problems which are hard for both quantum *and* classical annealing and that on these quantum annealing does not perform significantly better.

Additionally, this paper experimentally demonstrates that dealing with the qubits interconnection topology issue in existing quantum annealers is a necessary step on the road to unleash the full potential of this technology. First, the need for qubit duplication severely limits the size of the problems which can be mapped on quantum annealing devices. Furthermore, this need for duplication also tends to obfuscate the optimization problem to be solved leading to results of significantly lower quality. This fact unfortunately tends to obliterate the overwhelming timing advantage of quantum over simulated annealing. Hence, although there might of course be many pitfalls laying ahead quantum analog computing, we argue that unless much denser qubits interconnects are developed, it will be difficult for the approach to compete with classical algorithms on real-world problems, both in terms of size and model complexity, even if the number of qubits keeps on increasing.

In terms of perspectives, it would of course be interesting to test larger instances on D-Wave machines with more qubits. It would also be very interesting to benchmark a device with the next generation of D-Wave qubit interconnection topology (the so-called Pegasus topology [11]) which is significantly denser than the Chimera topology and also have larger coherence domains. Both these advances should be relevant for the possible outcomes of the computation, but that requires testing. On the more theoretical side of things, trying to port Sasaki and Hajek proof [26] to the framework of quantum annealing, although easier said than done, is also an insightful perspective. Lastly, bipartite matching over the  $G_n$  graphs family also gives an interesting playground to study or benchmark emerging classical quantum-inspired algorithms (e.g. Simulated Quantum Annealing [10]) or annealers.

**Acknowledgements** The authors wish to thanks Daniel Estève and Denis Vion, from the Quantronics Group at CEA Paris-Saclay, for their support and fruitful discussions. The authors would also like to warmly thank Pr Daniel Lidar for granting them access to the D-Wave 2X operated at the University of Southern California Center for Quantum Information Science & Technology on which our experiments were run as well as for providing precious feedback and suggestions on early versions of this work.

## Compliance with Ethical Standards

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethical statement** This research work did not include work on or with human or animal subjects.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source,

provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Albash T, Lidar D. Demonstration of a scaling advantage for a quantum annealer over simulated annealing. *Phys Rev X*. 2018;8.
- Albash T, Lidar DA. Demonstration of a scaling advantage for a quantum annealer over simulated annealing. *Phys Rev X*. 2018;8(3):031016.
- Baker T, Gill J, Solovay R. Relativizations of the  $P = ?NP$  question. *SIAM J Comput*. 1975;4:431–42.
- Bian Z, Chudak F, Israel R, Lackey B, Macready WG, Roy A. Discrete optimization using quantum annealing on sparse ising models. *Front Phys*. 2014;2:56.
- Bian Z, Chudak F, Israel RB, Lackey B, Macready WG, Roy A. Mapping constrained optimization problems to quantum annealing with application to fault diagnosis. *Front ICT*. 2016;3:14.
- Boixo S, Albash T, Spedalieri FM, Chancellor N, Lidar DA. Experimental signature of programmable quantum annealing. *Nat Commun*. 2013;4:2067.
- Boothby K, Bunyk P, Raymond J, Roy A. Next-generation topology of d-wave quantum processors. 2020 arXiv preprint [arXiv:2003.00133](https://arxiv.org/abs/2003.00133)
- Burkard RE, Fincke U. Probabilistic asymptotic properties of some combinatorial optimization problems. *Discr Math*. 1985;12:21–9.
- Cerny V. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *J Optim Theory Appl*. 1985;5:41–51.
- Crosson E, Harrow AW. Simulated quantum annealing can be exponentially faster than classical simulated annealing. In: *IEEE FOCS*, 2016;pp. 714–723
- Dattani N, Szalay S, Chancellor N. Pegasus: The second connectivity graph for large-scale quantum annealing hardware. [arXiv:1901.07636](https://arxiv.org/abs/1901.07636) (2019)
- Farhi E, Goldstone J, Gutmann S. Quantum adiabatic evolution algorithms versus simulated annealing. *Tech. Rep.* 2002;0201031, [arXiv:quant-ph](https://arxiv.org/abs/quant-ph)
- Farhi E, Goldstone J, Gutmann S, Lapan J, Lundgren A, Preda D. A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. *Science*. 2001;292:472–6.
- Farhi E, Goldstone J, Gutmann S, Sipser M. Quantum computation by adiabatic evolution. [arXiv:0001106](https://arxiv.org/abs/0001106) (2000)
- Geman S, Geman D. Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *IEEE Trans Pattern Anal Mach Intell*. 1984;721–741
- Hajek B. Cooling schedules for optimal annealing. *Math Oper Res* 1988;311–329
- Harris R, Johansson J, Berkley AJ, Johnson MW, Lanting T, Han S, Bunyk P, Ladizinsky E, Oh T, Perminov I, et al. Experimental demonstration of a robust and scalable flux qubit. *Phys Rev B*. 2010;81(13):134510.
- Frenk JBG, Kan MHAHGR. Asymptotic properties of the quadratic assignment problem. *Math Oper Res*. 1985;10:100–116.
- Kirkpatrick S, Gelatt CD Jr, Vecchi MP. Optimization by simulated annealing. *Science*. 1983.
- Korte B, Vygen J. *Combinatorial optimization, theory and algorithms*. Berlin: Springer; 2012.
- Nolte A, Schrader R. Simulated annealing and its problems to color graphs. In: *AlgorithmsESA 96, Lecture Notes in Computer Science*, vol. 1136, pp. 138–151. Springer (1996)
- Preskill, J. Quantum computing in the nisq era and beyond. *Tech Rep* [arXiv:1801.00862](https://arxiv.org/abs/1801.00862) (2018)
- Quantum GA, collaborators: Quantum supremacy using a programmable superconducting processor 2019
- Reichardt BE. The quantum adiabatic optimization algorithm and local minima. In: *ACM STOC*, pp. 502–510 (2004)
- Santoro GE, Martonak R, Tosatti E, Car R. Theory of quantum annealing of spin glass. *Science*. 2016;295:2427–30.
- Sasaki GH, Hajek B. The time complexity of maximum matching by simulated annealing. *J ACM*. 1988;35:387–403.
- Schauer J. Asymptotic behavior of the quadratic knapsack problems. *Eur J Oper Res*. 2016;255:357–63.
- Vert D, Sirdey R, Louise S. On the limitations of the chimera graph topology in using analog quantum computers. In: *Proceedings of the 16th ACM international conference on computing frontiers*, pp. 226–229. ACM (2019)
- Vert D, Sirdey R, Louise S. Revisiting old combinatorial beasts in the quantum age: quantum annealing versus maximal matching. *Tech. Rep.* [arXiv:1910.05129](https://arxiv.org/abs/1910.05129) quant-ph

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.