Verifying the output of quantum optimizers with ground-state energy lower bounds

Flavio Baccari,[1,2] Christian Gogolin ,[1] Peter Wittek,[3,4,5,6,*] and Antonio Acín [1,7]

[1]*ICFO-Institut de Ciencies Fotoniques, The Barcelona Institute of Science and Technology, 08860 Castelldefels (Barcelona), Spain*
[2]*Max-Planck-Institut für Quantenoptik, Hans-Kopfermann-Straße 1, 85748 Garching, Germany*
[3]*Rotman School of Management, University of Toronto, M5S 3E6 Toronto, Canada*
[4]*Creative Destruction Lab, M5S 3E6 Toronto, Canada*
[5]*Vector Institute for Artificial Intelligence, M5G 1M1 Toronto, Canada*
[6]*Perimeter Institute for Theoretical Physics, N2L 2Y5 Waterloo, Canada*
[7]*ICREA, Passeig Lluis Companys 23, 08010 Barcelona, Spain*

Solving optimization problems encoded in the ground state of classical-spin systems is a focus area for quantum computing devices, providing upper bounds to the unknown solution. To certify these bounds, they are compared to those obtained by classical methods. However, even if the quantum bound beats them, this says little about how close it is to the unknown solution. We consider the use of relaxations to the ground-state problem as a benchmark for the output of quantum optimizers. These relaxations are radically more informative because they provide lower bounds to the ground-state energy. The chordal branch and bound algorithm we present provides a series of systematically improving confidence regions where the ground-state energy provably lies. Interestingly, each step in the process requires only an effort polynomial in the system size. Additionally, the algorithm exploits the locality and sparsity of relevant Ising spin models in a systematic way. This yields certified solutions for many of the problems that are currently addressed by heuristic optimization algorithms more efficiently and for larger system sizes. We apply the method to verify the output of a D-Wave 2000Q device and identify instances where the annealer does not reach the ground-state energy and, more importantly, instances where it does, something impossible to do by means of standard variational approaches. Our work provides a flexible and scalable method for the verification of the outputs produced by quantum optimization devices.

I. INTRODUCTION

Classical Ising models are among the most paradigmatic and widely studied models in statistical physics. They are capable of describing an immense variety of interesting physics, ranging from ferromagnetic to frustrated and glassy phases. Moreover, they are important in fields as diverse as risk assessment in finance, logistics, machine learning [1], and image denoising [2] because the solution of many optimization and decision problems, such as partitioning, covering, and satisfiability, can be encoded in the ground state of such models [3]. Their generality and the exponentially growing spaces of spin configurations, however, preclude the existence of any efficient general purpose algorithm to obtain the ground state. It is hence no surprise that a wealth of approximate but more scalable classical techniques for the energy minimization in such models have been developed.

Recently, novel approaches that leverage the power of near-term quantum devices such as quantum annealers, vari-

ational quantum eigensolvers [4], variational circuits [5,6], or networks of degenerate optical parametric oscillators [7] are proposed for performing such tasks [8–10]. The quality of their outputs is usually benchmarked against some of the most scalable classical approaches, e.g., simulated annealing [11] or variational ansatz classes based on tensor-network states [12]. All of these methods share one common feature: they only provide upper bounds on the ground-state energy. On the one hand, this feature limits the verification power of these approaches, which are only able to identify instances where quantum devices do not reach the ground-state energy. On the other hand, even when a quantum optimizer beats all classical variational methods, there is no way to know if the output of the quantum device is actually close to the true ground-state energy, unless the test is performed on problems for which the solution is already known [13]. To overcome these limitations, it is important to develop schemes that provide reliable lower bounds to the ground-state energy of spin problems, against which the results of quantum devices, and also classical variational methods, can be compared.

In this work, we tackle this issue by leveraging relaxations of polynomial optimization problems through semidefinite programming (SDP). The proposed method provides lower bounds on the ground-state energy by optimizing over a larger set than the physical spin configurations. We improve the scalability of this type of relaxations, by making use of a method known as the chordal extension, which allows us to

---

exploit the physical locality and sparsity structure present in relevant problem instances. All in all, this yields an increasingly precise hierarchy of rigorous lower and, in fact, also upper bounds on the ground-state energy. Combining these bounds we obtain a scalable and flexible method that provides with polynomial effort a confidence region inside which the ground-state energy provably lies. By making use of a branch-and-bound scheme, the confidence region can be systematically improved. Although the complexity of the general Ising problem implies that one might have to run an exponential number of steps to achieve convergence, our numerical experiments show that in many instances the confidence region collapses to the exact ground state after few iterations. A benchmark on a D-Wave 2000Q device shows how our chordal branch-and-bound (CBB) method can be used both to detect situations in which the quantum solution differs from the optimum and, more importantly, verify when the annealer has actually reached the ground-state energy and no further optimization is required.

Our approach to verification consists of relaxing the initial optimization problem and, therefore, results in lower bounds to the ground-state energy, something impossible using standard variational techniques. Certification methods based on relaxations constitute a valid approach for benchmarking any heuristic optimization, classical or quantum. The main purpose of this work is to demonstrate how these methods can be applied to asses the quality of the outputs of intermediate-scale quantum computing devices, whose timeliness is particularly motivated by recent progresses in the field [14].

II. PRELIMINARIES

We consider classical spin systems whose configurations $\vec{\sigma} := (\sigma_1, \ldots, \sigma_N)$ are vectors of N spin variables $\sigma_i \in \{-1, 1\}$ to each of which a Hamiltonian H assigns an energy $H(\vec{\sigma})$. We are mostly interested in Hamiltonians of Ising type, that can be written in the form

$$H(\vec{\sigma}) = \sum_{1 \leqslant i < j \leqslant N} J_{ij} \sigma_i \sigma_j + \sum_{i=1}^{N} h_i \sigma_i, \tag{1}$$

with couplings J_{ij} and local fields h_i. The method we develop, however, is more general and can also be applied to Hamiltonians that are higher order polynomials of the spin variables σ_i and couple three or more spins in a single term. Among all the configurations of such a system there are those that achieve the minimal possible energy, also known as the ground-state energy and defined as

$$E_g := \min_{\vec{\sigma} \in \{-1,1\}^N} H(\vec{\sigma}). \tag{2}$$

This minimization is a polynomial optimization problem in the spin variables. If the Hamiltonian is of the form in (1), then it is quadratic. For our purposes, solving the ground-state problem for a given Hamiltonian means finding E_g and outputting a configuration that achieves it. Obviously, finding the ground state is an optimization problem that can, in principle, be solved by brute force search. This however quickly becomes infeasible as the number of configurations grows as 2^N, restricting this approach to systems of few tens of particles.

Many spin models of interest in physics and beyond are characterized by a locality structure defined by a graph G: Spins are located in the nodes of the graph and the interacting terms J_{ij} are nonzero only between neighboring sites, that is, spins connected by an edge of the graph. Local interactions also appear naturally in physical solvers of spin models, such as, for instance, a quantum annealer. Such locality of interactions implies a sparsity of the resulting Hamiltonian and hence optimization problem. However, exploring this structure still remains a really nontrivial task: Even for rather restricted classes of graphs, finding the ground state is an NP-hard problem. This precludes the existence of any efficient and general algorithm. The complexity of the Ising ground-state problem thereby depends on subtle details of the problem class (see Appendix A for more details).

It is convenient for what follows to present an equivalent formulation of the ground-state problem in which the energy is computed over the set of expectation values of products of spin variables, such as the σ_i and $\sigma_i \sigma_j$ that appear in the Hamiltonian, instead of spin variables directly. The connection with the original optimization problem (2) is made by introducing the notion of a state of an N-spin system as a generic probability distribution P over the set of configurations $\{-1, 1\}^N$. For every function $f : \{-1, 1\}^N \to \mathbb{R}$ we can then define its expectation value in the state P, denoted by

$$\langle f \rangle_P = \sum_{\vec{\sigma} \in \{-1,1\}^N} f(\vec{\sigma}) P(\vec{\sigma}). \tag{3}$$

In the following, whenever the connection to a specific physical state P is not evident, we simply denote expectation values by $\langle f \rangle$. With this notation, the equivalent energy minimization problem reads

$$H(\vec{\sigma}) = \min_P - \sum_{1 \leqslant i < j \leqslant N} J_{ij} \langle \sigma_i \sigma_j \rangle_P + \sum_{i=1}^{N} h_i \langle \sigma_i \rangle_P. \tag{4}$$

We call any state P that is supported only on the ground-state space (i.e., the collection of all configurations that achieve the ground-state energy) of a model a ground state and such P are manifestly those that achieve the minimal possible expectation value for the Hamiltonian, i.e., $\min_P \langle H \rangle_P = E_g$. Optimizing over probability distributions instead of spin configurations requires a similar exponential effort in the system size. Nonetheless, it opens the way for a relaxation in terms of an SDP problem, which is one of the main ingredients of our method, which we present in the following section.

