Readiness of Quantum Optimization Machines for Industrial Applications

Alejandro Perdomo-Ortiz, Alexander Feldman, Asier Ozaeta, Sergei V. Isakov, Zheng Zhu, Bryan O'Gorman, Helmut G. Katzgraber, Alexander Diedrich, Hartmut Neven, Johan de Kleer, Brad Lackey, and Rupak Biswas

Phys. Rev. Applied **12**, 014004 — Published 2 July 2019

DOI: 10.1103/PhysRevApplied.12.014004

On the readiness of quantum optimization machines for industrial applications

Alejandro Perdomo-Ortiz,[1,2,3,4,*] Alexander Feldman,[5] Asier Ozaeta,[6] Sergei V. Isakov,[7] Zheng Zhu,[8] Bryan O'Gorman,[1,9,10] Helmut G. Katzgraber,[8,11,12] Alexander Diedrich,[13] Hartmut Neven,[14] Johan de Kleer,[5] Brad Lackey,[15,16,17] and Rupak Biswas[18]

[1]*Quantum Artificial Intelligence Lab., NASA Ames Research Center, Moffett Field, California 94035, USA*
[2]*USRA Research Institute for Advanced Computer Science (RIACS), Mountain View California 94043, USA*
[3]*Zapata Computing Inc., 439 University Avenue, Office 535, Toronto, ON, M5G 1Y8*
[4]*Department of Computer Science, University College London, WC1E 6BT London, UK*
[5]*Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, California 94304, USA*
[6]*QC Ware Corp., 125 University Ave., Suite 260, Palo Alto, California 94301, USA*
[7]*Google Inc., 8002 Zurich, Switzerland*
[8]*Department of Physics and Astronomy, Texas A&M University, College Station, Texas 77843-4242, USA*
[9]*Berkeley Center for Quantum Information and Computation, Berkeley, California 94720 USA*
[10]*Department of Chemistry, University of California, Berkeley, California 94720 USA*
[11]*1QB Information Technologies (1QBit), Vancouver, British Columbia, Canada V6B 4W4*
[12]*Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, New Mexico 87501, USA*
[13]*Fraunhofer IOSB-INA, Lemgo, Germany*
[14]*Google Inc., Venice, California 90291, USA*
[15]*Joint Center for Quantum Information and Computer Science, University of Maryland, College Park, Maryland 20742, USA*
[16]*Departments of Computer Science and Mathematics, University of Maryland, College Park, Maryland 20742, USA*
[17]*Mathematics Research Group, National Security Agency, Ft. George G. Meade, Maryland 20755, USA*
[18]*Exploration Technology Directorate, NASA Ames Research Center, Moffett Field, California 94035, USA*

(Dated: May 29, 2019)

There have been multiple attempts to demonstrate that quantum annealing and, in particular, quantum annealing on quantum annealing machines, has the potential to outperform current classical optimization algorithms implemented on CMOS technologies. The benchmarking of these devices has been controversial. Initially, random spin-glass problems were used, however, these were quickly shown to be not well suited to detect any quantum speedup. Subsequently, benchmarking shifted to carefully crafted synthetic problems designed to highlight the quantum nature of the hardware while (often) ensuring that classical optimization techniques do not perform well on them. Even worse, to date a true sign of improved scaling with the number of problem variables remains elusive when compared to classical optimization techniques. Here, we analyze the readiness of quantum annealing machines for real-world application problems. These are typically not random and have an underlying structure that is hard to capture in synthetic benchmarks, thus posing unexpected challenges for optimization techniques, both classical and quantum alike. We present a comprehensive computational scaling analysis of fault diagnosis in digital circuits, considering architectures beyond D-wave quantum annealers. We find that the instances generated from real data in multiplier circuits are harder than other representative random spin-glass benchmarks with a comparable number of variables. Although our results show that transverse-field quantum annealing is outperformed by state-of-the-art classical optimization algorithms, these benchmark instances are hard and small in the size of the input, therefore representing the first industrial application ideally suited for testing near-term quantum annealers and other quantum algorithmic strategies for optimization problems.

I. INTRODUCTION

Quantum annealing (QA) [1–7] has been proposed as the most natural quantum computing framework to tackle combinatorial optimization problems, where finding the configuration that minimizes an application-specific cost function is at the core of the computational task. Despite multiple studies [8–20], a definite detection of quantum speedup [13, 21] remains elusive. Random spin-glass benchmarks [13] have been shown to be deficient in the detection of quantum speedup [11, 14], which is why the community has shifted to carefully-crafted synthetic benchmarks [19, 20]. While these have shown that QA has a constant speedup over state-of-the-art classical optimization techniques, their value for real-world applications remains controversial.

Although the first proposal for a QA implementing combinatorial optimization problems with real constrains as they appear in real-world application was proposed close to a decade ago [22], the question of whether a quantum annealer can have a quantum speedup on any real-world applications remains an open one. From the many applications implemented in quantum annealers (see for example, Refs. [17, 23–28]), fault diagnosis has been one of the leading candidates to benchmark the performance of D-Wave devices as optimizers [26, 29]. From the range of circuit model-based fault diagnosis problems [30] we restrict our attention here to *combinational circuit fault diagnosis* (CCFD), which in contrast to sequential circuits, does

*Electronic address: alejandro@zapatacomputing.com

not have any memory components and the output is entirely determined by the present inputs.

Using CCFDs, we illustrate the challenges and the readiness of quantum annealers for solving real-world problems by providing a comprehensive computational scaling analysis of a real-world application. We compare quantum Monte Carlo (QMC) simulations and QA experiments on the D-Wave Systems Inc. D-Wave 2X quantum annealer to several state-of-the-art classical solvers on conventional computer hardware. More specifically, our work is motivated by these open questions in quantum optimization with QA hardware:

1. What is the payoff of investing in the construction of specialized quantum hardware that natively matches the connectivity and interactions (e.g., many-body terms in higher-order Hamiltonians) dictated by the cost function of an actual application?

2. What could be the impact in the computational scaling of different annealing schedules or the addition of more complex driver, such as non-stoquastic Hamiltonians?

3. Does quantum Monte Carlo reproduce the computational scaling of the current generation of D-Wave QA machines?

Keeping in mind these are very general and ambitious goals for a single work like the one presented here, we focus our scope only to the case of optimization instances generated from this real-world scenarios. We discuss the importance of each of these algorithmic and architectural design aspects related to each of the questions above, from an application-centric and physics-focused perspective, providing answers or insights only in some cases and under the assumptions and computational resources described throughout this work. It is demonstrated that CCFD instances based on Boolean multiplier circuits are harder than other representative random spin-glass benchmarks. This makes the diagnosis of Boolean multipliers a prime application for benchmarking QA architectures. Since our work hints the need for further developments, with the inclusion of more powerful driver Hamiltonians among one of the interesting research direction in the search for quantum advantage, CCFD instances are ideal industrial application problems for testing such incremental improvements in near-term quantum annealers and novel quantum algorithmic strategies for optimization problems.

Although tangential to the key results in this paper, in Appendix D we discuss the last of the three questions above. The main reason for including this section is to highlight that from our perspective of the first scaling analysis of a real-world application, our results indicate that given the hardness of our instances compared to synthetic data sets, the scaling becomes a moot question. This is, even assuming a favorable scenario where SQA scaling slope matches the DW2X scaling, the prefactor is large enough that attempting to use computational resources for simulating SQA becomes prohibitedly expensive. This has not been the case with other studies on synthetic instances [19].

II. BENCHMARK PROBLEM

To benchmark quantum annealers with different physical hardware specifications, we generate a family of multiplier circuits of varying size. The circuit size is determined by the size of two binary numbers of bit-lengths n and m, respectively, to be multiplied. Figure 1 illustrates the layout of the multiplication circuit for two binary numbers, each of length k.

The optimization problem consists in diagnosing the health status of each of the gates in the circuit, given an observation vector consisting of inputs and outputs, as illustrated in Fig. 2. For the generation of the problem instances, we focus on problems where the output is not consistent with the multiplication of the two input numbers and therefore the system is expected to have at least one fault. Under the assumption that all the gates have the same failure probability, the problem of finding the most probable diagnosis is reduced to finding the valid diagnoses with the minimal number of faults (see Appendix C for details). It is important to note that all CCFD instances used in this study were randomly generated by injecting a number of faults equivalent to the number of outputs in the circuit [$(n + m)$ for a (n-bit) $\times$(m-bit) multiplier circuit]. After the random fault injection of cardinality $(n + m)$, a random input is generated and the corresponding output is obtained by propagation of the input under the corresponding fault injection. Hence, we guarantee that every random input/output pair generated this way has at least one solution. The simpler strategy of generating random input/output vectors can lead to problems that do not have a solution under the diagnosis model. In the case of instances with many valid minimal solutions, we count all the ones found by the stochastic algorithms in the estimation of the success probability.

From a computational complexity perspective, the CCFD problem is NP-hard [32], and it corresponds to the minimization task we aim to solve either with QA on the D-Wave 2X device (DW2X) at NASA, a continuous time version [33] of simulated quantum annealing (SQA) [6, 12, 13] as a QMC-based solver, or other classical optimization techniques, such as simulated annealing (SA) [34, 35], parallel tempering Monte Carlo (modified as a solver) [36–38] combined with isoenergetic cluster updates [39] (PTICM), or current specialized SAT-based solvers tailored for this CCFD problem described in Appendix B.

To perform a scaling analysis it is key to be able to generate a data set with varying input size and with a high intrinsic hardness such that classical solvers have a harder time, increasing the chances that our instances fall into the hard asymptotic regime for both classical and quantum approaches. This has been one of the challenges for benchmarking early QA devices, where the first proposals [12, 13] were convenient but turned out to be too easy for benchmarking purposes [11]. More recently, benchmarking have focused on carefully-designed synthetic problems [16, 19, 20, 40]. However, as we shall demonstrate in Sec. III B, CCFD-based problems are the hardest benchmarking problems currently available.

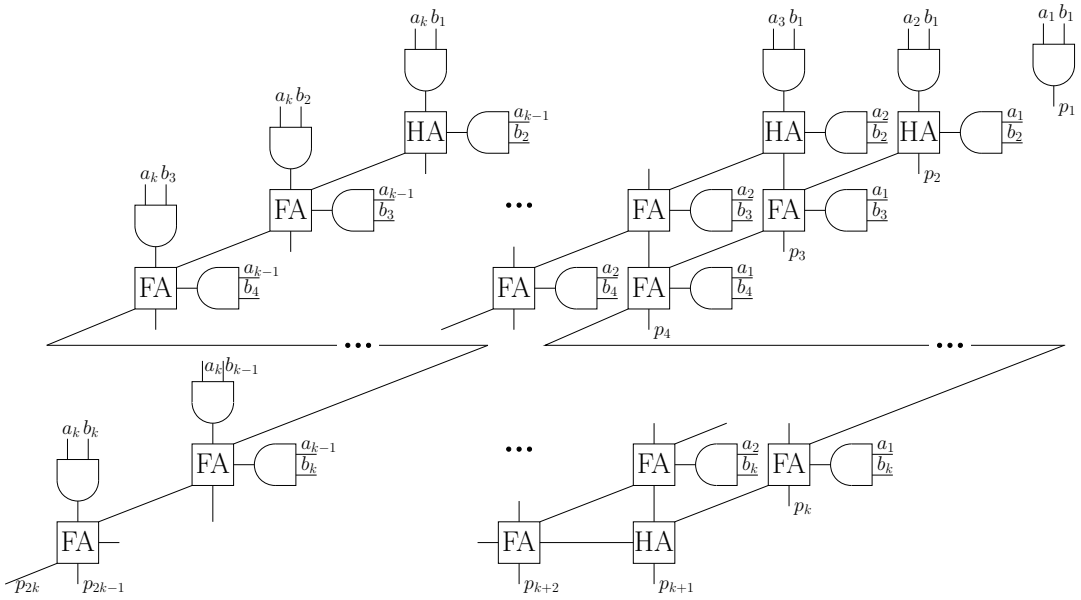

FIG. 1: Multiplier circuits used to generate our CCFD benchmark instances. In this example the multiplication of two numbers represented as k-digit binary numbers, $a_1a_2\cdots a_k$ and $b_1b_2\cdots b_k$ is shown, resulting in a product output of length $2k$, corresponding to $p_1p_2\cdots p_{2k}$. HA and FA denote half-adder and full-adder circuit modules, respectively.

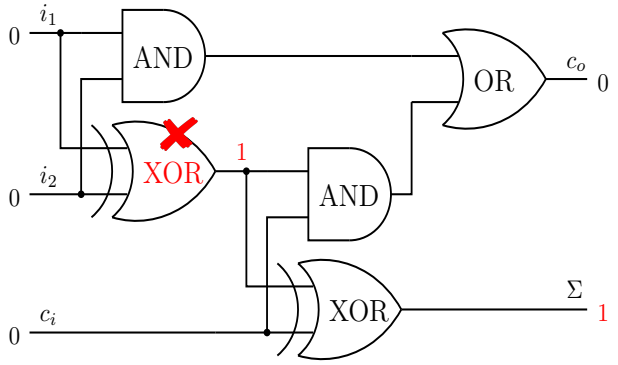

FIG. 2: Example of a model-based fault-diagnosis in combinational circuits (CCFD) on a small full adder circuit. In this work, the CCFD optimization problem consists in finding a smallest set of Boolean gate outputs that, when stuck-at-one, match an input/output observation vector. This setting where one restricts the expected fault behavior of the gates is known as the *strong-fault model* (see, for example, Ref. [31]). Although in principle each gate can have a characteristic fault mode, without loss of generality, we adopted stuck-at-one as the fault mode for all gates. Generalizations to other common fault modes and multiple fault modes per gate are detailed in Appendix C. In this example, the flagged XOR gate is faulty, because its nominal behavior should yield an output equal to zero. The diagnosis explains the input $\{i_1 = 0, i_2 = 0, c_i = 0\}$ and the apparently anomalous output $\{c_0 = 0, \Sigma = 1\}$.

III. RESULTS AND DISCUSSION

A. Benchmarking real-world applications

Figure 3 summarizes the main challenges when benchmarking applications with QA devices. The first step consist of translating the standard format describing the rules and constrains of the minimization problem into a pseudo-Boolean polynomial function $H_{\rm P}(\mathbf{s}_{\rm P})$, with domain $\mathbf{s}_{\rm P} \in \{+1, -1\}^{N_{\rm P}}$ and co-domain in $\mathbb{R}$. Appendix C details the construction of $H_{\rm P}(\mathbf{s}_{\rm P})$ for this problem of minimal fault diagnosis in combinational circuits. The task to be solved consists in finding, within the search space with $2^{N_{\rm P}}$ possible solutions, the assignment $\mathbf{s}^*_{\rm P}$ that minimizes $H_{\rm P}(\mathbf{s}_{\rm P})$. Because the pseudo-Boolean function is a polynomial expression in the binary variables $\mathbf{s}_{\rm P}$, this optimization problem is known as a PUBO problem which stands for *polynomial unconstrained binary optimization* problem. Note that sometimes these problems are also referred to as HOBOs, i.e, *higher-order binary optimization* problem. The specific case of a quadratic function leads to the known QUBO [41] which is the type that is natively implemented in D-Wave quantum annealers. See Appendix. A for more details on the QA implementation.

