Practical engineering of hard spin-glass instances

Jeffrey Marshall,[1,2,*] Victor Martin-Mayor,[3,4] and Itay Hen[2,5]

[1]*Department of Physics and Astronomy, University of Southern California, Los Angeles, California 90089, USA*
[2]*Center for Quantum Information Science & Technology, University of Southern California, Los Angeles, California 90089, USA*
[3]*Departamento de Física Teórica I, Universidad Complutense, 28040 Madrid, Spain*
[4]*Instituto de Biocomputación y Física de Sistemas Complejos (BIFI), Zaragoza, Spain*
[5]*Information Sciences Institute, University of Southern California, Marina del Rey, California 90292, USA*

(Received 19 May 2016; published 14 July 2016)

Recent technological developments in the field of experimental quantum annealing have made prototypical annealing optimizers with hundreds of qubits commercially available. The experimental demonstration of a quantum speedup for optimization problems has since then become a coveted, albeit elusive goal. Recent studies have shown that the so far inconclusive results, regarding a quantum enhancement, may have been partly due to the benchmark problems used being unsuitable. In particular, these problems had inherently too simple a structure, allowing for both traditional resources and quantum annealers to solve them with no special efforts. The need therefore has arisen for the generation of harder benchmarks which would hopefully possess the discriminative power to separate classical scaling of performance with size from quantum. We introduce here a practical technique for the engineering of extremely hard spin-glass Ising-type problem instances that does not require "cherry picking" from large ensembles of randomly generated instances. We accomplish this by treating the generation of hard optimization problems itself as an optimization problem, for which we offer a heuristic algorithm that solves it. We demonstrate the genuine thermal hardness of our generated instances by examining them thermodynamically and analyzing their energy landscapes, as well as by testing the performance of various state-of-the-art algorithms on them. We argue that a proper characterization of the generated instances offers a practical, efficient way to properly benchmark experimental quantum annealers, as well as any other optimization algorithm.

DOI: 10.1103/PhysRevA.94.012320

I. INTRODUCTION

Many problems of theoretical and practical relevance consist of searching for the global minimum of a cost function. These optimization problems are on the one hand notoriously hard to solve but on the other hand ubiquitous, and appear in diverse fields such as machine learning, material design, and software verification, to mention a few diverse examples. The computational difficulties associated with solving these optimization problems stem from the intricate structure of the cost function that needs to be optimized which often has a rough landscape with many local minima. The design of fast and practical algorithms for optimization has therefore become one of the most important challenges of many areas of science and technology [1].

Recent theoretical and technological breakthroughs have triggered an enormous interest in one such nontraditional method, commonly referred to as quantum annealing (QA) [2–8]. The uniqueness of this approach stems from the fact that it does not rely on traditional computation resources but rather manipulates data structures called quantum bits, or qubits, that obey the laws of quantum mechanics. It is believed that by utilizing uniquely quantum features such as entanglement, massive parallelism, and tunneling, a quantum computer can solve certain computational problems in a way which scales better with problem size than is possible on a classical machine.

A huge amount of progress has recently been made in the building of experimental quantum annealers [9,10], the most notable of which are the D-Wave processors consisting of hundreds of coupled superconducting flux qubits. These devices offer a very natural approach to solving optimization problems utilizing gradually decreasing quantum fluctuations to traverse the barriers in the energy landscape in search of global optima [2,3,5–7]. As an inherently quantum technique, QA holds the so-far unfulfilled promise to solve combinatorial optimization problems faster than traditional "classical" algorithms [11–14]. However, to date there is no experimental (nor theoretical) evidence that quantum annealers are capable of producing such speedups [15,16].

Extensive studies designed to properly benchmark experimental QA processors, such as the D-Wave annealers, have resulted for the most part in inconclusive results, despite accumulating evidence for the (indirect detection) of genuinely quantum effects such as entanglement and multiqubit tunneling [17–21]. Indeed, direct comparison tests between the 128-qubit D-Wave One and later the 512-qubit D-Wave Two (DW2) processors and classical state-of-the-art algorithms on randomly generated Ising-model instances have shown no evidence of a quantum speedup [15,16,19].