III. THE CHORDAL BRANCH AND BOUND ALGORITHM

In this section we describe the main tools we build upon to devise the CBB algorithm. First, we present the SDP relaxation we exploit to obtain both a lower and upper bound to the ground-state energy. These bounds define an energy confidence region in which the unknown ground-state energy provably lies. Then we introduce the branch-and-bound procedure as a tool to systematically improve this confidence region. Last, we describe the chordal extension method that exploits the sparsity of relevant physical models in order to increase the scalability of the SDP. Technical details of

the algorithm and its implementation can be found in the Appendixes B and C.

A. SDP relaxations

As mentioned in the previous section, the ground-state minimization problem can be relaxed to obtain an efficient method to derive lower bounds on the ground-state energy through the formulation in terms of expectation values (4). Specifically we use a method pioneered by Lassere [15,16], which relaxes the polynomial optimization over any distribution P into an SDP.

Let us consider a vector $\vec{x} := (x_\alpha)_{\alpha=1}^k$ of monomials of the spin variables. For any state P we define its moment matrix $\Gamma(P)$ with respect to $\vec{x}$ as the $k \times k$ matrix of expectation values $\Gamma_{\alpha\beta}(P) := \langle x_\alpha x_\beta \rangle_P$. Any such moment matrix $\Gamma(P)$, being defined via an outer product, is manifestly positive semidefinite, i.e., $\Gamma(P) \succeq 0$, and, depending on what the elements of $\vec{x}$ are, it further obeys certain linear constraints, which follow from the fact the monomials are made of spin variables. More precisely, the constraints reflect the two basic properties of these variables, namely that they take dichotomic values $\sigma_i \in \{-1, 1\}$ and commute with each other. This leads to conditions such as, for instance, $\langle \sigma_i \sigma_j \sigma_i \rangle_P = \langle \sigma_j \rangle_P$.

We illustrate this with an example: take as a generating set of monomials $\vec{x}$ the spin variables themselves together with the identity, i.e., $\vec{x} = \{1, \sigma_1, \ldots, \sigma_N\}$. The resulting Γ matrix takes the following form:

$$\Gamma = \begin{pmatrix} 1 & \langle \sigma_1 \rangle & \langle \sigma_2 \rangle & \ldots & \langle \sigma_N \rangle \\ \langle \sigma_1 \rangle & 1 & \langle \sigma_1 \sigma_2 \rangle & \ldots & \langle \sigma_1 \sigma_N \rangle \\ \langle \sigma_2 \rangle & \langle \sigma_1 \sigma_2 \rangle & 1 & \ldots & \langle \sigma_2 \sigma_N \rangle \\ \vdots & \vdots & & \ddots & \\ \langle \sigma_N \rangle & \langle \sigma_1 \sigma_N \rangle & \ldots & & 1 \end{pmatrix}. \quad (5)$$

Notice how the expectation value of any Hamiltonian of the form given in (1) can be expressed as a linear function of the entries of Γ, given by $\mathrm{tr}(h\,\Gamma)$, where h is a matrix defined by the system Hamiltonian. A lower bound to its ground-state energy can then be obtained by minimizing $\mathrm{tr}(h\,\Gamma)$ over all positive semidefinite matrices Γ that fulfill the linear constraints discussed above, expressed also as linear functions of the entries of Γ in terms of some matrices F_m,

$$\begin{aligned} \min_\Gamma \quad & \mathrm{tr}(h\,\Gamma) \\ \text{s.t.} \quad & \Gamma \succeq 0, \\ & \mathrm{tr}(F_m\,\Gamma) = 0, \quad \forall\, m \in \{1, \ldots, k\}. \end{aligned} \quad (6)$$

This defines an SDP relaxation of the problem, since not every such positive matrix Γ satisfying the linear constraints encapsulated by the matrices F_m necessarily arise as a moment matrix $\Gamma(P)$ of a physical state. In contrast with the original minimization, the presented SDP can be solved efficiently, since the amount of variables involved in the moment matrix scales only quadratically with the number of spins.

Interestingly, from the solution of the considered relaxations one can also extract a spin configuration with no additional computational cost. Let Γ^* be the optimal solution to the SDP. We can associate to it a configuration $\vec{\sigma}^*$ by taking the sign of the entries in that matrix that correspond to the expectation values $\langle \sigma_i \rangle$, namely set $\sigma_i^* := \mathrm{sgn}(\Gamma^*_{1,i+1})$. The

energy of that configuration $H(\vec{\sigma}^*)$ clearly provides an upper bound to the ground-state energy.

Moreover, the approximation to the exact ground-state energy can be improved by considering the moment matrix generated by the vector $\vec{x}^{(\nu)}$ of all monomials of spin variables of degree up to ν. For every such a vector, it is possible to construct the corresponding moment matrix $\Gamma^{(\nu)}$ and solve the corresponding relaxation, as in (6). This process defines a hierarchy of relaxations ordered according to the degree of the considered monomials ν that provides an asymptotically converging series of tighter and tighter lower bounds on the ground-state energy (see Appendix B for more details). All the steps in the hierarchy are efficient, as they define SDP problems involving matrices that scale polynomially with the number of spins.

B. Branch and bound

The derived lower and upper bounds through the previous SDP relaxation can be combined with a so-called branch-and-bound (BB) technique to obtain a series of complementing bounds converging to the exact solution. This is a general iteration strategy that has been applied in several different ways (see, for instance, Ref. [17] for a review). The main ingredient of a BB iteration is the branching procedure, which consists in dividing the original problem into two subproblems that correspond to the opposite cases of a dichotomic choice. In the ground-state problem, it can be done by choosing a spin i and considering the two subsets of spin configurations that have $\sigma_i = \pm 1$ fixed. Finding the ground state in both subcases can be cast as another ground-state problem for a modified graph where the vertex i has been removed and the couplings have been modified accordingly. Obviously, the value of the original ground-state energy is just the minimum between the solutions of the two subcases.

The trick is now to use the upper and lower bounds to reduce the number of branches to explore. The BB procedure does that as follows:

(i) Start with the original graph and compute a lower and upper bound z_L, z_U to the ground-state energy, in our case using the previous SDP relaxation

(ii) If the bounds differ, choose a branching and compute lower and upper bounds for the two subcases

(iii) Keep track of the best upper bound $\bar{z}_U$ encountered so far and discard all the explored branches in which the lower bound is higher than $\bar{z}_U$

(iv) From the reduced list of branches, pick the one corresponding to the lowest z_L, if it still differs from the best upper bound $\bar{z}_U$, go back to point (ii) and perform another branching

(v) Keep repeating until the lowest z_L and the best upper bound $\bar{z}_U$ coincide.

Although the BB procedure always converges to the solution, it may require an exponential number of steps when implemented on hard instances. Yet, the method presents two important properties: (1) it provides a constantly improving energy range for the ground-state energy and (2) at all steps, it is known whether the searched solution has been reached, possibly up to numerical precision, and no more steps are needed. In fact, we have observed that in many situations the algorithm can be stopped after a few steps because it has been able to
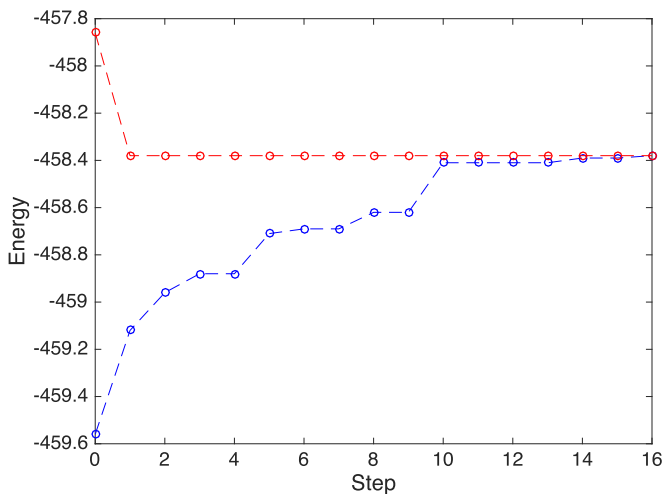
FIG. 1. Example of sequences of upper and lower bounds obtained during the branching procedure with the CBB method, for a 2D Ising model of lattice size 15×15. The convergence is met, yielding a certified value for the ground-state energy and configuration, after less than 20 steps, instead of having to explore all of the 2^{225} configurations.

find the solution. Figure 1 shows a typical instance of the BB procedure in which the lower and upper bounds converge after a few iterations. More details about the implementation of the BB procedure can be found in Appendix C.

C. Chordal extension

The last ingredient in our construction uses the fact that, for the considered spin systems with local interactions, the optimization problem defined by the Hamiltonian (1) is sparse. As shown in Refs. [18,19], one can exploit this sparsity to derive a similar relaxation that is more scalable than the previous one. Intuitively, the idea behind the modification is the following: for any pair of noninteracting spins i, j, the corresponding two-body expectation value $\langle \sigma_i \sigma_j \rangle$ is not needed for the computation of the energy. Thus, a moment matrix with all two-body correlations is including some potentially unnecessary information. Finding the minimal amount of moments that is sufficient to effectively constrain the optimization of the energy helps defining a more efficient, and therefore scalable, relaxation.