In the case of the benchmark of multiplier circuits, the standard problem description format is a list of propositional logic formulas similar to the ones given in Fig. 3, corresponding to the nominal behavior of each gate within the full-adder circuit illustrated in Fig. 2. For the case of the strong-fault model [31] considered here, one needs to add specific propositional logic formulas associated with the expected behavior when each gate is faulty. Without loss of generality, and for the purpose of the benchmark generated here, we considered that whenever any gate fails, it would be in a *stuck-at-one* mode or equivalently, in propositional logic, $f_i \Rightarrow z_i$. Here f_i denotes the health variable associated with the i-th gate and z_i its corresponding gate output. Note that $f_i = 1$ means faulty and $f_i = 0$ nominal. Extensions to other fault modes are described

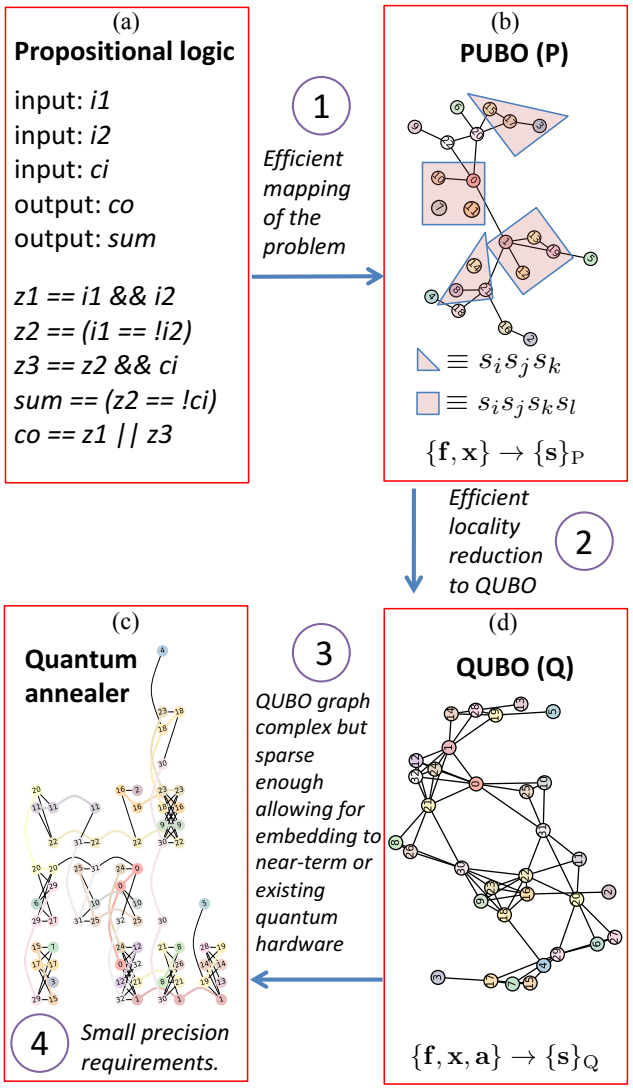

FIG. 3: Generation of benchmarks driven with real-world scenarios flowchart. The first challenge consists in finding an efficient translation of the natural description of the optimization problem into a pseudo-Boolean function [panel (a)], $H_{\mathrm{P}}(\mathbf{s}_{\mathrm{P}})$, with domain $\mathbf{s}_{\mathrm{P}} \in \{+1, -1\}^{N_{\mathrm{P}}}$ and co-domain in $\mathbb{R}$. The resulting polynomial unconstrained binary optimization (PUBO) problem consists in finding assignments $\mathbf{s}_{\mathrm{P}}^{*}$ which minimizes the quartic-degree polynomial $H_{\mathrm{P}}(\mathbf{s}_{\mathrm{P}})$ [panel (b)]. This specific degree, known as the *locality* of the Hamiltonian (here, 4-local), arises from the effective local interactions between gates with two input variables $x_{i,1}$ and $x_{i,2}$, the wire output z_i, and the health variable, f_i, associated to each gate. By adding ancilla qubits, $(\mathbf{a})$, the quartic (4-local) $s_i s_j s_k s_l$ and cubic (3-local) $s_i s_j s_k$ terms can be reduced to an effective 2-local Hamiltonian defining an effective quadratic unconstrained binary optimization (QUBO) version of the problem instance [panel (d)]. Finally, minor-embedding can be used to embed the QUBO into the physical hardware – in this case the chimera structure (see Fig. 7) of the D-Wave quantum annealers [panel (c)]. The cost of the embedding is an additional overhead in the number of qubits. While the "propositional logic" panel contains the description of the full-adder circuit in Fig. 2, the remaining are realistic representations of one of the smallest instances from our multiplier circuit with 23, 33, and 72 qubits (or spin variables), for its PUBO, QUBO, and DW2X representation, respectively. In this work, we assess the impact on the performance of each of these representations and also perform experiments on the DW2X. Steps 1 – 4 denote some of the desiderata for an application to be a potential candidate for benchmarking next generation of quantum annealers. Note that while the D-Wave device requires the embedding of a QUBO. However, future hardware implementations might include k-local interactions with $k > 2$. Therefore, we perform the classical simulations both in the QUBO and PUBO representations to compare both approaches.

in Appendix. C. In the specific mapping considered here, $\mathbf{s}_{\mathrm{P}}$ contains the health variables, along with variables specifying the values for each of the internal wires within the multiplier circuit.

A generic classical solver such as SA or PTICM can tackle the optimization problem in th PUBO representation directly because one can easily evaluate $H_{\mathrm{P}}(\mathbf{s}_{\mathrm{P}})$. As shown in Sec. III C, working in this PUBO representation is the preferred approach from the application perspective. As mentioned in the explicit mapping construction in Sec. C, $H_{\mathrm{P}}(\mathbf{s}_{\mathrm{P}})$ is a polynomial with at most quartic degree, independent of the circuit size. A quantum annealer capable of implementing such quartic polynomials can certainly aim at solving the problem in this representation. Given the possibility of such experimental designs (see, for example, Ref. [42]), we also consider hypothetical quantum annealers that we study using SQA to assess the impact in the performance of working with a quantum annealer that can natively solve the PUBO problem. Unfortunately, no such devices exist to date and there is an overhead in representing the quartic (4-local) and cubic (3-local) monomials in $H_{\mathrm{P}}(\mathbf{s}_{\mathrm{P}})$ with a resulting only-quadratic expression (2-local). The contraction techniques [43] used to reduce the locality incur an overhead of variables by introducing ancillas (for a tutorial of a specific practical example see Ref. [22]). This is not desirable because it increases the search space from $2^{N_{\mathrm{P}}}$ to $2^{N_{\mathrm{Q}}}$, with N_{Q} the number of variables $\mathbf{s}_{\mathrm{Q}}$ in the resulting new quadratic expressions $H_{\mathrm{Q}}(\mathbf{s}_{\mathrm{Q}})$ as the new representation from $H_{\mathrm{P}}(\mathbf{s}_{\mathrm{P}})$. $\mathbf{s}_{\mathrm{Q}}$ is now the union of the health variables $\mathbf{f}$, the wires $\mathbf{x}$, and the ancilla set $\mathbf{a}$. The overhead is linear in our case as shown in Fig. 9. The next challenge presented in Fig. 3 towards implementing a real-world application is that most likely there will be quadratic term in $H_{\mathrm{Q}}(\mathbf{s}_{\mathrm{Q}})$ representing qubit-qubit interactions not present in the physical hardware. This will be the case unless one specifically designs the layout of the quantum annealer hardware to match the resulting connectivity graph dictated directly by the application through $H_{\mathrm{Q}}(\mathbf{s}_{\mathrm{Q}})$. Representing the logical graph within another graph is called the minor-embedding problem [44]. For the case of the connectivity graph pre-defined in the D-Wave devices, also known as *chimera* graph, we use the heuristic solver developed in Ref. [45]. As can be seen in Fig. 9, the overhead is linear given the relatively sparsity

of the graphs resulting from the multiplier circuits. This is an encouraging result given that the overhead for an all-to-all connectivity graph embedded onto the chimera architecture is quadratic in the number of variables. A much larger problem than minor embedding when embedding an application onto a limited connectivity hardware graph is parameter setting. For example, there is no rule of thumb as to how strong the couplers for a set of ferromagnetically-coupled qubits defining a physical qubit should be. A sweet-spot value is expected, however it is not easy to determine or predict in the most general setting. In this work, and for all the experiments on the D-Wave 2X, we used the strategy proposed in Ref. [46] for both setting the strength of the ferromagnetic couplers and for the selection of gauges. The final challenge when embedding applications is the requirement that the pseudo-Boolean function to be minimized has a low precision requirement because analog QA machines operate on a limited precision dictated by the intrinsic noise and finite dynamical range of parameters found in these devices.

Summarizing, from our experience with applications, the CCFD instances considered here are the best candidate to match each one of the aforementioned requirements. The mapping from propositional logic to PUBO is compact and efficient given that in the digital circuits considered here all the input, outputs, health variables and wires are all binary variables, the resulting QUBO graph is sparse enough that the overhead to embed onto hardware is linear, and the randomly-generated instances have a higher intrinsic hardness compared to other random spin-glass previously studied, as will be shown in Sec. III B.

Although we do not expect the intrinsic exponential scaling of this problem to disappear for the worst case scenario by a mere change of representation or the solver used, the results could be different for each setting when computational times for typical instances are considered, and for the accesible problem sizes. The details and scaling slopes obtained for each of the approaches considered here are of extreme importance from a practical point of view, and used for addressing any meaningful advantage in the following sections.

B. Hardness compared to other random spin-glass benchmarks

Figure 4 addresses the hardness of instances embedded in the chimera topology (C) generated from the CCFD data set by comparing to random spin-glass problems used to benchmark the performance of D-Wave quantum annealers [see Eq. (A1) for the actual Hamiltonian to be minimized]. Bimodal instances were the first to be used in benchmarking studies [12, 13] and are the simplest to generate. For these, the available couplers in the D-Wave 2X are randomly chosen to be $J_{ij} \in \{\pm 1\}$ with biases $h_i = 0$. The reason why random bimodal instances are too easy for quantum and classical algorithms alike is their high degeneracy resulting in a large number of floppy spins. To overcome this problem, Refs. [14, 47] introduced couplers distributed according to Sidon sets [48] combined with post-selection procedures. These naturally increase the hardness of problems by reducing degeneracy to a minimum and removing floppy spins. For the case of Sidon instances [47] the values of the couplers J_{ij} are randomly selected from the set $\{\pm 5, \pm 6, \pm 7\}$, with $h_i = 0$. Planted/C instances correspond to an attempt to increase the hardness of random spin-glass instances (see Ref. [16]), but with a known solution. For the data shown, we asked the main author in Ref. [16], if he could provide us with the hardest set of instances he could generate; the only restriction being that they would need to be generated randomly and not being post-selected for hardness as the rest of all the other families of instances here. The attempt consisted of drawing the couplers from a continuous distribution instead of from a discrete distribution as the one in the original paper, Ref. [16], or as in the case of the Bimodal and Sidon set considered here.

Figure 4(a) illustrates that already for approximately 600 variables the CCFD/C instances are at least one order of magnitude harder than Sidon/C which is the hardest set among the random spin-glass problems. Figure 4(b) summarizes the asymptotic scaling of each of these problem types, clearly separating our CCFD instances from any of the random spin-glass instances, with Sidon and Bimodal having roughly the same scaling. Here we assume that the time-to-solution TTS in μs can be fit to $\mathrm{TTS} \sim \exp(b\sqrt{N})]$ with N the number of variables. This conclusion is independent of the percentile considered as shown in Fig. 10 in Appendix. F. Our results also show that the attempt to make hard planted instances did not provide any additional hardness compared to the other random spin-glass problems, at least when they are evaluated with PTICM. Therefore, the CCFD/C instances are not only harder in terms of computational effort, according to TTS, but also from a scaling perspective.

The data set Bimodal/CCFD provides insights as to why these instances are hard. There are three options of why these instances are intrinsically harder than any other random data set explored here. One option is that the underlying CCFD graph defined by the QUBO problem for each multiplier type has some sort of nontrivial long-range correlations or a much higher dimensionality in such a way that the problems, when minor-embedded onto the chimera lattice, become harder than typical chimera instances. Another explanation relies on the characteristic value of biases h and coupler values J in the Hamiltonian [Eq. (A1)] which could be responsible for the complex-to-traverse energy landscape. Furthermore, there could be interplay between the two aforementioned options. To address this question, we generate Bimodal instances on the *native* QUBO graph defined by multiplier circuits of varying sizes, denoted here as "Bimodal/CCFD." If the underlying graph contain features that intrinsically "host" hard instances, then one would expect that both the scaling and TTS could be different than those on the chimera graph. Figure 4 shows that the Bimodal/CCFD instances happen to be even easier than the Bimodal instances embedded onto the chimera graph. This means that the intrinsic hardness of these CCFD instances most likely is related to the structure and the relationship between the specific biases h and couplers J values defining them. Further studies are being performed to study this in more detail.

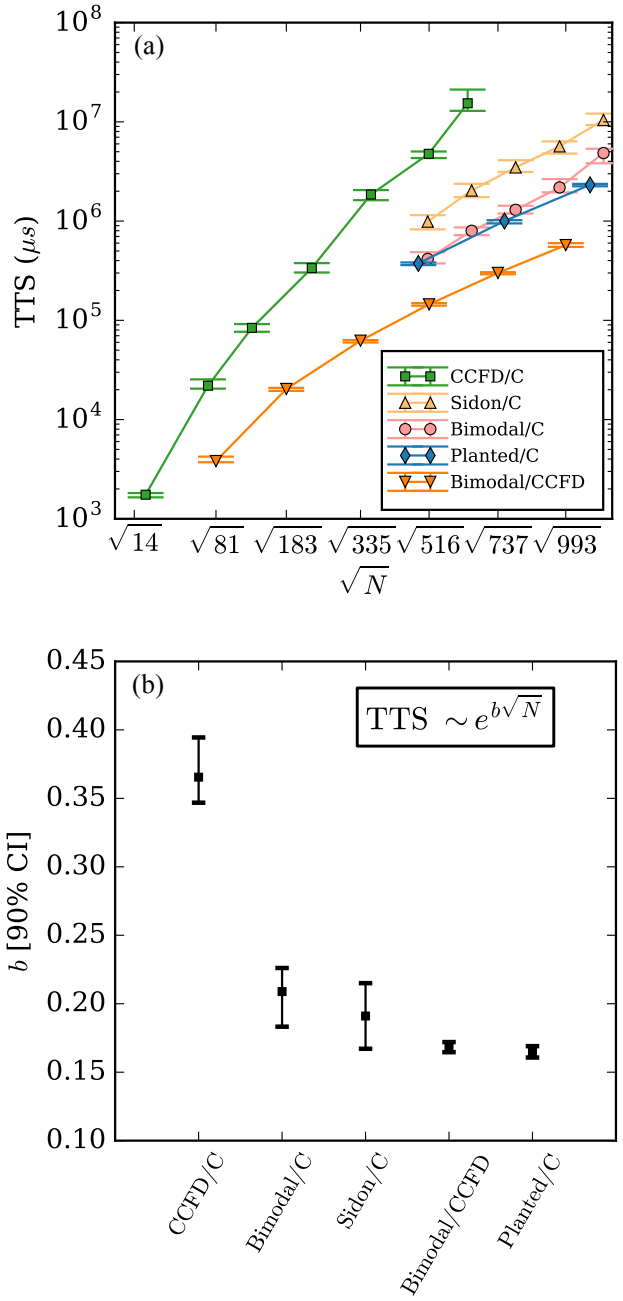

FIG. 4: Hardness comparison of the chimera representation of the CCFD instances with other representative random spin glass problems from the literature. (a) For consistency, all the time-to-solutions (TTS) in μs were obtained with PTICM with the same single-core processors. (b) Note the steeper scaling [larger value for b from a fit of the TTS to TTS $\sim \exp(b\sqrt{N})$] for the CCFD problems compared to the other classes of random spin-glass benchmarks. Data points correspond to the median values extracted from a bootstrapping statistical analysis from 100 instances per problem size, with error bars indicating the 90% confidence intervals (CI). The different instance classes are described in the main text.

C. Scaling analysis: Application vs physics perspective

Fig. 5 provides insights about the CCFD instances from a physics and from the application perspective. While in the former we analyze the scaling of computational resources via the time-to-solution TTS using the number of variables $\sqrt{N}$, in the latter we analyze the resource requirements by the application-specific variables, namely the type of multiplier used. The physics perspective here aims to answer questions about the performance of QA compared to other classical solvers on a comparable footing, ignoring for a moment that the instances were generated from a specific application. For example, we compare here the performance of QA to other classical and alternative quantum solvers on instances represented on a chimera graph (C); similar to previous extensive benchmarking work on synthetic random spin-glass instances. We go beyond such studies and provide as well insights on the performance of QA for the QUBO (Q) instances on their native graph dictated by the CCFD application and also on hypothetical quantum annealer devices capable of natively encoding up to quartic interactions (P).

For the physics scaling analysis we chose to plot the TTS computational effort as a function of $\sqrt{N}$, with N being the problem size in terms of number of spins, regardless of whether the problem to be minimized is in a PUBO (P), QUBO(Q), or chimera (C) format. This selection is motivated by the linear relation between any pair of problem sizes $N_{\rm P}$, $N_{\rm Q}$ or $N_{\rm C}$ (see Fig. 9) and the fact that the scaling for problems on the quasi-planar Chimera graphs is expected to be a stretched exponential, largely due to its tree width $\sim \sqrt{N_{\rm C}}$, in contrast to a tree width $\sim N$ characteristic of fully-connected graphs [49].

The analysis from the application perspective aims for insights on the performance where the sole purpose is to find the solution to the CCFD problem. Here, it is natural to plot the TTS computational effort as a function of a characteristic property of the circuit scaling with the problem size, regardless if one considers a symmetric multiplier, mult[n-n] or an asymmetric one, i.e., mult[n-m]. We chose this quantity to be the number of gates in the circuit, $N_{\rm gates}$ (or more precisely $\sqrt{N_{\rm gates}}$), which is justified given the linear relationship between $N_{\rm gates} \propto N_{\rm P} \propto N_{\rm C}$ illustrated in Fig. 9, and the expected stretched exponential behaviour in $\sqrt{N_{\rm C}}$ discussed above for chimera graphs.

a. Limited quantum speedup — Figure 5(a) compares the single-core computational effort of SA, PTICM, SQA (with both linear and DW2X schedules), and the experimental results obtained with the DW2X quantum annealer. Represented with diamond symbols in Fig. 6 and with values on the right axis, we plot the asymptotic analysis performed by considering only the four largest sizes from each of the data sets. From this scaling analysis and the value of the main scaling exponent b (slopes of curves in Fig. 5) for the chimera instances SA/C and DW2X, it can be seen that we also find here limited quantum speedup (without optimizing annealing schedules) [21] as found for the benchmarks on synthetic instances used in the study by the Google Inc. [19]. From this physics perspective, there seems to be even a quantum advan-

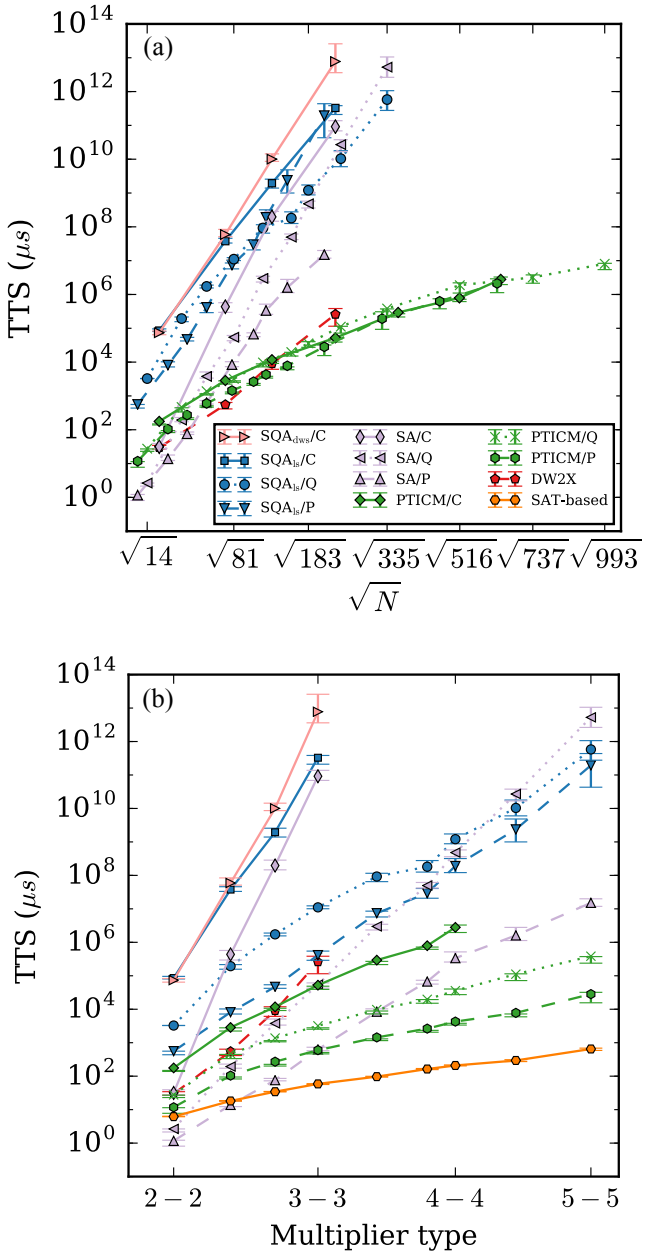

FIG. 5: Scaling analysis from a (a) physics, and (b) an application-centric perspective. The TTS is plotted as a function of $\sqrt{N}$, with N the number of spin variables in each of the problem representations [PUBO (P), QUBO (Q), or chimera (C)]. Panel (b) corresponds to $\sqrt{N_{\text{gates}}}$, with N_{gates} the number of gates regardless if we are considering symmetric multipliers, mult[n-n] as in Fig. 1, or asymmetric ones (mult[n-m]). The legend for the data sets depicted in panel (a) is shared with panel (b), with SAT-based results only appearing in panel (b). SQA/C runs were performed with an optimized linear schedule, as well as the DW2X schedule, marked with "ls" and "dws" subscripts, respectively, (details in Appendix D).

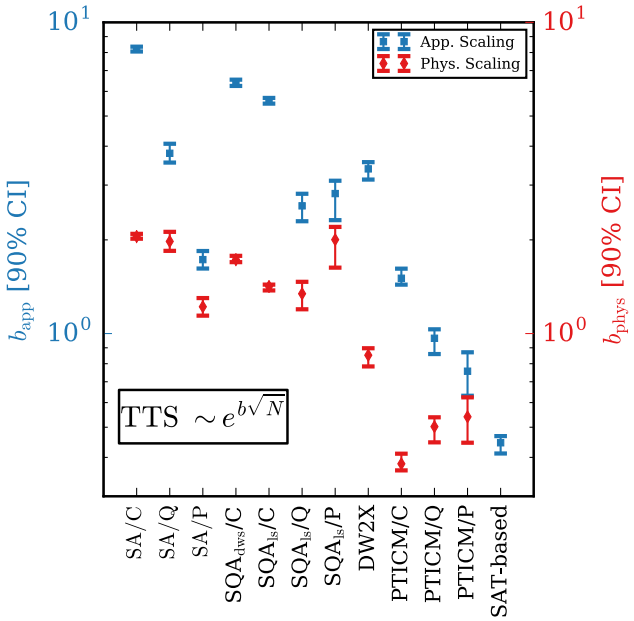

FIG. 6: Asymptotic scaling analysis. The asymptotic scaling exponent b_{app} refers to the multiplier representation, whereas b_{phys} refers to the physical representation of the problem. Data points correspond to the median values extracted from a bootstrapping statistical analysis from 100 instances per problem size, with error bars indicating the 90% confidence intervals (CI).

tage when comparing with SA at the PUBO level, SA/P, which happen to have a better scaling than both of their quadratic counterparts, SA/Q and SA/C. The values are close enough that one has to be careful because the real b_{phys} (DW2X) might be larger than the calculated in our analysis due to sub-optimal annealing time [13, 21, 50]. On the other hand, note that the quantum advantage at the level of the same representation where b_{phys}(DW2X) $\ll$ b_{phys}(SA/C) also holds against any of the optimized SQA/C implementations, either with a linear or DW2X schedule. Although we believe that it is very unlikely that sub-optimal time can change our *limited quantum speedup* conclusion because b_{phys}(DW2X) $\ll$ b_{phys}(SA/C), we note that optimized SQA corroborates these claims. This scaling advantage already yields a difference of approximately six orders of magnitude on a single-core CPU in the TTS between DW2X and SA/C for the largest problem studied (mult[4-4]). It is important to remind the reader that our results are with a fixed annealing time, and although we justify that it would be very unlikely that the slope of DW2X could match that of SA/C, the best practice to have conclusive limited quantum speedups results would be by optimizing the annealing time in the quantum annealers runs [13, 51]. Exploration of the impact of the optimal annealing time in the CCFD instances could be an interesting piece of work in its own and it is left as future work.

b. SQA vs DW2X and impact of the annealing schedule — From a computational prefactor perspective note that the computational effort for the DW2X is smaller by six to eight orders of magnitude than the SQA/C implementations with either linear or the D-wave schedule. It is important to note that the TTS in Figs. 5(a) and 5(b) is for SQA as a classical com-

putational solver. For a fair comparison of the scaling of SQA to that of a physical quantum annealer such as the DW2X, the SQA TTS results must be divided by N to account for the intrinsic parallelism in QA, as illustrated in Fig. 8. Further analysis of the scaling comparison of SQA and the DW2X device can be found in Appendix D.

Figures 5(a) and 5(b) illustrate that the selection of a poor schedule (the D-wave schedule in this case in comparison to simpler linear one) can have a significant impact in the computational efficiency of SQA as as classical computational solver. As discussed in Appendix D, most likely the difference is only at the level of a prefactor and most likely it is not a scaling advantage. Whether there are schedules that can change the asymptotic scaling remains an open question. In Appendix D we also discuss that although there seems to be a scaling advantage of the DW2X over the SQA simulations, the results are also inconclusive given that the scaling of the DW2X might be slightly different due to any sub-optimal annealing times. We leave it to future work to optimize the annealing time of the DW2X because it is beyond the scope of this work given the sizable computational requirements needed.

c. QA performance for Hamiltonians with higher-order interactions — A question not addressed to date is the performance comparison between QA architectures with 2-local and k-local ($k > 2$) interactions within the scope of real-world applications. For example, the CCFD mapping used in this work (see Appendix C for details) natively contains cubic (3-local) and quartic (4-local) interactions and one might think a quantum annealer natively encoding those might have an advantage over 2-local terms. Perhaps one of the most remarkable findings in this study from our SQA simulations is that working directly with a Hamiltonian containing such quartic interactions does not seem to help QA with a transverse field, because $b_{\rm phys}$(SQA/Q) $< b_{\rm phys}$(SQA/P). Note that this result is in contrast to the behavior of the classical algorithms considered here. In the case of SA there seems to be an advantage for solving the instances in the PUBO representation over the QUBO [$b_{\rm phys}$(SA/Q) $> b_{\rm phys}$(SA/P)]. In the case of PTICM, $b_{\rm phys}$(PTICM/Q) $\approx b_{\rm phys}$(PTICM/P). These remarks on the physics scaling have a significant impact on the scaling from the application perspective. Note that while in all the classical methods there is a clear preference to solve the problem in the PUBO representation, the case of SQA shows no advantage for the quantum annealer in the PUBO representation. On the contrary, as shown in Fig. 11 for the higher percentiles above the median, there seems to be a slight preference of SQA/Q over SQA/P not only in the absolute value of computational effort measured in TTS [see last data point in Fig. 11(d)], but also in scaling terms.

The insight to be extracted from the SQA simulations in the context of this CCFD application is that simply adding higher-order terms would not necessarily imply any enhancement in the performance. This result is striking for two reason.

First, because in the application scaling we plot the TTS results vs $\sqrt{N_{\rm gates}}$, then when changing representations from PUBO to QUBO, there is a natural tendency for $b_{\rm app}$(Q)/$b_{\rm app}$(P) $> b_{\rm phys}$(Q)/$b_{\rm phys}$(P). This is because $N_{\rm Q}$ is always greater than $N_{\rm P}$, and therefore even in the case of comparable physics scaling slopes as is the case of PTICM with $b_{\rm phys}$(PTICM/Q) $\approx b_{\rm phys}$(PTICM/P) this would imply that

$$\mathrm{TTS}_{\rm PTICM/P} \sim e^{b_{\rm phys}\sqrt{N_{\rm P}}} \sim e^{b_{\rm phys}\sqrt{\alpha_{\rm P\leftarrow g}}\sqrt{N_{\rm gates}}},$$

while

$$\mathrm{TTS}_{\rm PTICM/Q} \sim e^{b_{\rm phys}\sqrt{N_{\rm Q}}} \sim e^{b_{\rm phys}\sqrt{\alpha_{\rm Q\leftarrow P}\alpha_{\rm P\leftarrow g}}\sqrt{N_{\rm gates}}},$$

valid in the asymptotic limit $N_{\rm gates} \gg 1$ of interest here. Therefore,

$$b^{\rm Q}_{\rm app} = b_{\rm phys}\sqrt{\alpha_{\rm Q\leftarrow P}\alpha_{\rm P\leftarrow g}} > b^{\rm P}_{\rm app} = b_{\rm phys}\sqrt{\alpha_{\rm P\leftarrow g}}.$$

Here we have used that, in this limit, $N_{\rm P} \sim \alpha_{\rm P\leftarrow g} N_{\rm gates}$ and $N_{\rm Q} \sim \alpha_{\rm Q\leftarrow P} N_{\rm P}$, and both, $\alpha_{\rm Q\leftarrow P}$ and $\alpha_{\rm P\leftarrow g}$ are greater than 1, as shown in Fig. 9.

Second, the penalties of the locality reduction ancillas change the energy scale and it is expected that stochastic solvers such as SA (which heavily depend on the barriers in the energy landscape) also suffer from the new QUBO energy landscape with taller barriers. This is indeed what we observe because $b_{\rm phys}$(SA/Q) $> b_{\rm phys}$(SA/P). Note that PTICM seems to be more resilient to these barriers and, as discussed before, $b_{\rm phys}$(PTICM/Q) $\approx b_{\rm phys}$(PTICM/P). Both of these driving forces would imply that the application perspective scaling of QA working in the PUBO representation should be better than in the QUBO representation and it is not what we observe here. This second explanation is reasonable and a good indication that SQA is doing a good job at not "feeling" these taller barriers, something that could be explained by means of quantum tunneling.