To illustrate the method, let us suppose that the graph G of the problem is already chordal. A graph is said to be chordal if all its cycles of four or more vertices have a chord, i.e., an edge that is not part of the cycle but connects two vertices of the cycle. If G is not chordal, it is always possible to associate to it a so-called chordal extension G_C by properly adding some edges (see Fig. 2 for an example). Notice that the chordal extension of a graph is not unique, but a chordal extension can always be found, because the fully connected graph is chordal. For a chordal extension to be useful for our CBB method it needs to be still relatively sparse, as in the example presented here. In Appendix B we provide a general polytime technique to find good chordal extensions for all the cases studied in this paper.
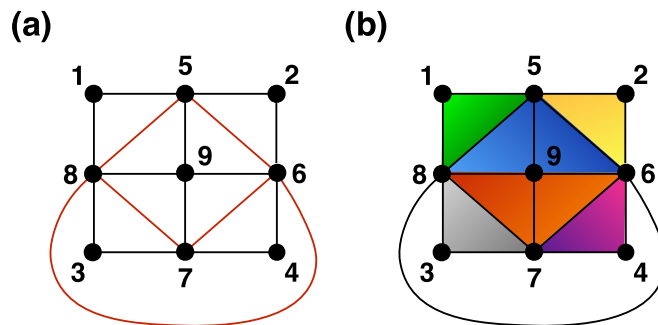

FIG. 2. (a) An example of graph G that is not chordal, together with an option of edges to add (shown in red) in order to make it chordal, obtaining a chordal extension G_C. The new edges are chosen to "break" the larger cycles, such as the cycle $\{1, 5, 9, 8\}$. (b) The graph G_C and its corresponding maximal cliques C_l, shown with the different colors. In this example, the graph has $n_c = 6$ cliques, out of them two are composed of four vertices, while the remaining ones consist of three vertices each.

Once the chordal extension has been derived, one can then introduce a relaxation where the original Γ matrix is replaced by a direct sum of smaller matrices Γ_l, constructed only from the spin variables belonging to the each of the n_C maximal cliques, i.e., fully connected subgraphs of maximal size, of G_C. Note that some spin variables appear in more than one clique, which means that the SDP does not completely decouple into separate SDPs for each block Γ_l, so the optimization involves moments appearing in multiple blocks.

As an illustration let us go back to the previous relaxation and suppose one wants to solve the 1D Ising model with Hamiltonian $H = \sum_{i=1}^{N} J_{i,i+1} \sigma_i \sigma_{i+1}$. The corresponding dependency graph G is already chordal and is composed of N cliques $C_i = \sigma_i, \sigma_{i+1}$, with $i = 1, \dots, N-1$. Then the matrix (5) can be substituted by a direct sum of the N blocks

$$\Gamma_{C_i} = \begin{pmatrix} 1 & \langle \sigma_i \rangle & \langle \sigma_{i+1} \rangle \\ \langle \sigma_i \rangle & 1 & \langle \sigma_i \sigma_{i+1} \rangle \\ \langle \sigma_{i+1} \rangle & \langle \sigma_i \sigma_{i+1} \rangle & 1 \end{pmatrix}. \quad (7)$$

Unnecessary expectation values in (5), such as $\langle \sigma_i \sigma_{i+2} \rangle$, no longer appear in any of the N smaller blocks Γ_{C_i}, but all the expectation values that are needed to define the Hamiltonian as a linear function of the moment matrices are still present. This simplification is particularly useful because it can significantly reduce the number of variables involved in the SDP and it reduces the size of the largest positive semidefinite block, which dramatically reduces computational footprint in the numerical algorithms solving the optimization. In this example, we go from a matrix whose size increases quadratically with the number of spins, to a linearly increasing set of constant size matrices. In general, the constraints between different blocks do not allow to split the problem into n_C independent ones, but one can still see that the scaling of the computational effort is dominated by the size of the largest block alone (see Appendix B for more details on the chordal extension hierarchy).

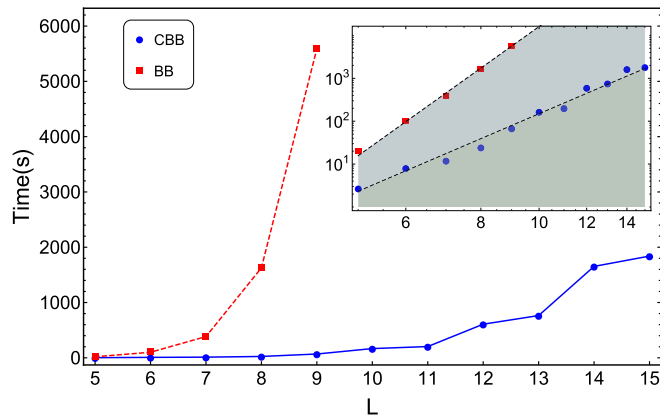As an illustration of the gain in scalability provided by the chordal extension, we compare the performance of

FIG. 3. Time comparison for a square lattice of N sites between a standard SDP-based branch and bound (BB) algorithm and the one augmented with the chordal extension (CBB), as a function of the linear lattice size $L = \sqrt{N}$. The problem is finding the ground-state energy for a 2D Ising ferromagnetic model with random Gaussian magnetic field, close to the phase transition at $\sigma = 1.5$, i.e., where the ground state is already partially disordered. The time estimation was averaged over 100 disorder realizations, except for the largest system size, where the averaging was reduced to 10 samples. Due to the large amount of disorder averaging, we limit ourselves to system sizes $L \leqslant 15$, far below the maximum sizes we can tackle on our hardware. The comparison is shown in a linear scale and double logarithmic in the inset. The dashed lines in the inset are power laws of the form L^{10} and L^6, demonstrating the claimed polynomial scaling of the runtime N^5 for BB vs N^3 for CBB.

the CBB method with a sparse Ising problem in the two cases of exploiting and not exploiting the chordal extension. As a benchmark of a sparse instance, we consider the standard 2D ferromagnetic Ising model in a statically disordered magnetic field (quenched disorder) that is picked independently from normal distributions of mean zero and variance σ for each site. Similar results are obtained for other models with local interactions. As a function of the disorder strength σ, the model undergoes a phase transition from a ferromagnetic ground state (in which all states are aligned with each other) to a disordered phase (in which, for extremely large disorder, the spins are aligned with the local magnetic fields). For this model it is known that the ground state can in principle be found in polynomial time (see Appendix A).

Indeed, the nonchordal BB method is able to find the solution with an effort that scales roughly with a N^5 dependence; see Fig. 3. However, and especially in the interesting region close to the phase transition, fast growing memory requirements and runtime make the method impractical for systems that are larger than 15×15 on the hardware we have at our disposal. The CBB method, in contrast, allows us to solve systems of over 35×35 sites on the same hardware, due to both lower memory requirements and a very significantly reduced runtime, both in terms of absolute numbers and in terms of scaling (see Fig. 3 for a comparison). When using the chordal extension, the method scales roughly as N^3, as opposed to the N^5 dependence without it.

IV. VERIFYING THE SOLUTION OF A QUANTUM ANNEALER

Once all the ingredients of the method have been presented, we now turn to the main part of our work and show how to apply our approach to verify the results of energy minimizations performed on an actual quantum annealing device.

Numerical computations in this work were run on a workstation with an Intel Xeon E5-1650v4 processor with six physical cores clocked at 3.60 GHz base frequency and 128 GByte RAM. Due to the polynomial scaling of the method, much larger system sizes can be reached with more powerful hardware. The sparse semidefinite relaxations were generated by Ncpol2sdpa [20], and the semidefinite programs were solved by Mosek [21]. The code for the experiments is available under an open source license [22].

A. Verifying solutions for a triangular graph

To show the flexibility of the CBB method and also to verify a quantum annealing solution for the largest system size simulable on a state-of-the art annealer, we considered a 2D triangular lattice. In fact, spin models on the triangular lattices display a wealth of interesting physical phenomena, many driven by the possibility to have frustrated interactions. To remain in a regime that is comparable to the benchmarking we did before, we, however, concentrate on the interplay of ferromagnetic interactions with a disordered magnetic field (for a numerical analysis of the corresponding phase transition, see Appendix D).

We used a D-Wave 2000Q quantum annealer with 2040 functional qubits. The chip had eight faulty qubits and the corresponding couplings were removed from a full-yield 16×16 Chimera graph. We used the virtual full-yield Chimera graph abstraction to ensure consistent embeddings and improve the quality of the results. The coupling strengths were automatically scaled to the interval $[-1, 1]$, and the logical qubits used a coupling strength of -2 to hold a chain of physical spins together. The minor embedding was a heuristic method, yielding a chain length of 7. We also tried chains up to length 22, without significant change in the results, showing that the scaling in the couplings ensures that the chains do not break. For each data point, we sampled a thousand data points and chose the one with the lowest energy as the optimum. This takes constant time irrespective of the values set, in the range of milliseconds. The flux bias of the qubits was not offset.