From the first argument it follows that $b_{\rm app}$(SQA/Q) $\approx b_{\rm app}$(SQA/P), implying that $b_{\rm phys}$(SQA/Q) $< b_{\rm phys}$(SQA/P) which is quite distinctive and different from what we observe in the classical approaches. It is clear that SQA is having a harder time traversing the PUBO energy landscape and finding the ground state in this representation, despite the smaller problem size. One plausible explanation is that the transverse-field implementation is not powerful enough to take advantage of the compactness of the PUBO energy landscape. We thus emphasize that any development of new architectures with k-local couplers with $k > 2$ should be accompanied by other developments, that could enhance its computational power, such as the inclusion of more sophisticated driver Hamiltonians.

d. Impact of the limited connectivity — Here we address the issues that occur with limited-connectivity hardware (see Fig. 3). From the physics scaling perspective, Fig. 5(a) and Fig. 6 show that there are no major effects in solving the problems with the QUBO or with the chimera representation. This seems to be a common feature across classical and quantum approaches. Following the argument just previously made in the case of the PUBO vs QUBO discussion, we show that $b_{\rm phys}$(Q) $\approx$ $b_{\rm phys}$(C) and $N_{\rm C} \sim \alpha_{\rm C\leftarrow Q} N_{\rm Q}$, implies that $b_{\rm app}$(C) $>$ $b_{\rm app}$(Q). Here, $\alpha_{\rm C\leftarrow Q} = 3.5026$ from Fig. 9. Although it has always been expected that more connectivity should be better, to the best of our knowledge this is the first demonstration that having a quantum annealing device with

more connectivity can have a significant impact when solving real-world applications. Our results, within the context of the CCFD application, show that the advantage here is not simply an overall prefactor improvement in the TTS but that an important asymptotic scaling advantage is expected as well. As a reminder to the reader, this *in-silico* advantage from SQA will be matched by a quantum hardware implementation only under the assumption that the asymptotic scaling of SQA "mimics" the performance of QA. As stated in the introduction, this is an unsettled question and beyond the scope of our work.

e. Comparison of QA with generic and tailored algorithms for CCFD — The main question that motivated this study was if QA can efficiently solve CCFD problems. Figures 5(b) and 6 show that from the application perspective the scaling of the DW2X quantum annealer and of any of the SQA variants considered here does not look favorable for QA. In fact, the DW2X does not even scale better than simulated annealing (SA/P). One of the major challenges for devices with a small finite graph degree connectivity (such as the DW2X with the chimera topology) is that to solve real-world applications it carries all the qubit overhead from the transformations PUBO to QUBO over to chimera. However, the application scaling can be improved. For example, reducing the number of qubits needed to represent the application would most likely improve the scaling performance. In Sec. C we present an alternative and more efficient mapping that we aim to explore in further studies. It is important to note that the new mapping improves the performance of SA and PTICM accordingly. Note that from all the approaches, the most efficient with the best scaling is a SAT-based solver developed by our team for this study which excels in this strong-fault model-based diagnosis of multiplier circuits. The SAT-based solver does not depend on any of the PUBO, QUBO or Chimera representation because it has the advantage of constructing its own variable representation and set of satisfiability constrains directly from the propositional logic level shown in panel (a) of Fig. 3. Although all other classical stochastic solvers used (SA, PTICM, and SQA) might work directly with the propositional logic as well, the evaluation of the cost function would be highly nonlocal compared to the evaluation of the difference in energy required for the Metropolis update in the case of the polynomial evaluation. By nonlocal we mean that if we were to work with only the fault variables and use the propositional logic instead of constructing the PUBO and including the internal wire variables, then we would need to propagate from inputs all the way through each gates and their health status assignment to obtain the predicted outputs and subsequently an effective energy that can be used in the Metropolis update. For every pair considered in an Metropolis update, the whole process need to be applied and later subtracted. In contrast, in any of the polynomial representation, because all the faults and wires variables are considered, the evaluation of the energy difference can be applied very efficiently by only considering the few terms that change the energy by the respective variable flip. Most importantly, because the main point of this contribution is to compare with algorithms that could be implemented in QA architectures, we did not explore this implementation. We leave it as an open question whether algorithms like SA can have a better scaling on that propositional logic representation.

IV. CONCLUSIONS

Regardless of the substantial efforts in benchmarking, the study of early generation quantum annealers has been done exclusively with synthetic spin-glass benchmarks. However, comprehensive studies comparing several quantum and classical algorithmic approaches, including state-of-the-art tailored solvers for real-world applications, had been missing. In this work we present a comprehensive benchmarking study on a concrete application, namely the diagnosis of faults in digital circuits, referred in the main text as CCFD. More specifically, we provide insights on the performance of QA in the context of the CCFD instances by performing an asymptotic scaling analysis involving five different approaches: QA experiments on the DW2X compared to three classical (SA, PTICM, and a CCFD-tailored SAT-based solver), and extensive QMC simulations, most of them on three different problem representations (PUBO, QUBO on the native CCFD graphs, and QUBO on the DW2X chimera topology), for instances of multiplier circuits of varying size. It is important to note that by asymptotic analysis we refer to conclusions drawn from the largest problem sizes accessible to us to experiment with in each of these approaches.

We have analyzed the problem with two foci: a *physics perspective* and an *application-centric perspective*. The emphasis of the physics perspective is similar to previous representative benchmark studies [8–20, 50], that aim at probing the computational resources of QA, and to answer questions such as whether it is even possible in synthetic data sets to prove an asymptotic quantum speedup, or to address the role of quantum tunneling, among other open questions in the field. Within our physics perspective we add several issues not thoroughly considered in other benchmark studies. For example, what is the impact in the computational scaling of solving the problem directly with Hamiltonians natively encoding many-body interactions beyond pairwise as those naturally appearing in real-world applications? What is the impact in the scaling from solving the problem instances on (hypothetical) physical hardware with different qubit connectivity constrains, e.g., by comparing the QA performance on connectivity graphs dictated by the CCFD instances and the minor-embedded representation in the DW2X Chimera graph?

From this physics perspective we show that our instances are hardest when compared to any of the proposed random spin-glass instances (see Fig.. 4). Intrinsic hardness is one of the long sought-after features when performing benchmark studies [11, 16, 20], therefore making our CCFD instances currently the best candidate for benchmarking the next generation of quantum annealers. In particular, because these problems stem from real-world applications, in contrast to random synthetic benchmarks on the native D-wave's chimera graph which have been dulled not only for giving an advantage to the hardware but also for lacking practical importance [52, 53].

We also address the question of whether SQA can repro-

duce the scaling of the DW2X for the CCFD application. Although the results in Fig. 8 might lead to the conclusion that clearly SQA has a different scaling than the DW2X, the fact that most likely the DW2X is running at a sub-optimal annealing time might be distorting the scaling and resulting in a better apparent scaling. More extensive studies with enough data points — where one can optimize for the optimal annealing time — might reveal the real scaling of the device. Although this is, in principle, feasible on quantum annealers, the main challenge might rely on SQA simulations which are already at the limit of what is computationally feasible. In Appendix D we discuss the apparent different scaling of SQA with a linear schedule compared to SQA with the same schedule as the D-Wave device and the challenges on drawing any meaningful conclusions about the difference in scaling between the DW2X and SQA. From the choice of schedule perspective, we find that within SQA as a solver, the linear schedule seems to be more efficient, but most likely not bringing any scaling advantage.

When compared on the same representation (either native-QUBO or Chimera-QUBO) we show that both, SQA and the DW2X have a limited quantum speedup by showing a scaling advantage over SA. We arrive at this conclusion assuming the DW2X scaling obtained here is not drastically affected by the non-optimal annealing time, which is very unlikely due to the large difference in the slopes between SA and SQA and DW2X. These results confirm the presence of quantum tunneling in the DW2X; a quantum speedup restricted to sequential algorithms [21] similar to the Google Inc. study on the weak-strong clusters instances [19]. One important highlight here is that ours is the first demonstration on instances generated from a concrete real-world application and where the multi-spin co-tunneling needs to happen more often on the strongly ferromagnetically coupled physical qubits encoding the logical units from the original QUBO problem. Although it is encouraging to see that such co-tunneling events seem to be happening in the hardware at the problem sizes considered here, the minor-embedding mapping logical variables into physical qubits in the hardware usually involves the generation of long "chains." Further studies need to be performed using larger instances to see if this advantage remains, and where longer "chains" with 10 or more qubits would be more frequent. The comparison against other generic solvers like PTICM or tailored solvers like the SAT-based solver developed here, were not favorable for our SQA simulations and DW2X experiments. It is important to consider that both, SQA simulations and DW2X experiments, were done with stoquastic Hamiltonians as the only ones available in current hardware. It is expected that non-stoquastic Hamiltonians will bring a boost in performance [54, 55], although it is an open question if they will have any asymptotic scaling advantage.

The application-centric perspective is more challenging and raises the bar significantly for quantum annealers. Here, we find that the tailored SAT-based algorithm performs best. We note that the performance is even better that the PTICM algorithm, which is currently the state-of-the-art in the field [56]. Although the results using quantum optimization approaches as seen from the application perspective are not that encouraging, our study suggests next steps to be taken in the field of quantum optimization. First, there is a clear need for higher-connectivity devices. Second, our SQA results suggest that adding higher-order qubit interactions [42, 57] to new hardware might require also the addition of more complex driving Hamiltonians.

This rather extensive study should be considered as a baseline for future application studies. We do emphasize, however, that the conclusions should be interpreted within the context of the particular CCFD application. Furthermore, the results are for the specific case of conventional QA with a transverse field driver. The poor performance of QA should be seen as an incentive for the community to address important missing ingredients in the search for quantum advantage for real-world applications. Other variable efficient mappings (as shown in Appendix C 2) could also provide a performance boost. We note that the latter should also provide an advantage for classical solvers, because larger systems could be studied. A detailed performance comparison of our CCFD benchmarks to other mapping strategies [29] will be done in a subsequent study.

Further adding other features, such as better control of the annealing schedules via "seeding" of solutions [58], and the subsequent developments of classical-quantum hybrid heuristic strategies [58–61] will likely lead to breakthroughs in quantum optimization. However, more simulations are needed to guide the design of new machines.

Acknowledgments

The work of A.P.O. was supported in part by the AFRL Information Directorate under grant F4HBKC4162G001, the Office of the Director of National Intelligence (ODNI), and the Intelligence Advanced Research Projects Activity (IARPA), via IAA 145483. Z.Z. and H.G.K. acknowledge support from the National Science Foundation (Grant No. DMR-1151387). The work of H.G.K. and Z.Z. is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via MIT Lincoln Laboratory Air Force Contract No. FA8721-05-C-0002. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, AFRL, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright annotation thereon. We thank Delfina Garcia-Pintos for the implementation of the libraries used in the bootstrapping analysis of the slopes and their confidence intervals of our asymptotic analysis. We also thank Tayo Oguntebi for initial support and discussions related to this CCFD application. We also thank Catherine McGeoch for feedback on an earlier version of this manuscript.

Appendix A: QA for combinatorial optimization problems

The quantum hardware employed consists of 144 unit cells with eight qubits each, as characterized in Refs. [8, 62]. Post-fabrication characterization determined that only 1097 qubits from the 1152 qubit array can be reliably used for computation, as shown in Fig. 7. The array of coupled superconducting flux qubits is, effectively, an artificial Ising spin system with programmable spin-spin couplings and magnetic fields. It is designed to solve instances of the following (NP-hard [63]) classical optimization problem: given a set of local longitudinal fields $\{h_i\}$ and an interaction matrix $\{J_{ij}\}$, find an assignment $\mathbf{s}^* = s_1^* s_2^* \cdots s_N^*$, that minimizes the objective function $E : \{-1, +1\}^N \to \mathbb{R}$, where

$$E(\mathbf{s}_{\rm C}) = \sum_{1\le i\le N} h_i s_i + \sum_{1\le i<j\le N} J_{ij} s_i s_j. \qquad \text{(A1)}$$

Here, $|h_i| \le 2$, $|J_{ij}| \le 1$, and $s_i \in \{+1, -1\}$. The subscript "C" is to emphasize that the spins are within the *chimera* graph, and to differentiate these from the other two representations studied in the paper at the PUBO ($\mathbf{s}_{\rm P}$) and QUBO ($\mathbf{s}_{\rm Q}$) level, respectively.

Finding the optimal set of variables $\mathbf{s}^*$ is equivalent to finding the ground state of the corresponding Ising classical Hamiltonian,

$$H_p = \sum_{1\le i\le N}^{N} h_i \sigma_i^z + \sum_{1\le i<j\le N}^{N} J_{ij} \sigma_i^z \sigma_j^z, \qquad \text{(A2)}$$

where σ_i^z is a Pauli z matrix acting on the ith spin.

Experimentally, the time-dependent quantum Hamiltonian implemented in the superconducting-qubit array via

$$H(\tau) = A(\tau) H_b + B(\tau) H_p, \qquad \tau = t/t_a, \qquad \text{(A3)}$$

with $H_b = -\sum_i \sigma_i^x$ the transverse-field driving Hamiltonian responsible for quantum tunneling between the classical states constituting the computational basis, which is also an Eigenbasis of H_p. The time-dependent functions $A(\tau)$ and $B(\tau)$ are such that $A(0) \gg B(0)$ and $A(1) \ll B(1)$. In Fig.8(a), we plot these functions as implemented in the experiment. t_a denotes the time elapsed between the preparation of the initial state and the measurement, referred to hereafter as the *annealing time*.

QA as an algorithmic strategy to solve classical optimization problems exploits quantum fluctuations and the adiabatic theorem of quantum mechanics. This theorem states that a quantum system initialized in the ground state of a time-dependent Hamiltonian remains in the instantaneous ground state if the Hamiltonian changes sufficiently slow. Because the ground state of H_p encodes the solution to the optimization problem, the idea behind QA is to adiabatically prepare this ground state by initializing the quantum system in the easy-to-prepare ground state of H_b, which corresponds to a superposition of all 2^N states of the computational basis, and then slowly interpolating to the problem Hamiltonian, $H(\tau = 1) \approx H_p$.

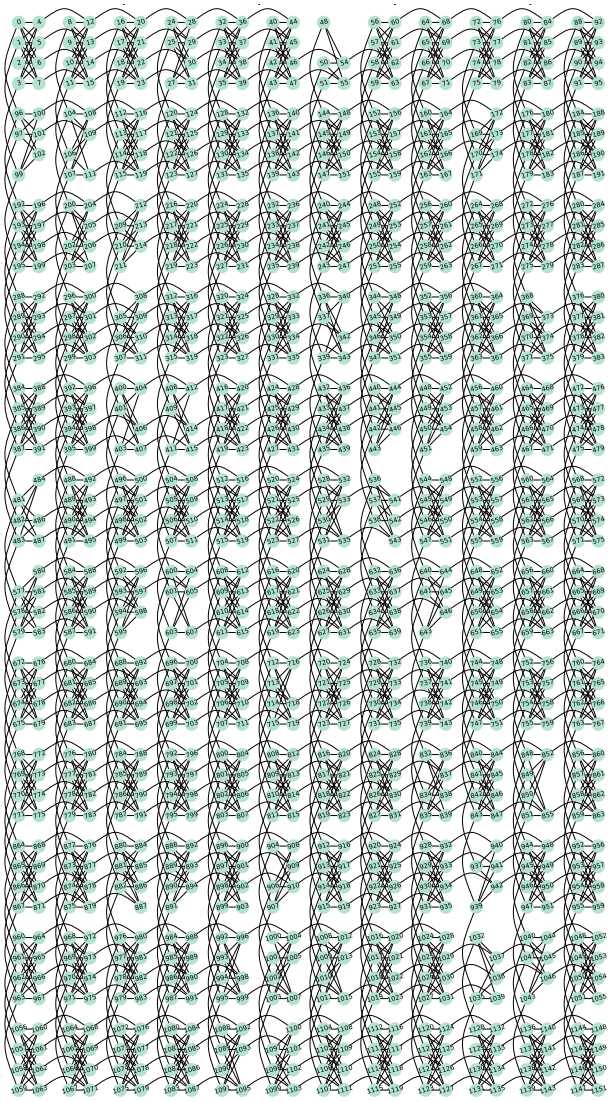

FIG. 7: Device architecture and qubit connectivity. The array of superconducting quantum bits is arranged in 12×12 unit cells that consist of 8 quantum bits each. Within a unit cell, each of the 4 qubits on the left-hand partition (LHP) connects to all 4 qubits on the right-hand partition (RHP), and vice versa. A qubit in the LHP (RHP) also connects to the corresponding qubit in the LHP (RHP) of the units cells above and below (to the left and right of) it. Edges between qubits represent couplers with programmable coupling strengths. We show only the 1097 functional qubits out of the 1152 qubit array.

In a realistic experimental implementation, the quantum processor will operate at a finite temperature, and in addition to thermal fluctuations, other types of noise are unavoidable, leading to dissipation processes not captured in $H(t)$. Deviations from adiabaticity affecting the performance of the quantum algorithm seem to be a delicate balance between the quantum coherence effects and the interaction with the environment, responsible for, e.g., thermal excitation (relaxation) processes out of (into) the ground state [23, 64].