Both the quantum annealer and the CBB simulation were done for the same disorder realizations (the disorder in the annealer is fully programmable) to obtain directly comparable results. We observe that there are indeed some cases in which even after 1000 repetitions, the lowest energy found by the quantum annealer is still higher than the exact value computed by CBB. Here the optimal spin configuration found by the quantum annealer typically differs markedly from the the output of our method, which detects that the quantum device probably got stuck in a local minimum. Interestingly, our method is also able to certify that, for some disorder realizations, the quantum annealer is able to find the exact ground-state energy. It does that typically in a very short time. This is true even for intermediate disorder strengths,
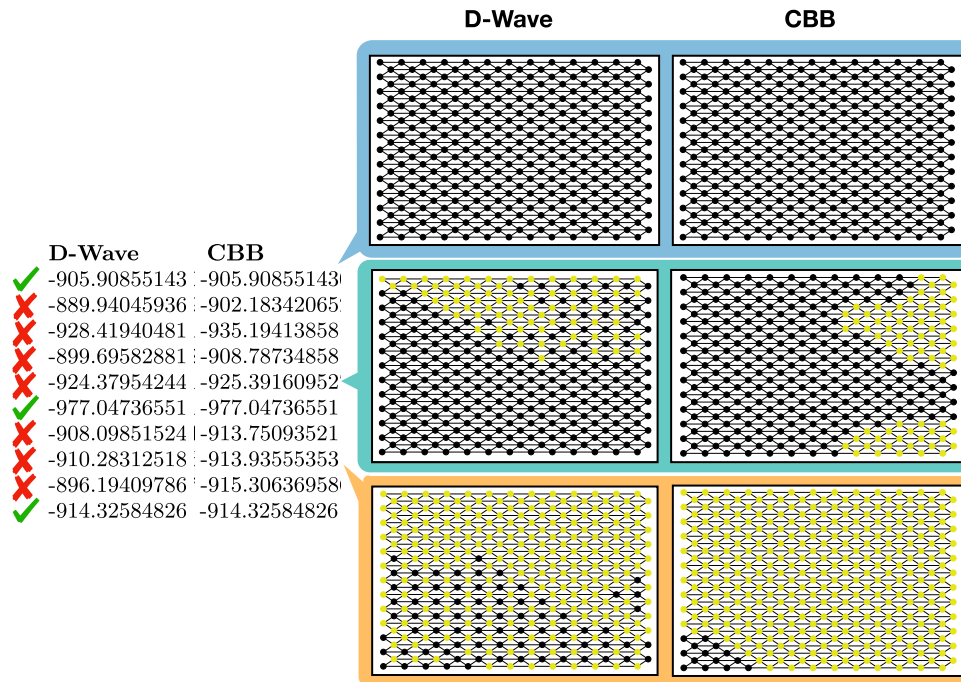
FIG. 4. Left: Comparison of the lowest energy value for a disordered phase at $\sigma = 1.5$ on a triangular lattice on a 2040-qubit D-Wave quantum annealer and the CBB algorithm. Notice that in all the instances considered here the confidence region provided by CBB converged in few steps and the algorithm returned the exact ground-state energy. Right: Comparison of the corresponding ground-state spin configuration both for a case where D-Wave achieves the lowest energy and for two cases where it does not. Yellow spins are $+1$ and black one -1. It can be seen clearly that, even when the energies provided by the two methods are close, the corresponding spin configurations can be very different. This shows that the excited state that the D-Wave quantum annealer returns does not necessarily resemble the globally optimal solution.

around $\sigma = 1.5$, where the ground-state spin pattern shows macroscopic islands of aligned spins whose precise shape and positions depend nontrivially on the disorder realization; see Fig. 4. Verifying that the annealer did reach the correct solution is impossible with standard variational approaches used so far and clearly demonstrates the relevance of the introduced CBB method for the benchmarking of quantum optimizers.

B. Towards the verification for a Chimera graph

Lastly, we consider the application of CBB to a denser graph. For this purpose we choose the Chimera architecture [23], which is the natural graph on the D-wave 2000Q hardware. The corresponding graph is composed of $K_{4,4}$ fully connected bipartite unit cells, consisting of eight spins—four horizontal and four vertical—with edges between each horizontal or vertical pair. These unit cells are arranged to form a 2D square lattice of size L and a total number of $N = 8L^2$ spins. Because of the in-cell connectivity, such a graph is clearly nonplanar and thus has the potential to encode NP-hard Ising models. Even though the Chimera graph is nonplanar and denser than 2D rectangular and triangular lattice, using the chordal extension still gives a remarkable advantage, allowing us to reach system sizes of $L = 9$, compared to just $L = 5$ (on the same hardware) for an SDP-based BB method without the chordal extension.

Although the D-Wave 200Q quantum annealer is currently implementing a Chimera graph with 2040 functional physical qubits, they are seldom actually used as logical spins. Most recent studies encode each $K_{4,4}$ cell as a single logical spin, in

order to suppress errors due to the finite size and qubit quality of the system [10]. This results in effectively solving Ising models on a 2D square lattice which, being planar, is actually proven to be polynomially solvable (cf. Appendix A). The numerical test was performed on the actual Chimera graph. This opens up the way to benchmarking future annealing devices, once their physical qubit quality has improved to a point that makes the individual spins useful, in the much more interesting regime of nonplanar graphs.

V. COMPARISON WITH OTHER VERIFICATION METHODS

The problem of verifying a quantum optimizer is very rich and has many interesting sides. We can identify two relevant players in the problem: the provider and the user. A first problem consists of verifying that the quantum hardware performs some form of quantum process, say quantum annealing, with no classical analog or that is hard to simulate classically. This type of verification is available to the provider when constructing and testing the device, but generally impossible for the user. Here the verification methods must be quantum specific. The provider may tomographically reconstruct the different quantum steps in the optimization, or may simply want to certify that the device, when solving the optimization problem, generates a large amount of some given quantum properties, such as coherence, entanglement or quantum nonlocality. This may be a direct evidence of some form of "quantumness," but it does not guarantee that the device performs better than a classical approach, as at the moment it is not clear which

quantum properties—if any—could provide a quantum computational advantage. Moreover, it is difficult to see how this approach could detect instances where the quantum device gets stuck in a local minimum, the typical challenge in the considered optimization problems.

Our work falls into a second class of methods, which attempts at verifying the device only from the outputs it produces, without any reference to the quantum process that gave raise to it. Note that, being based only on a classical output, none of these methods is quantum specific and all can be applied to any optimizer, classical or quantum. This is the type of verification that is more relevant to the user. For optimization problems, this verification will mostly be based on heuristics: most of the relevant problems are NP-hard, and it is expected that will remain hard also for quantum computers. Yet one can not exclude the possibility that quantum devices might eventually give better solutions than any classical solution "on average," or at least for some families of practically relevant problems. In fact, the search for a quantum advantage is one of the main focus of research on quantum algorithms for classical optimization problems. One can identify three ways of verifying the outputs of quantum optimizers (or, as said, any optimizer).

A. Planted solutions

In this approach, an optimizer is run on problems for which the solution is known [13,24]. This is a way of testing that the considered algorithm is able to obtain the expected solution and hope that it will perform equally well for other problems for which the solution is unknown. In our opinion, this approach is especially relevant for the development of quantum optimizers, for instance to understand which instances are difficult for them. But it also has clear limitations. First, the considered optimization problems are so diverse that it is conceivable that a device may not be able to find the exact solution for problems where it is known in advance while giving reasonably good results for other problems of relevance. Second, testing a device on problems for which the solution is known can not lead to any quantum advantage by definition. Therefore, if any quantum advantage were to be demonstrated, it will never be by running a quantum device on problems with a planted solution.

B. Variational methods

We group in this class any approach providing a candidate solution, not necessarily the optimal one, to the problem. By definition, these approaches provide upper bounds to the searched solution, as the quantum optimizer. Simulated annealing [11] or tensor-network algorithms [12] are notable classical examples of this approach. If the value provided by the quantum optimizer E_q, is larger than the best value obtained by one of these classical approaches, E_c, one can certify that the quantum device has not reached the searched solution, E_g. Classical variational methods have a long tradition and can deal with very large systems, at the moment larger than those available for quantum optimizers. If, however, quantum optimizers will ever reach the quantum advantage, one will encounter a situation in which they will be able to provide

the smallest available upper bound to the solution. In a resulting scenario where $E_g \leqslant E_q < E_c$, the limitations of classical variational methods are clear: it is hard to determine whether the better performances of quantum optimizers are an indication of to their intrinsically higher computational power or the lack of a more efficient classical optimization algorithm.