Determining the optimum value of t_a is an important and nontrivial problem in itself. To the best of our knowledge this

question related to the scaling of t_a in a noisy environment is still largely unexplored, with progress only in the case of canonical models [65]. From an experimental standpoint, the main limitation is the limited size of the available quantum devices, but now with new generation of devices with more than 2000 qubits the question is within reach in the case of synthetic data sets [50]. Studying this question within the context of real-world applications is now within reach. We will leave this study for future work.

Appendix B: Methods

a. Simulated Quantum Annealing (SQA) — QMC simulations are performed using a variant of the continuous time QMC algorithm [33], which we refer to here and in the main text as SQA. We build clusters in the imaginary time direction in the same way as done in Ref. [33]. However, because here we study frustrated systems, we do not build clusters in the spatial directions. To flip segments of finite imaginary time extent, we use the Metropolis algorithm [66]. This algorithm was also used in previous benchmark studies of the D-Wave devices [12, 13].

In the SQA simulations we use a linear schedule. We fix the diagonal interaction strength $B(\tau) = 1$ and vary the transverse field strength as $A(\tau) = \Gamma_0(1 - \tau)$, where τ is the annealing time and Γ_0 is the initial transverse field strength; see Fig. 8. We use different values of Γ_0 for different problem representations. $\Gamma_0 = 0.8$ for PUBO, $\Gamma_0 = 1.6$ for QUBO, and $\Gamma_0 = 6$ for instances on the chimera graph. These representations are referred as "P," "Q," and "C," respectively, in the main text. In addition, we also implement the $A(\tau)$ and $B(\tau)$ annealing schedule used in the DW2X device, as depicted in Fig. 8.

*b. Estimation of the time-to-solution (*TTS*)* — For stochastic algorithms, the time-to-solution depends on the desired confidence, i.e., the probability P required such that the solver produces the target solution. For example, in all previous studies the level of certainty required from the solver was 99%, i.e., $P = 0.99$, and the relevant metric, denoted R_{99} is the number of repetitions needed such that the probability that the solution is found is at least once is 99%. Let us denote by $p_{\rm s}$ the success probability to obtain the target solution in a single execution or repetition of the solver. Because the probability F of *not* observing the solution after R_P repetitions is $F = (1 - p_{\rm s})^{R_P} = 1 - P$, the number of repetitions R_{99} needed to obtain the desired solution with probability at least 99% is

$$R_{99} = \lceil \frac{\log(1 - 0.99)}{\log(1 - p_{\rm s})} \rceil. \tag{B1}$$

Therefore, the time-to-solution (TTS) under this criteria is the product of R_{99} times the time it takes to perform one execution or each repetition, $t_{\rm rep}$:

$$\mathrm{TTS} = \mathrm{t_{rep}R_{99}}. \tag{B2}$$

For the DW2X, $t_{\rm rep}$ was set to the annealing time of 5 μs. For SA, QMC, and PTICM, $t_{\rm rep}$ can be estimated as:

$$t_{\rm rep} = t_{\rm su} N \mathrm{MCS_{opt}}, \tag{B3}$$

with $\mathrm{MCS_{opt}}$ the optimal number of Monte Carlo sweeps (MCS), i.e., the number of MCS that minimizes the TTS and $t_{\rm su}$ the time it takes to make a MC spin update. In the case of PTICM, the values of $\mathrm{MCS_{opt}}$ include a factor of 120 coming from the 4 replicas and 30 temperatures considered in our implementation. Additionally, we multiply by a factor of 1.2 to account an estimated 20% overhead coming from other steps in the PT implementation and not present in SA, such as swaps of configurations and cluster updates. Here we optimize the TTS per instance to obtain the best scaling for each algorithm. The computational effort of the algorithm optimization of this procedure compared to optimizing over different annealing times as proposed in Ref. [13] should yield comparable scaling results. We prefer this approach because it required the same computational effort as analyzing the data at different annealing times and it provides more reliable information on the intrinsic difficulty of each instances. For example, in the limit of very large annealing time, where all the instances have probability one, most instances have the same computational effort and it is not possible to identify which instances are intrinsically harder. This is related to the problem with reported DW2X scaling for small instances, for which the minimum available annealing time is greater than optimal. Note that $p_{\rm s}$ is a function of the number of MCS, in the same way that it is a function of the annealing time in the case of the DW2X. Therefore, we estimate $p_{\rm s}$ for different values for the number of MCS, calculate R_{99} and from the values considered we select the optimum. Since one MCS involves N updates [67], with N the total number of spins in the problem, then to calculate the computational effort we need to multiply by N and by the effective time it takes to perform and evaluate each of these updates. The value $t_{\rm su}$ is different for each of the algorithms (e.g., SA vs SQA) and for each of the different representations (QUBO, PUBO, or chimera). The times estimated and used for the case of the CCFD instances are: $t_{\rm su}^{\rm SA/P} = t_{\rm su}^{\rm PTICM/P} = 5.5$ ns, $t_{\rm su}^{\rm SA/Q} = t_{\rm su}^{\rm PTICM/Q} = 3.42$ ns, $t_{\rm su}^{\rm SA/C} = t_{\rm su}^{\rm PTICM/C} = 2.6$ ns, $t_{su}^{\rm SQA_{ls}/P} = 1.08\mu$s, $t_{su}^{\rm SQA_{ls}/Q} = 1.88\mu$s, $t_{su}^{\rm SQA_{ls}/C} = 1.81\mu$s, $t_{su}^{\rm SQA_{dws}/C} = 48.8\mu$s. These times were used for all figures with the exception of Fig. 4 and Fig. 8. For the case of Fig. 4, and to give the best performance for each data set, we optimized for R_{99} as described above for every instance of each of the CCFD and random spin-glass data sets. Given that all the data sets were run with PTICM and under the same computational resources, we plotted directly the wall-clock time required after the aforementioned optimization of $MCS_{\rm opt}$.

To capture the computational scaling of SQA as a simulator of a hypothetical quantum annealer [denoted as SQA(q)] we used the same optimal values of $\mathrm{MCS_{opt}}$ used for SQA but we do not multiply by the factor of N. In this way we take into account the intrinsic parallelism of quantum annealers. The prefactor $t_{\rm su}$ is changed as well to an arbitrary constant parameter, denoted $t_{\rm SQA(q)}$ we can tune to make all the lines

in Fig. 8 to have a similar TTS as that value obtained by the DW2X device. The values used here were $t_{\mathrm{SQA(q)_{ls}}} = 5$ ns and $t_{\mathrm{SQA(q)_{dws}}} = 1.3$ ns.

Because the SAT-based solver described below is significantly different from the other stochastic solvers mentioned above, to estimate the TTS we run the SAT-solver 1000 times per instance and compute the TTS for each run. From this distribution of TTS values, we pick the 99% percentile as the TTS value we report since it matches the definition of the time needed to observe the desired solution

c. SAT-based solver tailored for CCFD — The SAT-based model-based diagnosis solver is implemented as follows. First it adds a tree-adder to the fault-augmented circuit to enforce the cardinality of the fault. Second, the formula is converted to Conjunctive Normal Form (CNF). Finally, a SAT solver is called n times, first for computing all zero-cardinality faults, then for all single-faults, etc., until a fault of cardinality n is found. For our implementation we have used the highly-optimized SAT-solver LINGELING [56]. It is a deterministic SAT-solver that uses Boolean search enhancements, including symbolic optimization, occurrence lists, literal stack, and clause distillation, etc.

d. DW2X programming details — When programming a quantum annealer to solve real-world applications, the process of minor-embedding introduces many other parameters that do not exist when benchmarking QA with a random spin-glass benchmark. One common misconception is that implementing real-world applications is harder because of the minor-embedding procedure. Although more efficient embedding strategies are always desirable, we want to emphasize here that it is not the main challenge when programming the device since heuristic algorithms solve this problem reasonably well [45]. It is also important to note here that the NP-hardness of finding the smallest minor-embedding (with respect to number of qubits) is largely moot, because the smallest minor-embedding is often far from optimal in terms of performance. For example, from our experience, sometimes it is preferable to have an embedding that uses more physical qubits but that has shorter "chains" representing logical variables. In our work we generated 100 embeddings per instance regardless of the problem size.

The main challenge (the *curse of limited connectivity* [68, 69] due to quantum annealers having a bounded number of couplers per qubit) does not lie in the minor embedding problem, but rather in the setting of the additional parameters once the minor-embedding has been chosen. Although proposals exist to cope with this challenge [46, 70, 71], the optimal setting of parameters is a largely open problem and one of the most important ones affecting the performance of quantum annealers as optimizers [46]. In this work, we use the strategy proposed in Ref. [46] to set the strength J_{F} of the ferromagnetic couplers, which enforce the embedding, and for gauge selection.

In addition to setting J_{F}, we must also distribute the logical biases $\{h_i\}$ and couplings $\{J_{i,j}\}$ over the available physical biases and couplings $\{\tilde{h}_k\}$ and $\{\tilde{J}_{k,l}\}$. The key consideration in parameter setting is the noise level of the programmable parameters of the quantum device. The noise margin of the D-Wave 2X machine is $\tilde{h}_j < 0.05$ for biases and $\tilde{J}_{k,l} < 0.1$ for couplers in a normalized, hardware-embedded problem, with the difference due to the difference in dynamic range.

We aim to divide the logical parameters as much as possible over the corresponding physical parameters subject to this precision limit using the following heuristic, which is similar to but distinct from that of Ref. [71]. Consider a logical bias h_i that corresponds to N_i hardware qubits. If h_i/N_i is greater than the 0.05 noise threshold, then each physical qubit j is given a bias $\tilde{h}_j = h_i/N_i$. If not, we consider the n_i hardware qubits within the chain that have nonzero inter-chain couplings. If h_i/n_i is greater than the threshold, we evenly distribute the logical bias amongst these n_i physical qubits. Finally, if neither of these strategies exceed the threshold, we assign the logical bias completely to hardware qubits with the lowest number of intra-chain couplings, breaking ties uniformly at random. The remaining hardware qubits within the chain are given a bias of zero. We distribute the logical couplers $J_{i,j}$ in a similar way. Suppose that the chains for logical qubits i and j have $N_{i,j}$ physical couplers between them. If $J_{i,j}/N_{i,j}$ is greater than 0.1 (noise threshold), we evenly distribute the logical coupling amongst the $N_{i,j}$ available physical couplers. Otherwise, the logical coupling is completely assigned to a single physical coupler uniformly at random from the $N_{i,j}$ options.

Appendix C: Mapping of minimal-cardinality fault diagnosis for combinational digital circuits to PUBO and QUBO

In this section we describe in detail two mappings of the fault diagnosis problem to QUBO, via a mapping to PUBO. The original instance consists of a set of m gates, each with a specified hard fault model. Excluding the inputs and outputs to the circuit, let $\mathbf{x} = (x_i)_{i=1}^{n} \in \{0,1\}^n$ indicate the value on every wire in the circuit. For gate i, let $\mathbf{y}_i \in \{0,1\}^*$ be the values of the input wires and z_i the value of the output wire. These are not new variables but rather alternative ways of referring to the variables $\mathbf{x}$. For example, if wire i is the output of gate j and the first input into gate k, then x_i, z_j, and $y_{k,1}$ all refer to the same variable. Let $g_i(\mathbf{y}_i) \in \{0,1\}$ be the Boolean function indicating the action of gate i, and $F_i(\mathbf{y}_i, z_i) \in \{0,1\}$ be the predicate indicating whether the combined input $\mathbf{y}_i$ and output z_i are consistent with the fault model for gate i. Several examples for g_i and F_i are given in Tables I and II, respectively.

Bian *et al.* [29] have also used fault diagnosis as a test bed for benchmarking novel techniques in QA. They used Satisfiability Modulo Theory to automatically generate functions representing the cost function and constraints, whereas here we do so manually, as described in this section. Their approach is further differentiated from the present one by their use of problem decomposition and locally-structured embedding.

Note that we describe the mapping to pseudo-Boolean polynomials over variables taking the values $\{0,1\}$, while the Hamiltonians in physical quantum annealers directly represent functions of variables taking the values ± 1, i.e., Ising spins. The two representations are equivalent with the follow-

TABLE I: Example gates and their representation as polynomials. For details see the main text.

Gate	$g_i(\mathbf{y}_i)$
OR	$y_{i,1} + y_{i,2} - y_{i,1}y_{i,2}$
AND	$y_{i,1}y_{i,2}$
XOR	$y_{i,1} + y_{i,2} - 2y_{i,1}y_{i,2}$
EQ	$1 - y_{i,1} - y_{i,2} + 2y_{i,1}y_{i,2}$
BUFFER	$y_{i,1}$
NOT	$1 - y_{i,1}$
NOR	$1 - y_{i,1} - y_{i,2} + y_{i,1}y_{i,2}$
NAND	$1 - y_{i,1}y_{i,2}$

TABLE II: Example fault models and their predicates as polynomials. For details see the main text.

Fault model	$F_i(\mathbf{y}_i, z_i)$
Stuck at 1	z_i
Stuck at 0	$1 - z_i$
Stuck at 0 or 1	1
Stuck at first input	$\mathrm{EQ}(z_i, y_{i,1})$
Stuck at first input or 0	$1 - y_{i,1}(1 - z_i)$

ing transformation:

$$b = (1 - s)/2, \qquad s = 1 - 2b, \tag{C1}$$

for $b \in \{0, 1\}$ and $s \in \{\pm 1\}$, with the latter being the conventionally used for physical implementations on quantum annealers, as in, e.g., Eq. (A1). Note that the substitutions leave the degree and connectivity of the polynomials unchanged.

1. Explicit mapping

For each gate i, introduce an additional variable f_i that indicates whether or not that gate is faulty. Assuming that $\mathbf{f} = (f_i)_{i=1}^{N_{\mathrm{gates}}}$ is consistent with $\mathbf{x}_i$, the number of faults is simply

$$H_{\mathrm{numfaults}}(\mathbf{f}) = \sum_{i=1}^{N_{\mathrm{gates}}} H^{(i)}_{\mathrm{numfaults}}(f_i) = \sum_{i=1}^{N_{\mathrm{gates}}} f_i. \tag{C2}$$

The consistency with the fault model is enforced by the penalty function

$$\begin{aligned} H_{\mathrm{faultset}}(\mathbf{x}, \mathbf{f}) &= \sum_{i=1}^{N_{\mathrm{gates}}} H^{(i)}_{\mathrm{faultset}}(\mathbf{y}_i, z_i, f_i), \\ H^{(i)}_{\mathrm{faultset}}(\mathbf{y}_i, z_i, f_i) &= \lambda^{(i)}_{\mathrm{faultset}} f_i \left[1 - F_i(\mathbf{y}_i, z_i)\right]. \end{aligned} \tag{C3}$$

Finally, we must also constrain the system to the appropriate behavior when there is no fault:

$$\begin{aligned} H_{\mathrm{gate}}(\mathbf{x}, \mathbf{f}) &= \sum_{i=1}^{N_{\mathrm{gates}}} H^{(i)}_{\mathrm{gate}}(\mathbf{x}, \mathbf{f}), \\ H^{(i)}_{\mathrm{gate}}(\mathbf{y}_i, z_i, f_i) &= \lambda^{(i)}_{\mathrm{gate}}(1 - f_i)\mathrm{XOR}[g_i(\mathbf{y}_i), z_i]. \end{aligned} \tag{C4}$$

The overall cost function is

$$\begin{aligned} H(\mathbf{x}, \mathbf{f}) &= H_{\mathrm{numfaults}}(\mathbf{f}) + H_{\mathrm{faultset}}(\mathbf{x}, \mathbf{f}) + H_{\mathrm{gate}}(\mathbf{x}, \mathbf{f}) \\ &= \sum_{i=1}^{N_{\mathrm{gates}}} H^{(i)}(\mathbf{y}_i, z_i, f_i), \end{aligned} \tag{C5}$$

where

$$\begin{aligned} H^{(i)}(\mathbf{y}_i, z_i, f_i) = H_{\mathrm{numfaults}}(f_i) + H_{\mathrm{faultset}}(\mathbf{y}_i, z_i, f_i) \\ + H_{\mathrm{gate}}(\mathbf{y}_i, z_i, f_i). \end{aligned} \tag{C6}$$

Note that, in general, this function is quartic. Using two ancilla bits per gate, the usual gadgets [22, 72] can be used to reduce this to quadratic as needed. Depending on the circuit, some ancilla bits may be reused to reduce the degree of the terms corresponding to more than one gate. For example, if the input $\mathbf{y}_i = (y_{i,1}y_{i,2})$ to gate i happens to be the same input to another gate j, then a single ancilla bit corresponding to $y_{i,1}y_{i,2}$ may be used for both gates. In this work, we use exactly two ancilla bits per gate, corresponding to the conjunctions $y_{i,1}y_{i,2}$ and $z_i f_i$.

The explicit mapping is easily extended to the case of $\nu > 1$ input-output pairs. Instead of the single $\mathbf{x}$, we have a copy $\mathbf{x}_\iota$ for each input-output pair, and use a single set of shared fault variables $\mathbf{f}$. $H_{\mathrm{numfaults}}$ remains exactly the same as above, while now there are copies of H_{faultset} and H_{gate} for each input-output pair:

$$\begin{aligned} H^{(i)}_{\mathrm{faultset}}(\mathbf{y}_i, \mathbf{z}_i, f_i) &= \sum_{\iota=1}^{\nu} H^{(i,\iota)}_{\mathrm{faultset}}(\mathbf{y}_{i,\iota}, z_{i,\iota}, f_i), \\ H^{(i,\iota)}_{\mathrm{faultset}}(\mathbf{y}_{i,\iota}, z_{i,\iota}, f_i) &= \lambda^{(i)}_{\mathrm{faultset}} f_i \left[1 - F_i(\mathbf{y}_{i,\iota}, z_{i,\iota})\right]; \end{aligned} \tag{C7}$$

and

$$\begin{aligned} H^{(i)}_{\mathrm{gate}}(\mathbf{y}_i, \mathbf{z}_i, f_i) &= \sum_{\iota=1}^{\nu} H^{(i,\iota)}_{\mathrm{gate}}(\mathbf{y}_{i,\iota}, z_{i,\iota}, f_i), \\ H^{(i,\iota)}_{\mathrm{gate}}(\mathbf{y}_{i,\iota}, z_{i,\iota}, f_i) &= \lambda^{(i)}_{\mathrm{gate}}(1 - f_i)\mathrm{XOR}\left[g_i(\mathbf{y}_{i,\iota}), z_{i,\iota}\right]; \end{aligned} \tag{C8}$$

where $\mathbf{y}_{i,\iota}$ and $z_{i,\iota}$ are input and output bits for gate i in $\mathbf{x}_\iota$, and $\mathbf{y}_i$ and $\mathbf{z}_i$ contain all ν copies thereof.

The explicit mapping is also easily extended further to the case of $\mu > 1$ fault modes. For each gate i, we use μ fault variables $\mathbf{f}_i = (f_{i,\alpha})_{\alpha=1}^{\mu}$, corresponding to the fault modes $(F_{i,\alpha})_{\alpha=1}^{\mu}$. Considering $f_i = \sum_{\alpha=1}^{\mu} f_{i,\alpha}$ as a function of $\mathbf{f}_i$ (rather than a separate bit on its own), $H_{\mathrm{numfaults}}$ and H_{gate}

remain unchanged from the single-fault case, even with multiple input-output pairs. Now there are μ copies of $H_{\rm faultset}$:

$$H^{(i)}_{\rm faultset}(\mathbf{y}_i, z_i, \mathbf{f}_i) = \sum_{\iota=1}^{\nu}\sum_{\alpha=1}^{\mu} H^{(i,\iota,\alpha)}_{\rm faultset}(\mathbf{y}_{i,\iota}, z_{i,\iota}, \mathbf{f}_{i,\alpha}),$$
$$H^{(i,\iota,\alpha)}_{\rm faultset}(\mathbf{y}_{i,\iota}, z_{i,\iota}, \mathbf{f}_{i,\alpha}) = \lambda^{(i)}_{\rm faultset} f_{i,\alpha}\left[1 - F_{i,\alpha}(\mathbf{y}_{i,\iota}, z_{i,\iota})\right]. \tag{C9}$$

Finally, to penalize situations in which more than one fault bit is set per gate, we add

$$H^{(i)}_{\rm multfault}(\mathbf{f}_i) = \lambda^{(i)}_{\rm multfault}\sum_{\alpha=1}^{\mu-1}\sum_{\beta=\alpha+1}^{\mu} f_{i,\alpha}f_{i,\beta}. \tag{C10}$$

So long as $\lambda^{(i)}_{\rm multfault} > \nu\lambda^{(i)}_{\rm gate}$, $H_{\rm multfault}$ will outweigh the potentially negative $H_{\rm gate}$ as needed. For each gate i, $\nu(1+\mu)$ ancilla bits suffice, corresponding to the conjunction of the bits $\mathbf{y}_{i,\iota}$ for each input-output pair ι and to the conjunction $z_{i,\iota}f_{i,\alpha}$ for every ι and mode α.

When the fault modes considered are simply stuck at 1 or stuck at 0, i.e. $F_i(\mathbf{y}_i, z_i) = F_i(z_i) = z_i$ or $1-z_i$, respectively, we can use the alternative

$$H^{(i,\iota)}_{\rm gate} = \lambda^{(i)}_{\rm gate}\left\{1 + f_i[1 - 2F_i(z_{i,\iota})]\right\}\mathrm{XOR}[g_i(\mathbf{y}_{i,\iota}), z_{i,\iota}], \tag{C11}$$

where $f_i = \sum_{\alpha=1}^{\mu}$ as before. When F_i is linear in z_i, this expression is quadratic in g_i, z_i, and f_i, so that it suffices to reduce g_i to linear using a single ancilla bit corresponding to the conjunction of the input bits $\mathbf{y}_i$. Overall, only ν ancilla bits are needed per gate.

2. Implicit mapping

Having the fault bits $\mathbf{f}$ are not necessary. Here we show how to construct the requisite energy functions using just the wire bits $\mathbf{x}$. Note that $H_{\rm faultset}$ was used only to enforce consistency of the fault bits with the wire bits, and so is obviated by the omission of the former. Recall that we would like to find the assignment of values to the wires that minimizes the number of faults while being consistent with the nominal gates and fault models. Therefore, we need a function $H^{(i)}_{\rm numfaults}$ that is zero when $z_i = g_i(\mathbf{y}_i)$ and is one when $z_i \neq g_i(\mathbf{y}_i)$ *and* $F_i(\mathbf{y}_i, z_i)$. Its behavior when $z_i \neq g_i(\mathbf{y}_i)$ and not $F_i(\mathbf{y}_i, z_i)$ only need be non-negative; penalizing that case is left to $H_{\rm gate}$. The following meets our needs:

$$\begin{aligned} H_{\rm numfaults}(\mathbf{x}) &= \sum_{i=1}^{N_{\rm gates}} H^{(i)}_{\rm numfaults}(\mathbf{y}_i, z_i) \\ &= \sum_{i=1}^{N_{\rm gates}} F_i(\mathbf{y}_i, z_i)\mathrm{XOR}[g_i(\mathbf{y}_i), z_i]. \end{aligned} \tag{C12}$$

To penalize the case when the output z_i of gate i is inconsistent with the input $\mathbf{y}_i$ but not in a way allowed by the fault model, we use

$$H^{(i)}_{\rm faultset}(\mathbf{y}_i, z_i) = \lambda^{(i)}_{\rm gate}[1 - F_i(\mathbf{y}_i, z_i)]\mathrm{XOR}[g_i(\mathbf{y}_i), z_i]. \tag{C13}$$

The overall energy function for each gate is simply

$$H^{(i)}(\mathbf{y}_i, z_i) = H^{(i)}_{\rm numfaults}(\mathbf{y}_i, z_i) + H^{(i)}_{\rm gate}(\mathbf{y}_i, z_i). \tag{C14}$$

Each $H^{(i)}$ is cubic, and can be reduced to quadratic using a single ancilla bit. As with the explicit mapping, in certain cases a single ancilla may be shared among multiple gates.

For a single input-output pair, the implicit mapping naturally generalizes to multiple fault modes, by considering a combined fault mode that is the conjunction of the multiple ones, i.e., using $F_i = \mathrm{OR}(F_{i,1}, \ldots, F_{i,\mu})$. Some examples, e.g., stuck at one or first input, are shown in Table II. This does not apply to multiple input-output pairs because it does not enforce that all copies are subject to the *same* fault mode. For particular gates and sets of fault models, it is likely most efficient to use a modification of the explicit mapping, as shown for the stuck at 0 and stuck at 1 cases above.

3. Logical penalty weights

Without loss of generality, in this work we have chosen only one penalty weight λ for both $\lambda^{(i)}_{\rm gate}$, which penalizes a mismatch between the input and output of a gate in the absence of a fault, and $\lambda^{(i)}_{\rm faultset}$, which enforces the fault model. That is, $\lambda^{(i)}_{\rm gate} = \lambda^{(i)}_{\rm faultset} = \lambda$ for all i. Setting $\lambda = N_{\rm gates} + 1$ suffices to guarantee that the global minima correspond to a valid diagnosis, i.e., those solutions $(\mathbf{x}, \mathbf{f})$ such that $H_{\rm gate}(\mathbf{x}, \mathbf{f}) = H_{\rm faultset}(\mathbf{x}, \mathbf{f}) = 0$. Any valid diagnosis has energy $H(\mathbf{x}, \mathbf{f}) = H_{\rm numfaults}$ at most $N_{\rm gates}$, so any violation of the constraints incurring a penalty at least $\lambda = N_{\rm gates} + 1$ yields a total energy greater than that of any valid diagnosis.

A weaker condition to require of the penalty weight λ is simply that the ground state of H is a valid diagnosis. That is, an invalid state (i.e., one that violates at least one of the model constraints) may have lower total energy than *some* valid state, but not than a *minimum*-fault valid state. One simple upper-bound on the minimum number of faults is the number of outputs, which thus also serves as a sufficient lower bound on λ. In the case of the multiplier circuits with k-bit and l-bit inputs, the length of the outputs in simply $k + l$ bits, which is much smaller than $N_{\rm gates}$.

Nevertheless, a much lower value of λ may suffice in practice for a particular set of instances. It is desirable to use the smallest λ possible, because when the coefficients of the Hamiltonian are rescaled for a hardware implementation, larger values of λ lead to higher precision requirements, which may not be met by limited-precision devices. For the generation of the PUBO expressions in the circuits considered here, up to mult8-8, we used a value of $\lambda = 4$, regardless of the size of the circuit. With the help of the complete SAT-based solver, we checked that, for all the instances studied here, this

value sufficed to ensure that the ground state corresponds to a valid diagnosis.

However, we did generate observations, not included in this study, for which $\lambda = 4$ was insufficient. This was extremely rare, from no such examples in the smaller circuits to at most one in five hundred for the largest circuits. Because we used the first hundred randomly generated instances for each size, $\lambda = 4$ sufficed for every instance used; this was highly likely though not guaranteed.

A more common event that we had to filter in the instance generation was the appearance of random instances where the minimal solution contains no faults. These are easy to eliminate since one can easily verify whether the output corresponds to the multiplication of the inputs and therefore the solution to our problem is trivial with a minimal fault cardinality of zero. It is interesting to note that in diagnosis task such instances are still valuable, since one considers not only the minimal cardinality but also the runners up could provide valuable information about the circuit. For example, it could be the case that there is indeed a fault in the circuit but the output observations still match the desired output, but the fault can only be unmasked for example, by using another observation in the circuit. The problem of selecting the best inputs to probe faults in circuits is another interesting NP-hard problems in its own. We focus here in the minimal cardinality case, given an input/output pairs.

4. PUBO to QUBO reduction

The cost function of the CCFD problem is initially expressed as a pseudo-Boolean expression (i.e., PUBO) of degree greater than two. We then transform the higher-degree PUBO expression into a quadratic one by using a conjunction gadget. The conjunction gadget introduces an ancilla bit $q_{i,j}$ that corresponds to a conjunction of two bits q_i and q_j in the PUBO, replaces all occurrences of the $q_i q_j$ with $q_{i,j}$, and adds a penalty function so that in any ground state of the QUBO expression the ancilla bit is appropriately set, $q_{i,j} = q_i q_j$. We use the penalty function [22, 72, 73]

$$H_{\text{ancilla}} = \delta(3q_{i,j} + q_i q_j - 2q_i q_{i,j} - 2q_j q_{i,j}), \qquad \text{(C15)}$$

which is zero when $q_{i,j} = q_i q_j$ and at least δ otherwise, where $\delta > 0$ is the penalty weight. The penalty weight δ needs to be large enough that states violating the ancilla constraint have energy much larger than the ground energy of the original PUBO expression, thus preserving the low-energy spectrum. As with the logical penalty λ, we would like δ to be as small as is necessary in order to minimize the precision needed to implement the cost function on a hardware device. For each logic gate in the CCFD problem, we determined that the following values to be best, as a multiple of the logical penalty weight λ:

$$\delta_{\text{AND}} = \delta_{\text{OR}} = 2.5\lambda;\ \delta_{\text{XOR}} = 2\lambda. \qquad \text{(C16)}$$

This controlled and optimized assignment of contraction penalties per logic gate is one of the remarkable features of this CCFD applications in contrast to others, where penalties can not only be higher but also scale with the number of variables [22, 23]. In this case, the penalties are independent of the circuit size.

Appendix D: SQA versus DW2X

Here we address in more details the question of whether SQA has the same scaling as the DW2X device and the comparison of the two schedules used in the SQA simulations. The linear schedule tends to underestimate the scaling exponents for easy problems and small systems sizes, when the required number of sweeps is small. This is because there might not be enough QMC time to remove the segments in the imaginary time direction that have different spin values.

For the DW schedule (dws) we cut the first 10% of the schedule. First, the initial part of the D-Wave schedule is not necessary because it is very easy to equilibrate QMC when the transverse field strength is large enough. Second, that leads to shorter simulation times as it takes roughly the same time to run the first 10% of the schedule as to run the rest of the schedule. This is because the SQA simulation time is roughly proportional to the transverse field strength and, in the first part of the schedule, the transverse field is largest. Strictly speaking, one probably can cut more than 10% of the initial schedule. One can also cut some fraction of the schedule at the end, but that will not improve simulation times significantly. However, that could lead to a different scaling for easy problems and small small problem sizes. This difference in scaling then could be fictitious and it might even disappear for larger systems sizes.

Therefore, it is difficult to make any conclusive statements about the apparent difference in scaling and significant further work is required to address this issue with more certainty. Besides emphasizing that such comparison are not straightforward, these further simulations and parameter fine-tuning is beyond the scope of this work.

Note that the two statements are not contradictory with our statements about limited quantum speedup in Sec. III C. If we had unlimited computational resources we expect the SQA slopes to come smaller in value, while in the case of the DW2X we expect that optimization of the annealing time would lead to a larger slope values compare to the current one. Although we declare the results of SQA vs DW2X inconclusive given the these two slopes might reach comparable values, given the expectation for SQA towards improving its scaling, these observation makes our claims about limited quantum speedup even stronger.