C. Relaxations

This is the approach considered in this work and, more in general, we refer here to any method providing a lower bound to the searched ground-state energy (note, however, that our approach can also be used to provide an upper bound with the same computational effort). These methods have much less history and cannot reach, at the moment, problem sizes comparable to variational methods. However, because of their complementarity, they have clear merits. As shown by our demonstration using the D-Wave machine, these methods can give a termination criterion for any optimization heuristics, certifying that the searched solution has been reached and no more rounds are needed. But even when a gap remains, the lower bounds obtained through relaxations can be complemented with the best upper bound, being quantum or classical, and provide an energy range in which the solution provably lies. Clearly, no quantum advantage is possible for those problems where the obtained lower bound matches the solution obtained by a classical optimizer. On the contrary, problems where there is a gap between the best known classical solution and the lower bound may be good candidates when searching for a quantum advantage. Finally, if the bound only matches the output produced by a quantum device, no classical approach will ever provide a strictly better solution. In our opinion these properties make this approach valuable and, in particular, especially relevant to advance the study and search for a quantum advantage in classical optimization problems.

VI. CONCLUSIONS

We introduced the chordal branch-and-bound (CBB) method that uses a hierarchy of efficiently computable upper and lower bounds on the ground-state energy of classical spin systems and exploits the physical locality structure of relevant Hamiltonians. Our numerical results show that the iterative branch-and-bound process often converges, providing an exact and certified value for the ground-state energy, together with a ground-state configuration. Even for those cases where convergence is not met, the method always provides with a polynomial effort an energy range where the searched solution provably lies. Moreover, the obtained lower bound can be used to benchmark the output of both quantum and classical optimization methods. We focused here on their use to assess the quality of current quantum annealers beyond their comparison to classical variational approaches. In particular, we were able to identify instances were the quantum optimizer reached the actual ground-state energy without resorting to planted solution problems. To our knowledge, to date this is the only approach providing this type of certification.

It would definitely be interesting to explore the performance of our method on other relevant Ising models that are proven to be NP-hard. The complexity of the model might

result in an exponential number of branch-and-bound steps to achieve convergence to the ground-state energy. However, let us stress that NP-hardness is a worst-case feature, hence it might be the case that an efficient convergence to the ground states is achievable on average even for some NP-hard models. In this respect, CBB can be essential to provide numerical evidence that identifies which Ising models are uniformly hard, among the NP-hard ones.

From a general viewpoint, recent progress of quantum computing devices [14] urge for the development of methods to benchmark them. Our work is an example of such effort and we are confident that the approach adopted here for benchmarking, based on relaxations of the initial problem, may find applications in other similar contexts.

APPENDIX A: COMPLEXITY OF FINDING ISING GROUND STATES

There is a wealth of results on the worst-case complexity of finding the ground state of various Ising models [25]. Thereby "worst-case complexity" is the complexity of the hardest instances within a class of families of problems of increasing size. How hard it is to solve the ground-state problem of such a family varies with the interaction graph and can crucially depend on seemingly unimportant details. We consider Hamiltonians that are polynomials (with fixed finite degree and finite precision coefficients) in the spin variables in the form

$$H(\vec{\sigma}) = \sum_{1 \leqslant i < j \leqslant N} J_{ij} \sigma_i \sigma_j + \sum_{i=1}^{N} h_i \sigma_i, \qquad \text{(A1)}$$

with couplings J_{ij} and local fields h_i, and interaction (hyper-)graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$. The size of a problem is the number of vertices $N := |\mathcal{V}|$. We say that an Ising model of the form (A1) has no fields if $h_i = 0$ for all i, we say it has an external field if all $h_i = h$ for all i and some h, and we say it has on-site fields if all h_i can be chosen independently.

Finding the ground state of Hamiltonians of the form (A1) for arbitrary graphs is in general NP-hard [26], even without any fields. This is still true for $J_{i,j} \in \{-1, 0, 1\}$ and $\mathcal{G}$ a finite 3D cubic grid graph and even for $\mathcal{G}$ a cubic two-layer 2D grid [26]. In contrast, for planar graphs $\mathcal{G}$ and without local fields, the ground state can be found efficiently even without the restriction $J_{i,j} \in \{-1, 0, 1\}$ [25]. With the restriction $J_{i,j} \in \{-1, 0, 1\}$ this even holds for toroidal graphs (grids on a torus, i.e., systems with periodic boundary conditions) [25]. Similarly, if $J_{i,j} \geqslant 0$, then even some systems with local fields can be solved in polynomial time [25]. On the other hand, for general planar graphs with interactions $J_{i,j} = 1$ and uniform external field $h_i = 1$, finding the ground state is again NP-hard [26]. Reference [25] contains a list of further concrete models whose complexity is either known to be in P or proven to be NP-hard.

These hardness results are typically obtained by reducing the ground-state problem to the so-called max-cut problem, which is known to the NP-hard. The polynomial time algorithms to solve the other families of systems, in turn, work by finding perfect matchings [26] or rely on so-called max-flow and min-cut methods [25].

APPENDIX B: INGREDIENTS OF THE CHORDAL BRANCH-AND-BOUND METHOD

In this Appendix we describe in detail the Lasserre hierarchy and the chordal extension method used to derive lower bounds to the ground-state energy of a classical Hamiltonian. To make this part self-contained, we revise the general method by using the notation used in the main text, while making comparisons to the more typical framework of polynomial optimization. Moreover, for the sake of completeness we will repeat some notions that have already been introduced in the main text.

We begin by recalling that a state of an N spin system is a probability distribution P over the set of configurations $\{-1, 1\}^N$. Expectation values of a function f for the state P are denoted by $\langle f \rangle_P$, or even shorter by $\langle f \rangle$ if the connection to a specific states is not evident. The ground state is defined by any state P achieving the minimal possible expectation value for the Hamiltonian, i.e., $\min_P \langle H \rangle_P = E_g$:

$$H(\vec{\sigma}) = \min_P - \sum_{1 \leqslant i < j \leqslant N} J_{ij} \langle \sigma_i \sigma_j \rangle_P + \sum_{i=1}^{N} h_i \langle \sigma_i \rangle_P. \qquad \text{(B1)}$$

1. The hierarchy of lower bounds

Let $\vec{x}^{(\nu)}$ be the vector of all monomials of spin variables of degree up to ν. For such vector and state P we define its moment matrix $\Gamma^{(\nu)}(P)$ as the $k \times k$ matrix of expectation values $\Gamma_{\alpha\beta}^{(\nu)}(P) := \langle x_\alpha^{(\nu)} x_\beta^{(\nu)} \rangle_P$. It is not hard to see that for any state P, any moment matrix $\Gamma^{(\nu)}(P)$ is positive semidefinite, i.e., $\Gamma^{(\nu)}(P) \succeq 0$, and that, depending on what the elements of $\vec{x}^{(\nu)}$ are, it further obeys certain linear constraints that reflect the two basic properties of classical spin variables of taking dichotomic values $\sigma_i \in \{-1, 1\}$ and commuting with each other. These properties imply relations among different expectation values such as, for example, $\langle \sigma_i \sigma_j \sigma_i \rangle_P = \langle \sigma_j \rangle_P$,

which can be expressed in terms of some matrices F_m such as $\mathrm{tr}(F_m \Gamma^{(\nu)}(P)) = 0$.

For a sufficiently large value of ν, the corresponding moment matrix $\Gamma^{(\nu)}$ contains all the expectation values needed for the computation of the energy, in the sense that there is a matrix h [depending on the J_{ij} and h_i in case the Hamiltonian is of the form in (A1)], such that for any physical state P it holds that $\langle H \rangle_P = \mathrm{tr}(h\,\Gamma^{(\nu)}(P))$. If this is the case, for that given ν, one can relax the ground-state problem by minimizing the energy over all matrices $\Gamma^{(\nu)}$ that are positive semidefinite and fulfill the above mentioned linear constraints, rather than over those that can actually arise from a physical state P. The resulting optimization problem reads

$$E_g^{(\nu)} = \min_{\Gamma^{(\nu)}} \mathrm{tr}(h\,\Gamma^{(\nu)})$$

$$\text{s.t.} \qquad \Gamma^{(\nu)} \succeq 0, \qquad \text{(B2)}$$

$$\forall m \in \{1, \ldots, k\}: \mathrm{tr}(F_m\,\Gamma^{(\nu)}) = 0.$$

The solution to this minimization problem $E_g^{(\nu)}$ provides a lower bound on the true ground-state energy, i.e., $E_g^{(\nu)} \leqslant E_g$. This follows from the fact that the set of all matrices $\Gamma^{(\nu)}$ that are positive semidefinite and fulfill the above mentioned linear constraints is larger than the set of moment matrices that can actually arise from a physical state P. It is further obvious that

the $E_g^{(\nu)}$ values are ordered in the sense that $E_g^{(\nu)} \leqslant E_g^{(\nu+1)}$ for any ν. Remarkably, if all the relevant F_m are taken into account, the bounds actually converge to the true ground-state energy for any fixed finite system size and Hamiltonian H, in the sense that $\lim_{\nu \to \infty} E_g^{(\nu)} = E_g$ [16,27]. This does not mean that convergence can only be attained in the limit. As a matter of fact, there are situations in which convergence is attained at a finite value of ν.