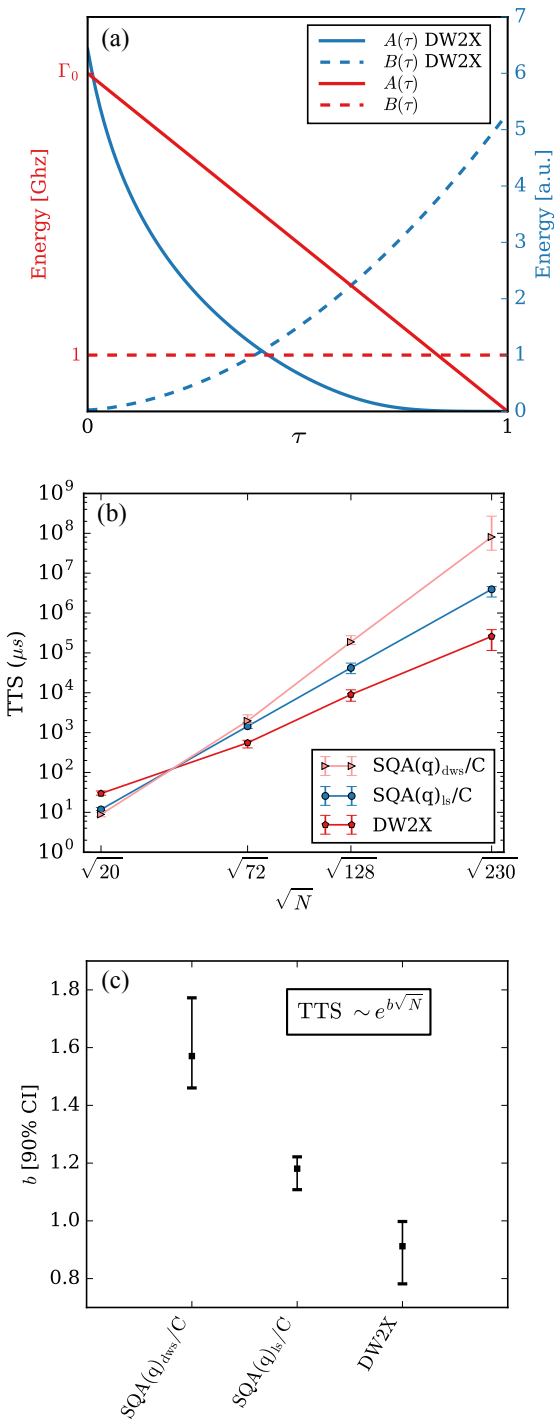

FIG. 8: (a) Details for the different annealing schedules used in this work. Panels (b) and (c) show a comparison of the DW2X experimental results and SQA simulations of hypothetical QA devices with a DW2X-like [SQA(q)$_{\rm dws}$] and with a linear annealing schedule [SQA(q)$_{\rm ls}$]. Data points correspond to the median values extracted from a bootstrapping analysis from 100 instances per problem size, with error bars indicating the 90% confidence intervals (CI).

Appendix E: Qubit resources for numerical simulation and experiments

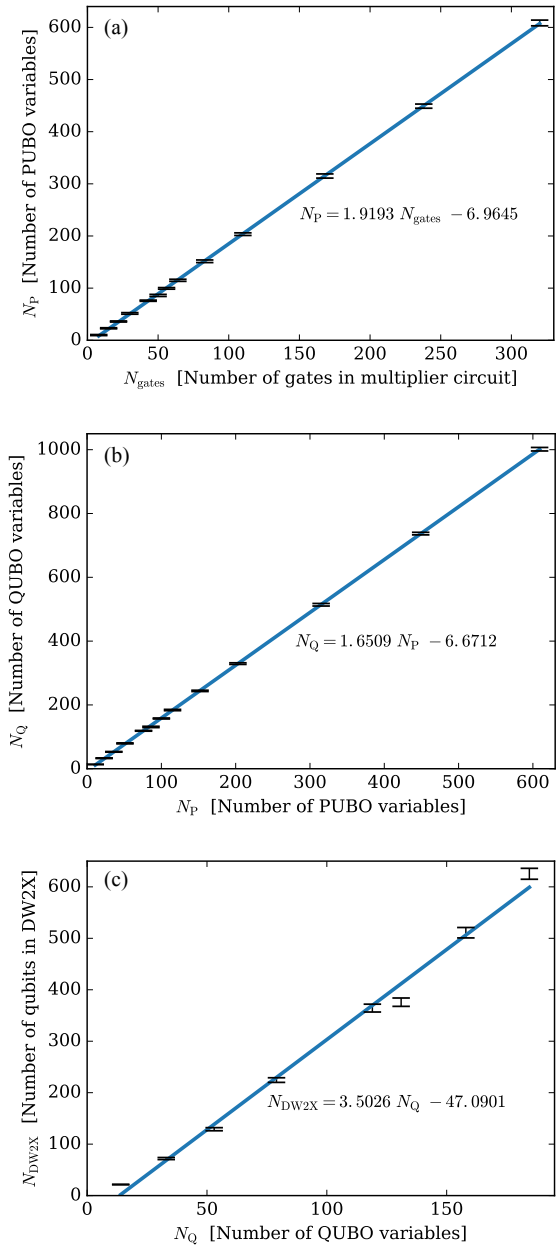

FIG. 9: Qubit resources for each of the problem representation [(a) PUBO, (b) QUBO and (c) chimera (DW2X)] considered in our benchmarking study of the CCFD instances. Data points correspond to the median values extracted from a bootstrapping statistical analysis from 100 instances per problem size, with error bars indicating the 90% confidence intervals (CI).

Appendix F: Intrinsic hardness of the CCFD instances compared to other random spin-glass problems

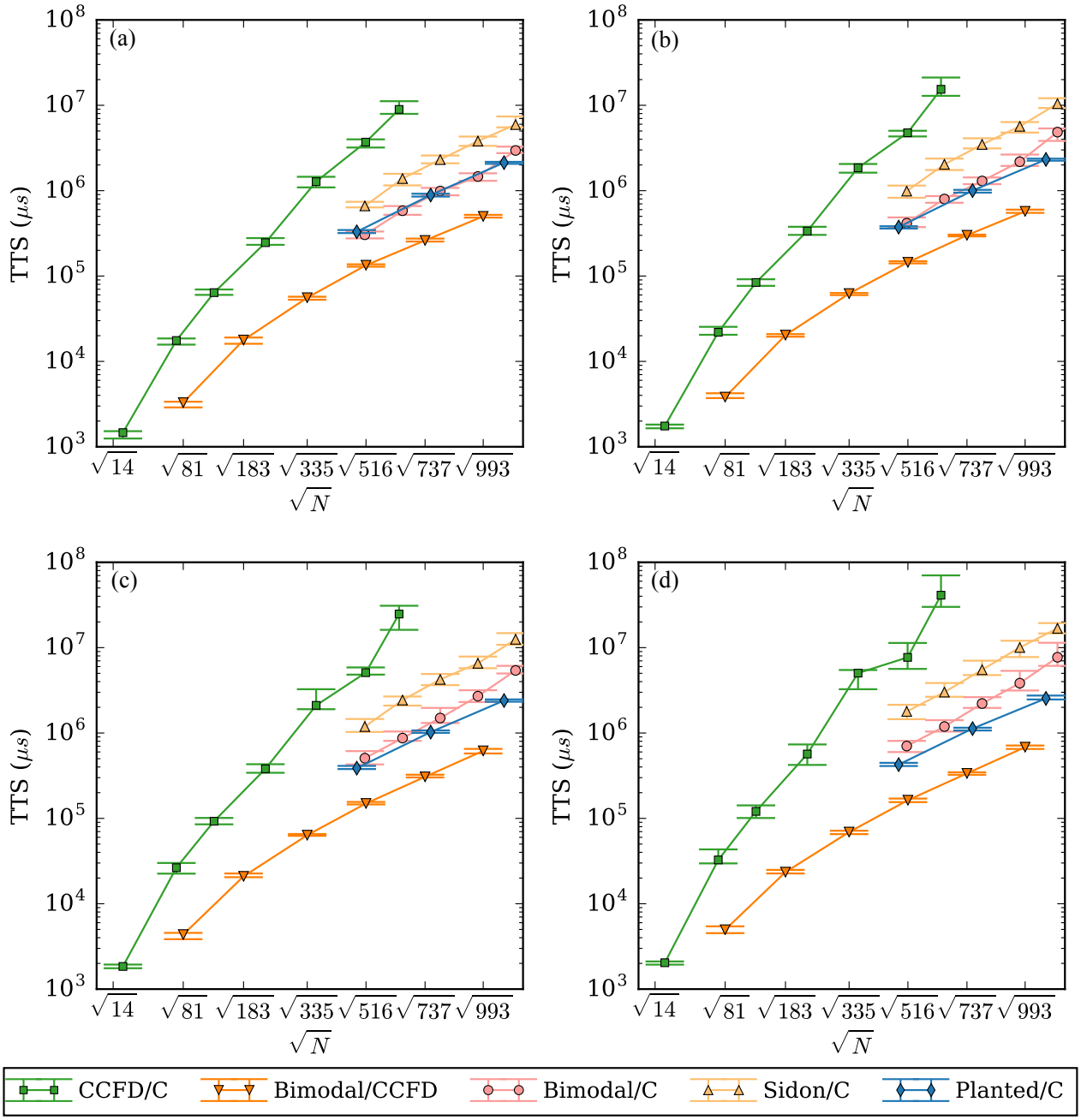

FIG. 10: Comparison of CCFD-based benchmark problems against other random spin-glass benchmark classes, at different percentiles. Data points correspond to the specific percentile value extracted from a bootstrapping statistical analysis from 100 instances per problem size, with error bars indicating the 90% confidence intervals (CI).

Appendix G: Scaling analysis from the application-centric perspective

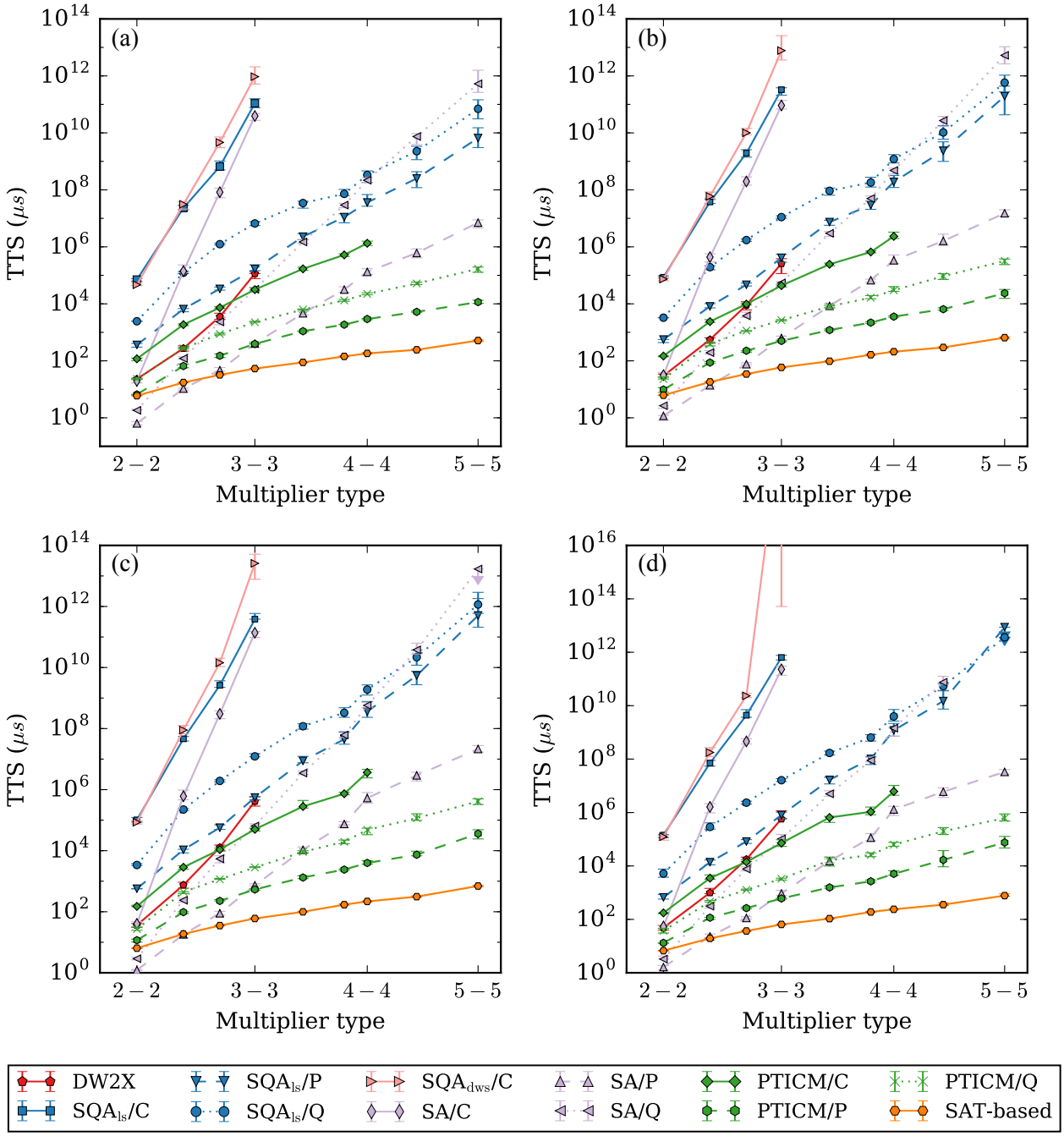

FIG. 11: Scaling analysis from the application-centric perspective at different percentile levels. (a) 25th, (b) 50th, (c) 60th, and (d) 75th percentile. Data points correspond to the specific percentile value extracted from a bootstrapping statistical analysis from 100 instances per problem size, with error bars indicating the 90% confidence intervals (CI).

Appendix H: Scaling analysis from the physics perspective

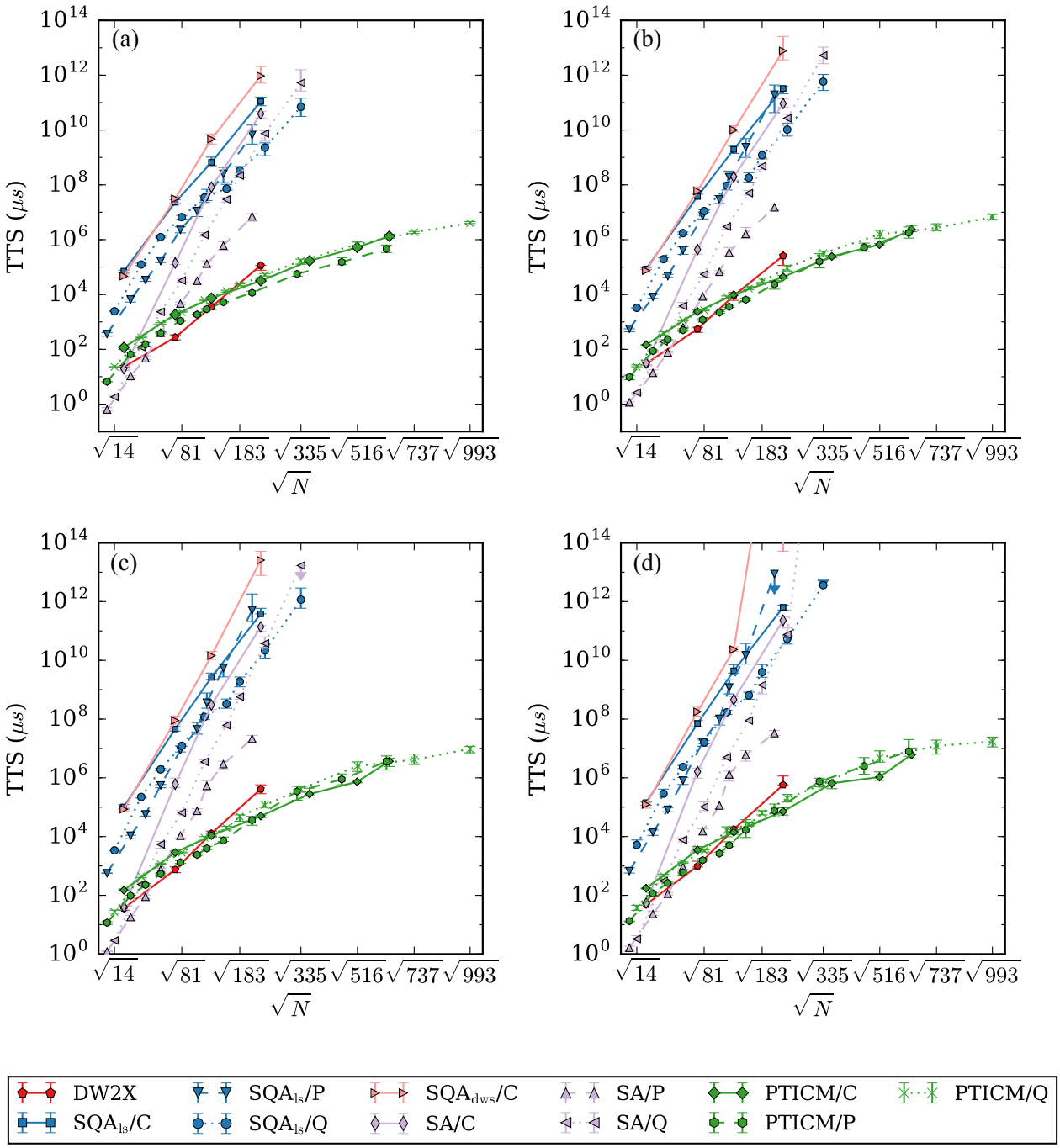

FIG. 12: Scaling analysis from the physics perspective at different percentile levels. (a) 25th, (b) 50th, (c) 60th, and (d) 75th percentile. Data points correspond to the specific percentile value extracted from a bootstrapping statistical analysis from 100 instances per problem size, with error bars indicating the 90% confidence intervals (CI).