To illustrate the first levels of the hierarchy, let us go back to the example presented in the main text with moment matrix

$$\Gamma = \begin{pmatrix} 1 & \langle\sigma_1\rangle & \langle\sigma_2\rangle & \ldots & \langle\sigma_N\rangle \\ \langle\sigma_1\rangle & 1 & \langle\sigma_1\sigma_2\rangle & \ldots & \langle\sigma_1\sigma_N\rangle \\ \langle\sigma_2\rangle & \langle\sigma_1\sigma_2\rangle & 1 & \ldots & \langle\sigma_2\sigma_N\rangle \\ \vdots & \vdots & & \ddots & \\ \langle\sigma_N\rangle & \langle\sigma_1\sigma_N\rangle & \ldots & & 1 \end{pmatrix}. \qquad \text{(B3)}$$

It is clear now that that moment matrix represents the hierarchy (B2) at level $\nu = 1$. If we take the special case of $N = 3$ spins, the whole moment matrix reads

$$\Gamma^{(1)} = \begin{pmatrix} 1 & \langle\sigma_1\rangle & \langle\sigma_2\rangle & \langle\sigma_3\rangle \\ \langle\sigma_1\rangle & 1 & \langle\sigma_1\sigma_2\rangle & \langle\sigma_1\sigma_3\rangle \\ \langle\sigma_2\rangle & \langle\sigma_1\sigma_2\rangle & 1 & \langle\sigma_2\sigma_3\rangle \\ \langle\sigma_3\rangle & \langle\sigma_1\sigma_3\rangle & \langle\sigma_1\sigma_3\rangle & 1 \end{pmatrix}. \qquad \text{(B4)}$$

Going at level at level $\nu = 2$ for a system of $N = 3$ spins, the corresponding moment matrix take the following form:

$$\Gamma^{(2)} = \begin{pmatrix} 1 & \langle\sigma_1\rangle & \langle\sigma_2\rangle & \langle\sigma_3\rangle & \langle\sigma_1\sigma_2\rangle & \langle\sigma_1\sigma_3\rangle & \langle\sigma_2\sigma_3\rangle \\ \langle\sigma_1\rangle & 1 & \langle\sigma_1\sigma_2\rangle & \langle\sigma_1\sigma_3\rangle & \langle\sigma_2\rangle & \langle\sigma_3\rangle & \langle\sigma_1\sigma_2\sigma_3\rangle \\ \langle\sigma_2\rangle & \langle\sigma_1\sigma_2\rangle & 1 & \langle\sigma_2\sigma_3\rangle & \langle\sigma_1\rangle & \langle\sigma_1\sigma_2\sigma_3\rangle & \langle\sigma_3\rangle \\ \langle\sigma_3\rangle & \langle\sigma_1\sigma_3\rangle & \langle\sigma_2\sigma_3\rangle & 1 & \langle\sigma_1\sigma_2\sigma_3\rangle & \langle\sigma_1\rangle & \langle\sigma_2\rangle \\ \langle\sigma_1\sigma_2\rangle & \langle\sigma_2\rangle & \langle\sigma_2\rangle & \langle\sigma_1\sigma_2\sigma_3\rangle & 1 & \langle\sigma_2\sigma_3\rangle & \langle\sigma_1\sigma_3\rangle \\ \langle\sigma_1\sigma_3\rangle & \langle\sigma_3\rangle & \langle\sigma_1\sigma_2\sigma_3\rangle & \langle\sigma_1\rangle & \langle\sigma_2\sigma_3\rangle & 1 & \langle\sigma_1\sigma_2\rangle \\ \langle\sigma_2\sigma_3\rangle & \langle\sigma_1\sigma_2\sigma_3\rangle & \langle\sigma_3\rangle & \langle\sigma_2\rangle & \langle\sigma_1\sigma_3\rangle & \langle\sigma_1\sigma_2\rangle & 1 \end{pmatrix}. \qquad \text{(B5)}$$

Matrix $\Gamma^{(1)}$ is contained in $\Gamma^{(2)}$ as a principal minor—the one generated by the first four rows and columns. This makes it so that the second level directly implies the constraint implied by the first one, while the nonnegativity of a bigger moment matrix results in a generally more stringent test. Moreover, the expectation value of any Hamiltonian of the form given in (A1) can be expressed as a function of the entries of both moment matrices as $\mathrm{tr}(h\,\Gamma^{(\nu)}) = J_{12}\Gamma_{15}^{(\nu)} + J_{13}\Gamma_{16}^{(\nu)} + J_{23}\Gamma_{17}^{(\nu)} + h_1\Gamma_{12}^{(\nu)} + h_2\Gamma_{13}^{(\nu)} + h_3\Gamma_{14}^{(\nu)}$ for $\nu = 1, 2$. Similarly, one can see how the linear constraints F_m on the entries reflect the properties of the spin variables. Dichotomy directly imposes the conditions $\Gamma_{ii}^{(\nu)} = 1$ on the diagonal variables and, combined with commutations, allows to identify some of the entries, such as for example $\Gamma_{12}^{(1)} = \Gamma_{21}^{(1)}$ and $\Gamma_{13}^{(2)} = \Gamma_{25}^{(2)}$.

As a comparison with previous similar methods, notice that the branch-and-bound technique introduced in Ref. [17] exploits a lower bound method that is almost equivalent to the first level of the relaxation (B2), with the addition of some hand crafted linear constraints. Indeed, the mentioned relaxation can be obtained by considering a moment matrix generated by the set of monomials composed of the spin

variables only, without the identity operator. Hence, it results in the first level of the Lasserre hierarchy, diminished by the absence of the one (the first) row of the matrix. In contrast, the hierarchy discussed here allows to systematically construct an infinite family of increasingly precise relaxations that yield better and better bounds at the price of an increasing computational cost.

2. Exploiting sparsity via the chordal extension

Depending on the kind of system considered, the optimization problem defined by the Hamiltonian (A1) can be sparse. As is shown in Refs. [18,19], we can exploit this sparsity to derive a more scalable relaxation than (B2).

The method works as follows: take the dependency graph G of the problem and check if it is chordal. As mentioned in the main text, a graph is said to be chordal if all its cycles of four or more vertices have a chord, i.e., an edge that is not part of the cycle but connects two vertices of the cycle. If G is not chordal, construct a so-called chordal extension G_C of G by suitably adding edges until the graph is chordal. The chordal extension of a graph is not unique, but a chordal

extension can always be found, simply because any fully connected graph is chordal. As it will became clear in the following, for our purposes it is crucial to find a chordal extension that is still relatively sparse. A polytime method that works well in this respect for all the cases studied here is to compute an approximate minimum degree ordering of the graph nodes, followed by Cholesky factorization of a positive semidefinite matrix with the associated sparsity pattern [28]. Once a specific chordal extension G_C is constructed, it will contain a number of n_C maximal cliques $C_l \subset \mathcal{V}$. A clique, that is a fully connected subgraph, is maximal if it cannot be extended by including any other adjacent vertex. Since the graph G represents a sparse Hamiltonian, and G_C is obtained from G by simply adding some edges, the function (A1) can be decomposed into a sum $H = \sum_l H_{C_l}$ of terms that each contain only variables contained in a given maximal clique C_l.

One can then modify the optimization problem in (B2) as follows: Replace the big $\Gamma^{(\nu)}$ matrix by a direct sum of smaller matrices $\Gamma_l^{(\nu)}$, one for each clique, constructed from the spin variables belonging to the clique C_l. Some spin variables appear in more than one clique, which can be captured with additional linear constraints that involve variables of the blocks $\Gamma_l^{(\nu)}$. Writing this explicitly, the chordal version of the SDP relaxation then reads as

$$
\begin{aligned}
\min_{\Gamma_l^{(\nu)}} \ & \sum_n \mathrm{tr}\left(h_n \Gamma_n^{(\nu)}\right) \\
\text{s.t. } \ & \Gamma_l^{(\nu)} \succeq 0, \quad \forall \, l = 1, \dots, n_C, \\
& \mathrm{tr}\left(F_{m,l} \Gamma_l^{(\nu)}\right) = 0 \quad m = 1, \dots, k_l \,, l = 1, \dots, n_C, \\
& \sum_n \mathrm{tr}\left((G_{l,n}) \Gamma_n^{(\nu)}\right) = 0 \quad l = 1, \dots, k, \quad \text{(B6)}
\end{aligned}
$$

where the $F_{m,l}$ are the intrablock constraints coming from the properties of the spin variables, while the $G_{l,n}$ correspond to the constraints identifying expectation values belonging to different blocks. Interestingly this relaxation still converges to the exact result [19].

Depending on the sparsity of the graph G (and its chordal extension G_C), substituting the original optimization relaxation (B2) by (B6) leads to a substantial simplification and improved scaling in runtime and memory. In practical applications, the latter are typically dominated by the the largest block, i.e., the largest maximal clique in G_C. Moreover, the block structure can be exploited to have a more finely tuned control of the lower bound precision, essentially by replacing a general hierarchy level ν by a moment matrix with block-dependent levels ν_l. This allows to define hybrid levels where, for instance, smaller blocks are generated at higher values of ν_l, thus improving the quality of the lower bound without significantly affecting the scalability of the SDP.

Let us illustrate how this chordal extended relaxation works in practice by going back to the three spins example introduced above and by considering the 1D Ising model with the Hamiltonian $H = \sum_{i=1}^{2} J_{i,i+1} \sigma_i \sigma_{i+1}$ presented above. The corresponding dependency graph G is already chordal and is composed of two cliques $C_1 = \sigma_1, \sigma_2$ and $C_2 = \sigma_2, \sigma_3$. Since the blocks at level $\nu = 1$ have been presented in the main text,

here we show the relaxation at level $\nu = 2$, where the matrix (5) can be substituted by the two blocks:

$$
\Gamma_{C_1}^{(2)} = \begin{pmatrix}
1 & \langle \sigma_1 \rangle & \langle \sigma_2 \rangle & \langle \sigma_1 \sigma_2 \rangle \\
\langle \sigma_1 \rangle & 1 & \langle \sigma_1 \sigma_2 \rangle & \langle \sigma_2 \rangle \\
\langle \sigma_2 \rangle & \langle \sigma_1 \sigma_2 \rangle & 1 & \langle \sigma_1 \rangle \\
\langle \sigma_1 \sigma_2 \rangle & \langle \sigma_2 \rangle & \langle \sigma_1 \rangle & 1
\end{pmatrix}, \quad \text{(B7)}
$$

$$
\Gamma_{C_2}^{(2)} = \begin{pmatrix}
1 & \langle \sigma_2 \rangle & \langle \sigma_3 \rangle & \langle \sigma_2 \sigma_3 \rangle \\
\langle \sigma_2 \rangle & 1 & \langle \sigma_2 \sigma_3 \rangle & \langle \sigma_3 \rangle \\
\langle \sigma_3 \rangle & \langle \sigma_2 \sigma_3 \rangle & 1 & \langle \sigma_2 \rangle \\
\langle \sigma_2 \sigma_3 \rangle & \langle \sigma_3 \rangle & \langle \sigma_2 \rangle & 1
\end{pmatrix}. \quad \text{(B8)}
$$

The constraints $G_{l,n}$ derive from the fact that the variable σ_2 belongs to both cliques, hence several expectation values appear in both blocks. Some of the unnecessary expectation values in (5), such as $\langle \sigma_1 \sigma_3 \rangle$ and $\langle \sigma_1 \sigma_2 \sigma_3 \rangle$ no longer appear in the two smaller blocks $\Gamma_{C_1}^{(2)}$ and $\Gamma_{C_2}^{(2)}$. Such a simplification is particularly useful because it reduces the number of variables involved in the SDP.

APPENDIX C: IMPLEMENTATION OF THE CHORDAL BRANCH-AND-BOUND METHOD

We now describe in detail the implementation of our chordal branch-and-bound (CBB) method. First, we elaborate on the three ingredients: the lower bound, the upper bound, and the branching rule. Last, we comment on possible improvements of our strategy.

1. Lower bound

The cheapest method to get a bound on the ground-state energy from below is to use the relaxation in (B6) at the lowest level ν that contains all the expectation values needed for the computation of the energy. In this work we focus on quadratic Hamiltonian functions, hence $\nu = 1$. However, in practical applications we observe that this often leads to a lower bound that can be more than 10% away from the corresponding upper bound. As already mentioned in Ref. [17], having such a big initial gap slows down the convergence of the branch and bound, making it very difficult to reach a point in the branching where the lower bound is high enough to start excluding the first branches. This problem is overcome by tightening the relaxation, which can be done in several ways.

The option considered in Ref. [17] was to tighten the relaxation at level 1 by adding hand crafted linear inequalities, so-called triangle inequalities, between entries of the matrix corresponding to the two-body correlations $\langle \sigma_i \sigma_j \rangle$ of triples of spin variables. Since the amount of all these possible constraints scales as N^3, usually only a fraction of them is introduced. In our CBB we can exploit the structure of the problem to obtain more systematic improvements. The chordal extension reduces the amount of meaningful constraints that can be added. Indeed, the resulting block structure implies that the only two-body expectation values $\langle \sigma_i \sigma_j \rangle$ that appear in the moment matrices correspond to spins i, j belonging to the same block. Hence, all the triangle inequalities that can actually be imposed have to involve triples i, j, k that appear in the same clique.

The numerical effort for one step in the CBB method is mostly determined by the largest block in the moment matrix.

Therefore, we choose to take a hybrid approach, introducing an intermediate level with $\nu = 2$ for all the blocks $\Gamma_l^{(\nu)}$ involving less than n_t variables, while keeping all the bigger blocks at level 1. Taking such a hybrid level yields a significant improvement in the initial lower-upper bound gap already for smaller values of n_t. In fact, we have checked numerically that this devises a test that corresponds at least to the case of level 1 plus the addition of all triangle inequalities between variables in the smaller blocks.

Moreover, we also allow for additional triangle constraints between two-body correlations belonging to bigger blocks. In particular, we add them in an iterative way, as shown in Ref. [17], until the improvement on the lower bound is smaller than some numerical precision. In most cases we tested, there was actually no need to introduce these additional constraints.

2. Upper bound

For the upper bound one needs a good guess for a spin configurations that is close to the ground-state energy. In our case, this is straightforward: from the moment matrices $\Gamma_l^{*(\nu)}$ obtained from the solution of the SDP (B6), construct the configuration $\vec{\sigma}^*$ where each spin σ_i^* is aligned according to the sign of the entry corresponding to the expectation value $\langle \sigma_i \rangle$. Intuitively, this can be seen as a way to obtain the spin configurations "closest" to the optimal (but typically unphysical) solution achieved by the relaxation. A nice feature of our strategy for the upper bound is that it basically adds no extra computational cost as it is derived directly from the moment matrix that is obtained by solving the SDP.

The above method makes a substantial improvement over earlier approaches. Indeed, as described in Ref. [17], the moment matrix used in previous SDP relaxations is missing the first row and column vector, and thus exactly the entries we need to extract our deterministic configuration. That is why former approaches usually resort on performing a Cholesky decomposition of the moment matrices $\Gamma_l^* = B_l^T B_l$ and, by interpreting each row of the resulting matrices B_l as a vector $v_{l,j}$, assigning the deterministic values to the spin variables by taking scalar products of these vectors with a randomly chosen one. Notice that, apart from requiring the additional computational effort of having to perform a Choleski decomposition, the above method is also hard to adapt to an SDP composed of more blocks, as the one resulting from the application of the chordal extension technique.

To conclude, once a valid spin configuration $\vec{\sigma}^*$ has been extracted, we simply set the upper bound to be its corresponding energy $H(\vec{\sigma}^*)$. Surprisingly, we noticed that by following this procedure, the exact ground states is usually recovered very soon in the branching (see Fig. 1 for an example). It then takes additional time to find a matching lower bound to verify that this is indeed the lowest achievable energy. This makes us believe that our procedure is very efficient in finding the optimal deterministic configuration.

3. Branching rule

Here we follow the same method outlined in Ref. [17], but with a different choice of branching procedure. Indeed, those authors choose the dichotomic choice to be the relative alignment of a pair of connected spins. That is, given a choice of indices i, j, the two branches correspond to the two cases $\sigma_i \pm \sigma_j = 0$. However, as mentioned before, we prefer to branch on the value of the single spin, by choosing between the two values $\sigma_i = \pm 1$. The reason for this is that, in the latter case the number of possible branching steps depends only on the number N of spins in the system. On the contrary, the former method involves an amount of branching choices that depends on the number of edges in the dependency graph G, which can be much higher, often as high as N^2.

For our branching rule strategy, there is the question of which spin i to choose for the next branching. The way we do this here is based on the expectation values $\langle \sigma_i \rangle$ recovered from the moment matrices $\Gamma_l^{(\nu)}$ and used for the construction of the upper bound. The intuition is the following: Spins with an expectation value close to zero are "difficult" choices, because flipping the value of such a spin is likely to lead to a slight change in the energy of the system; conversely, expectations values very close to ± 1 are identified as "easy" choices, because flipping such a spin is likely to lead to a significant change in the energy of the system and it is easier to discard a branch during the evolution of the branch-and-bound process. We set the branching rule to "easy-first," that is, at the end of each optimization round, the next branching is performed on the closest to deterministic spin in the $\Gamma_l^{(\nu)}$.

4. Possible improvements

There is some freedom in the choices we outlined in the previous subsections. Given the huge difference in complexity that can be exhibited by various instances of the Ising model, we expect the optimal choice to be model dependent. Here we briefly discuss which modifications we imagine to be most useful for practical applications.