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

[2] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll, *Quantum annealing: A new method for minimizing multidimensional functions*, Chem. Phys. Lett. **219**, 343 (1994).

[3] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, *A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem*, Science **292**, 472 (2001).

[4] G. Santoro, E. Martoňák, R. Tosatti, and R. Car, *Theory of quantum annealing of an Ising spin glass*, Science **295**, 2427 (2002).

[5] A. Das and B. K. Chakrabarti, *Quantum Annealing and Related Optimization Methods* (Edited by A. Das and B.K. Chakrabarti, Lecture Notes in Physics 679, Berlin: Springer, 2005).

[6] G. E. Santoro and E. Tosatti, *TOPICAL REVIEW: Optimization using quantum mechanics: quantum annealing through adiabatic evolution*, J. Phys. A **39**, R393 (2006).

[7] A. Das and B. K. Chakrabarti, *Quantum Annealing and Analog Quantum Computation*, Rev. Mod. Phys. **80**, 1061 (2008).

[8] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, et al., *Quantum annealing with manufactured spins*, Nature **473**, 194 (2011).

[9] N. G. Dickson, M. W. Johnson, M. H. Amin, R. Harris, F. Altomare, A. J. Berkley, P. Bunyk, J. Cai, E. M. Chapple, P. Chavez, et al., *Thermally assisted quantum annealing of a 16-qubit problem*, Nat. Commun. **4**, 1903 (2013).

[10] S. Boixo, T. Albash, F. M. Spedalieri, N. Chancellor, and D. A. Lidar, *Experimental signature of programmable quantum annealing*, Nat. Commun. **4**, 2067 (2013).

[11] H. G. Katzgraber, F. Hamze, and R. S. Andrist, *Glassy Chimeras Could Be Blind to Quantum Speedup: Designing Better Benchmarks for Quantum Annealing Machines*, Phys. Rev. X **4**, 021008 (2014).

[12] S. Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer, *Evidence for quantum annealing with more than one hundred qubits*, Nat. Phys. **10**, 218 (2014).

[13] T. F. Rønnow, Z. Wang, J. Job, S. Boixo, S. V. Isakov, D. Wecker, J. M. Martinis, D. A. Lidar, and M. Troyer, *Defining and detecting quantum speedup*, Science **345**, 420 (2014).

[14] H. G. Katzgraber, F. Hamze, Z. Zhu, A. J. Ochoa, and H. Munoz-Bauza, *Seeking Quantum Speedup Through Spin Glasses: The Good, the Bad, and the Ugly*, Phys. Rev. X **5**, 031026 (2015).

[15] B. Heim, T. F. Rønnow, S. V. Isakov, and M. Troyer, *Quantum versus classical annealing of Ising spin glasses*, Science **348**, 215 (2015).

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

[17] E. G. Rieffel, D. Venturelli, B. O'Gorman, M. B. Do, E. M. Prystay, and V. N. Smelyanskiy, *A case study in programming a quantum annealer for hard operational planning problems*, Quant. Inf. Proc. **14**, 1 (2015).

[18] S. Boixo, V. N. Smelyanskiy, A. Shabani, S. V. Isakov, M. Dykman, V. S. Denchev, M. H. Amin, A. Y. Smirnov, M. Mohseni, and H. Neven, *Computational multiqubit tunnelling in programmable quantum annealers*, Nat. Comm. **7**, 10327 (2016).

[19] V. S. Denchev, S. Boixo, S. V. Isakov, N. Ding, R. Babbush, V. Smelyanskiy, J. Martinis, and H. Neven, *What is the Computational Value of Finite Range Tunneling?*, Phys. Rev. X **6**, 031015 (2016).

[20] J. King, S. Yarkoni, J. Raymond, I. Ozfidan, A. D. King, M. M. Nevisi, J. P. Hilton, and C. C. McGeoch, *Quantum annealing amid local ruggedness and global frustration*, Journal of the Physical Society of Japan **88**, 061007 (2019).

[21] S. Mandrà, Z. Zhu, W. Wang, A. Perdomo-Ortiz, and H. G. Katzgraber, *Strengths and weaknesses of weak-strong cluster problems: A detailed overview of state-of-the-art classical heuristics versus quantum approaches*, Phys. Rev. A **94**, 022337 (2016).

[22] A. Perdomo, C. Truncik, I. Tubert-Brohman, G. Rose, and A. Aspuru-Guzik, *Construction of model hamiltonians for adiabatic quantum computation and its application to finding low-energy conformations of lattice protein models*, Phys. Rev. A **78**, 012320 (2008).

[23] A. Perdomo-Ortiz, N. Dickson, M. Drew-Brook, G. Rose, and A. Aspuru-Guzik, *Finding low-energy conformations of lattice protein models by quantum annealing*, Sci. Rep. **2**, 571 (2012).

[24] F. Gaitan and L. Clark, *Ramsey numbers and adiabatic quantum computing*, Phys. Rev. Lett. **108**, 010501 (2012).

[25] O'Gorman, B., Babbush, R., Perdomo-Ortiz, A., Aspuru-Guzik, A., and Smelyanskiy, V., *Bayesian network structure learning using quantum annealing*, Eur. Phys. J. Special Topics **224**, 163 (2015).

[26] A. Perdomo-Ortiz, J. Fluegemann, S. Narasimhan, R. Biswas, and V. N. Smelyanskiy, *A quantum annealing approach for fault detection and diagnosis of graph-based systems*, Eur. Phys. J. Special Topics **224**, 131 (2015).

[27] K. M. Zick, O. Shehab, and M. French, *Experimental quantum annealing: case study involving the graph isomorphism problem*, Scientific Reports **5**, 11168 EP (2015).

[28] F. Neukart, G. Compostella, C. Seidel, D. von Dollen, S. Yarkoni, and B. Parney, *Traffic flow optimization using a quantum annealer*, arXiv:1708.01625v2 (2017).

[29] Z. Bian, F. Chudak, R. B. Israel, B. Lackey, W. G. Macready, and A. Roy, *Mapping constrained optimization problems to quantum annealing with application to fault diagnosis*, Frontiers in ICT **3**, 14 (2016).

[30] A. Feldman, G. Provan, and A. van Gemund, *Approximate model-based diagnosis using greedy stochastic search*, Journal of Artificial Intelligence Research **38**, 371 (2010).

[31] A. Feldman, G. Provan, and A. van Gemund, in *Abstraction, Reformulation, and Approximation: 7th International Symposium, SARA 2007, Whistler, Canada, July 18-21, 2007. Proceedings*, edited by I. Miguel and W. Ruml (Springer Berlin Heidelberg, Berlin, Heidelberg, 2007), pp. 139–154.

[32] T. Eiter and G. Gottlob, *The complexity of logic-based abduction*, Journal of the ACM **42**, 3 (1995).

[33] Rieger, H. and Kawashima, N., *Application of a continuous time cluster algorithm to the two-dimensional random quantum ising ferromagnet*, Eur. Phys. J. B **9**, 233 (1999).

[34] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, *Optimization by simulated annealing*, Science **220**, 671 (1983).

[35] S. V. Isakov, I. N. Zintchenko, T. F. Rønnow, and M. Troyer, *Optimized simulated annealing for Ising spin glasses*, Comput. Phys. Commun. **192**, 265 (2015), (see also ancillary material to arxiv:cond-mat/1401.1084).

[36] K. Hukushima and K. Nemoto, *Exchange Monte Carlo method and application to spin glass simulations*, J. Phys. Soc. Jpn. **65**,

1604 (1996).

[37] H. G. Katzgraber, S. Trebst, D. A. Huse, and M. Troyer, *Feedback-optimized parallel tempering Monte Carlo*, J. Stat. Mech. P03018 (2006).

[38] J. J. Moreno, H. G. Katzgraber, and A. K. Hartmann, *Finding low-temperature states with parallel tempering, simulated annealing and simple Monte Carlo*, Int. J. Mod. Phys. C **14**, 285 (2003).

[39] Z. Zhu, A. J. Ochoa, and H. G. Katzgraber, *Efficient Cluster Algorithm for Spin Glasses in Any Space Dimension*, Phys. Rev. Lett. **115**, 077201 (2015).

[40] S. Mandrà, Z. Zhu, W. Wang, A. Perdomo-Ortiz, and H. G. Katzgraber, *Strengths and weaknesses of weak-strong cluster problems: A detailed overview of state-of-the-art classical heuristics versus quantum approaches*, Phys. Rev. A **94**, 022337 (2016).

[41] A. Lucas, *Ising formulations of many NP problems*, Front. Physics **12**, 5 (2014).

[42] N. Chancellor, S. Zohren, and P. A. Warburton, *Circuit design for multi-body interactions in superconducting quantum annealing systems with applications to a scalable architecture*, npj Quantum Information **3**, 21 (2017).

[43] E. Boros and P. L. Hammer, *Pseudo-boolean optimization*, Discrete Appl. Math. **123**, 155 (2002).

[44] V. Choi, *Minor-embedding in adiabatic quantum computation: II. minor-universal graph design*, Quantum Information Processing **10**, 343 (2011), ISSN 1570-0755.

[45] J. Cai, B. Macready, and A. Roy, *A practical heuristic for finding graph minors*, arXiv:1406.2741 (2014).

[46] A. Perdomo-Ortiz, J. Fluegemann, R. Biswas, and V. N. Smelyanskiy, *A performance estimator for quantum annealers: Gauge selection and parameter setting*, arXiv:1503.01083 (2015).

[47] Z. Zhu, A. J. Ochoa, F. Hamze, S. Schnabel, and H. G. Katzgraber, *Best-case performance of quantum annealers on native spin-glass benchmarks: How chaos can affect success probabilities*, Phys. Rev. A **93**, 012317 (2016).

[48] S. Sidon, *Ein Satz über trigonometrische Polynome und seine Anwendung in der Theorie der Fourier-Reihen*, Mathematische Annalen **106**, 536 (1932).

[49] A. Selby, *Efficient subgraph-based sampling of isingtype models with frustration*, arXiv:1409.3934 (2014).

[50] T. Albash and D. A. Lidar, *Evidence for a limited quantum speedup on a quantum annealer*, arXiv:1705.07452 (2017).

[51] J. Job and D. Lidar, *Test-driving 1000 qubits*, Quantum Science and Technology **3**, 030501 (2018).

[52] S. Aaronson, *Google, d-wave, and the case of the factor - 10**8 speedup for what?*, http://www.scottaaronson.com/blog/?p=2555 (2015).

[53] S. Aaronson, *Insert d-wave post here*, http://www.scottaaronson.com/blog/?p=3192 (2015).

[54] L. Hormozi, E. W. Brown, G. Carleo, and M. Troyer, *Nonstoquastic hamiltonians and quantum annealing of an ising spin glass*, Phys. Rev. B **95**, 184416 (2017).

[55] H. Nishimori and K. Takada, *Exponential enhancement of the efficiency of quantum annealing by non-stoquastic hamiltonians*, Frontiers in ICT **4**, 2 (2017).

[56] A. Biere, SPLATZ, LINGELING, PLINGELING, TREENGELING, YALSAT *entering the SAT competition 2016*, SAT COMPETITION 2016 p. 44 (2016).

[57] J. Strand, A. Przybysz, D. Ferguson, and K. Zick, *Zzz coupler for native embedding of max-3sat problem instances in quantum annealing hardware*, Bulletin of the American Physical Society (2017).

[58] A. Perdomo-Ortiz, S. E. Venegas-Andraca, and A. Aspuru-Guzik, *A study of heuristic guesses for adiabatic quantum computation*, Quantum Inf. Process. **10**, 33 (2010), ISSN 1570-0755, 1573-1332.

[59] N. Chancellor, *Modernizing quantum annealing using local searches*, New Journal of Physics **19**, 023024 (2017).

[60] N. Chancellor, *Modernizing quantum annealing ii: Genetic algorithms and inference*, arXiv:1609.05875 (2017).

[61] H. Karimi, G. Rosenberg, and H. G. Katzgraber, *Effective optimization using sample persistence: A case study on quantum annealers and various monte carlo optimization methods*, Phys. Rev. E **96**, 043312 (2017).

[62] R. Harris, M. W. Johnson, T. Lanting, A. J. Berkley, J. Johansson, P. Bunyk, E. Tolkacheva, E. Ladizinsky, N. Ladizinsky, T. Oh, et al., *Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor*, Phys. Rev. B. **82**, 024511 (2010).

[63] F. Barahona, *On the computational complexity of ising spin glass models*, Journal of Physics A: Mathematical and General **15**, 3241 (1982).

[64] T. Albash, S. Boixo, D. A. Lidar, and P. Zanardi, *Quantum adiabatic markovian master equations*, New J. Phys. **14**, 123016 (2012).

[65] V. N. Smelyanskiy, D. Venturelli, A. Perdomo-Ortiz, S. Knysh, and M. I. Dykman, *Quantum annealing via environment-mediated quantum diffusion*, Phys. Rev. Lett. **118**, 066802 (2017).

[66] S. V. Isakov and R. Moessner, *Interplay of quantum and thermal fluctuations in a frustrated magnet*, Phys. Rev. B **68**, 104409 (2003).

[67] H. G. Katzgraber, *Introduction to Monte Carlo Methods* (2009), (arXiv:0905.1629).

[68] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, *Quantum-assisted learning of hardware-embedded probabilistic graphical models*, Phys. Rev. X **7**, 041052 (2017).

[69] A. Perdomo-Ortiz, M. Benedetti, J. Realpe-Gómez, and R. Biswas, *Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers*, Quantum Science and Technology **3**, 030502 (2018).

[70] V. Choi, *Minor-embedding in adiabatic quantum computation: I. the parameter setting problem*, arXiv:0804.4884 (2008).

[71] K. L. Pudenz, *Parameter setting for quantum annealers*, arXiv:1611.07552 (2016).

[72] R. Babbush, A. Perdomo-Ortiz, B. O'Gorman, W. Macready, and A. Aspuru-Guzik, *Construction of energy functions for lattice heteropolymer models: A case study in constraint satisfaction programming and adiabatic quantum optimization*, Adv. Chem. Phys. **155**, 201 (2014).

[73] R. Babbush, B. O'Gorman, and A. Aspuru-Guzik, *Resource efficient gadgets for compiling adiabatic quantum optimization problems*, Annalen der Physik **525**, 877 (2013).