Let us start by recalling that, in order to accelerate the convergence process and to keep memory requirements low, it is crucial to reduce the initial lower-upper bound gap as much as possible and as early as possible. One way to do that is to modify the hybrid hierarchy level introduced above. In our applications, it was always enough to set the threshold to at most $n_t = 7$. However, such value can be significantly increased without affecting too much the scalability of the CBB. Indeed, the main bottleneck of our method is the memory required to solve the largest SDP. This depends mainly on the block l^* leading to the largest matrix $\Gamma_{l^*}^{(\nu)}$. Therefore, as long as increasing the level of the smaller blocks does not lead to bigger matrices that the one for the largest clique, the SDP will still have the same memory requirements—although the solving time will clearly increase.

Other branching rules can be also be adopted. For instance, one can replace the "easy-first" approach with a "difficult-first." In this case, one picks the next branching from the least deterministic spin in the $\Gamma_l^{(\nu)}$. We expect the choice of the most effective branching rule to depend on the system under consideration.

Last, there are instances in which CBB does not outperform other methods. This is true for specific cases of very sparse problems, where linear-programming relaxations were shown to work very well [29], or some hand-crafted models for which exact polynomial algorithms are known [30].
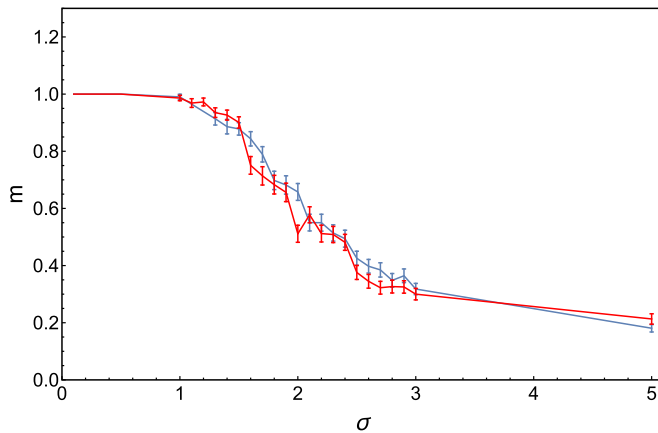
FIG. 5. Average magnetization m of the ground state of the ferromagnetic Ising model with disordered magnetic field, as a function of the disorder strength σ, for a 2D triangular lattice of linear size L. The plot shows the cases $L = 15$ (blue) and $L = 20$ (red). The phase transition could be pinned down more precisely by means of a finite-size scaling analysis. For our purposes here it is sufficient to know that in the range $\sigma \in [1, 3]$, the ground states has nontrivial spin patterns.

Nevertheless, it would still be interesting to see if one could combine the construction based on the chordal extension with those methods and provide some further advantage.

APPENDIX D: PHASE TRANSITION IN THE FERROMAGNETIC DISORDERED ISING MODEL IN A 2D TRIANGULAR LATTICE

Here we use the CBB method to study in more detail the 2D triangular Ising model used to benchmark the solution of the D-wave quantum annealer. Recall that we considered a ferromagnetic model in a statically disordered mangetic field, represented by the following Hamiltonian

$$H(\vec{\sigma}) = - \sum_{\langle i, j \rangle} \sigma_i \sigma_j + \sum_{i=1}^{N} h_i \sigma_i, \qquad \text{(D1)}$$

where the first sum runs over all pairs of connected spins $\langle i, j \rangle$ in the triangular lattice. The local fields h_i are drawn randomly from a Gaussian distribution with zero mean and variance σ for each site i. As a function of the disorder strength σ, the model undergoes a phase transition from a ferromagnetic ground state (in which all states are aligned with each other) to a disordered phase (in which, for extremely large disorder, the spins are aligned with the local magnetic fields). The transition can be detected by estimating the ground-state magnetization

$$m = \frac{\left| \sum_{i=1}^{N} \sigma_i^{(g)} \right|}{N}, \qquad \text{(D2)}$$

where $\vec{\sigma}^{(g)}$ is the ground-state configuration. Clearly, a ferromagnetic ground states is fully magnetized, hence it is characterized by $m = 1$. On the other hand, a disordered phase corresponds to $m \approx 0$ (notice that, because of finite-size effects, one can never reach an exactly zero value in numerical tests).

We analyzed the phase transition by computing the ground-state energy for the model with the CBB method and estimating its magnetization m as a function of the disorder strength σ. For each value of σ, the results were averaged over 100 samples. Figure 5 shows the obtained magnetization curves, for two different linear lattice sizes $L = 15, 20$.

[1] D. Koller, N. Friedman, L. Getoor, and B. Taskar, Graphical models in a nutshell, in *Introduction to Statistical Relational Learning*, edited by L. Getoor and B. Taskar (MIT Press, Cambridge, MA, 2007), pp. 13–56.

[2] M. R. Bhatt and U. B. Desai, Robust image restoration algorithm using Markov random field model, CVGIP: GraphModels Image Proc. **56**, 61 (1994).

[3] A. Lucas, Ising formulations of many NP problems, Front. Phys. **2**, 5 (2014).

[4] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nat. Commun. **5**, 4213 (2014).

[5] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, Tech. Rep., MIT, arXiv:1411.4028 (2014).

[6] G. E. Crooks, Performance of the quantum approximate optimization algorithm on the maximum cut problem, arXiv:1811.08419 (2018).

[7] Quantum Neural Network on Cloud, https://www.nii.ac.jp/en/news/release/2017/1120.html (2017).

[8] 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).

[9] E. Crosson and A. W. Harrow, Simulated quantum annealing can be exponentially faster than classical simulated annealing, in *2016 IEEE 57th Annual Symposium on Foundations of Computer Science* (IEEE, New York, 2016), Vol. 2016-Decem, pp. 714–723.

[10] S. Mandrá and H. G. Katzgraber, A deceptive step towards quantum speedup detection, Quant. Sci. Technol. **3**, 04LT01 (2018).

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

[12] R. Orus, A practical introduction to tensor networks: Matrix product states and projected entangled pair states, Ann. Phys. **349**, 117 (2014).

[13] 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).

[14] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[15] J. B. Lasserre, An explicit equivalent positive semidefinite program for nonlinear 0-1 programs, SIAM J. Optim. **12**, 756 (2002).

[16] J. B. Lasserre, Global optimization with polynomials and the problem of moments, SIAM J. Optim. **11**, 796 (2001).

[17] F. Rendl, G. Rinaldi, and A. Wiegele, Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations, Math. Program. **121**, 307 (2008).

[18] H. Waki, S. Kim, M. Kojima, and M. Muramatsu, Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity, SIAM J. Optim. **17**, 218 (2006).

[19] J. B. Lasserre, Convergent SDP-relaxations in polynomial optimization with sparsity, SIAM J. Optim. **17**, 822 (2006).

[20] P. Wittek, Algorithm 950: Ncpol2sdpa—Sparse semidefinite programming relaxations for polynomial optimization problems of noncommuting variables, ACM Trans. Math. Softw. **41**, 21 (2015).

[21] Mosek, MOSEK optimization software (2010).

[22] https://gitlab.com/FBaccari/Chordal_BB.

[23] P. I. Bunyk, E. M. Hoskinson, M. W. Johnson, E. Tolkacheva, F. Altomare, A. J. Berkley, R. Harris, J. P. Hilton, T. Lanting, A. J.

[24] D. Perera, I. Akpabio, F. Hamze, S. Mandra, N. Rose, M. Aramon, and H. G. Katzgraber, Chook—A comprehensive suite for generating binary optimization problems with planted solutions, arXiv:2005.14344.

[25] M. Grötschel, M. Jünger, and G. Reinelt, Calculating exact ground states of spin glasses: A polyhedral approach, in *Heidelberg Colloquium on Glassy Dynamics*, edited by J. L. van Hemmen and I. Morgenstern, Lecture Notes in Physics, Vol. 275 (Springer, Berlin, Heidelberg, 1987), pp. 325–353.

[26] F. Barahona, On the computational complexity of Ising spin glass models, J. Phys. A: Math. Gen. **15**, 3241 (1982).

[27] S. Pironio, M. Navascués, and A. Acín, Convergent relaxations of polynomial optimization problems with noncommuting variables, SIAM J. Optim. **20**, 2157 (2010).

[28] L. Vandenberghe and M. S. Andersen, Chordal graphs and semidefinite optimization, Found. Trends Optim. **1**, 241 (2015).

[29] F. Liers, M. Jünger, G. Reinelt, and G. Rinaldi, Computing exact ground states of hard Ising spin glass problems by branch-and-cut, in *New Optimization Algorithms in Physics*, edited by A. K. Hartmann and H. Rieger (Wiley-VCH, 2004), pp. 47–69.

[30] S. Mandrá, H. G. Katzgraber, and C. Thomas, The pitfalls of planar spin-glass benchmarks: Raising the bar for quantum annealers (again), Quantum Sci. Technol. **2**, 038501 (2017).

Przybysz *et al.*, Architectural considerations in the design of a superconducting quantum annealing processor, IEEE Trans. Appl. Supercond. **24**, 1 (2014).