The above lack of evidence has motivated a few recent studies to further explore problem classes where one might expect the occurrence of quantum speedups [16,22–24]. Katzgraber, Hamze, and Andrist [22] pointed out that the random Ising instances used in the previously mentioned comparison tests exhibit a spin-glass phase transition only at $T = 0$, i.e., at zero temperature. Spin glasses are disordered, frustrated spin systems that may be viewed as prototypical classically hard (also called NP-hard) optimization problems, that are so challenging that specialized hardware has been built

[*] jsmarsha@usc.edu

to simulate them [25–27] [the related cost function is in Eq. (1) below]. That the spin-glass transition occurs at $T = 0$ implies that, for any $T > 0$, the energy landscapes for these problems are in general fairly simple and can therefore be solved rather easily by classical heuristic solvers and hence do not require quantum tunneling reach global optima, thus rendering these instances less than ideal for benchmarking.

A subsequent study which examined the same class of uniformly random Ising problems on the D-Wave architecture [23] measured the correlation between the performance of the DW2 device and a physical effect referred to as *temperature chaos* [28–43], which has recently been identified as the culprit for the difficulties that classical thermal algorithms encounter when attempting to solve spin glasses [44,45]. Temperature chaos implies the presence of low-lying excited states (i.e., slightly suboptimal spin assignments) that have a large Hamming distance with respect to the minimizing assignment, or ground state (GS), of the instance. Furthermore, these excited states are not only stable against local excitations (i.e., bit flips), they also have a much larger entropy than the GS. As a consequence, classical state-of-the-art optimization algorithms such as simulated annealing or parallel tempering simulations often get trapped in one of these excited states. Indeed, nonchaotic problem instances are exponentially unlikely to be found as the problem size is increased [39,42,43]. However, while temperature-chaotic instances do indeed exist on the relatively tiny DW2 512-bit hardware graph, they become exceedingly rare with the degree to which they exhibit temperature chaos, and are therefore difficult to find [23].

Since these are the temperature-chaotic instances that are expected to have the discriminative power to separate classical scaling of performance with size, from quantum, a natural question thus arises. Can one efficiently find or generate “rare gem” instances, i.e., small size problems (so small that encoding it on a quantum device is feasible) that also display a large degree of temperature chaos, or inherent hardness? To date, several techniques for generating hard problems that go beyond random generation of instances have been explored, such as utilizing instances with planted solutions with tunable frustration [16], the deliberate reduction of GS degeneracy using Sidon sets for the couplings [46], or the porting of fully connected Sherrington-Kirkpatrick (SK) instances [24]. However, the obtained instances were found to lack the necessary degree of inherent hardness (i.e., hard problems are still rare), which as a result necessitated the generation of an initial huge pool of problems followed by the prohibitively expensive procedure of exhaustive “mining” for instances presenting high degrees of inherent thermal hardness [23,46].

In this paper, we propose an altogether different, adaptive algorithm to generate such rare gem instances—extremely hard spin-glass instances of relatively small size—in a considerably more efficient manner than current mining techniques. The remainder of the work is organized as follows. In Sec. II we present the basics of a heuristic algorithm which generates hard spin-glass instances. Following this, in Sec. III, we thermodynamically analyze the generated instances, and test them against classical optimizers, as well as the DW2 annealer. We also apply our technique to instances with planted solutions in Sec. IV and conclude in Sec. V.

II. ENGINEERING OF HARD SPIN-GLASS BENCHMARKS

For concreteness in what follows we apply our technique to the D-Wave Two “Chimera” hardware graph (see Appendix A) as this will allow us to experimentally test the hardness of the instances on an actual quantum annealer. However, it should be noted that the method proposed here is far more general and may apply to arbitrary connectivity graphs. The cost function on which we generate our instances is of the form

$$H = \sum_{\langle i,j \rangle} J_{ij} s_i s_j, \tag{1}$$

where the couplings $\{J_{ij}\}$ are programmable parameters that define the instance. The cost function H is to be minimized over the spin variables, $s_i = \pm 1$, where $i = 1 \dots N$ and N is the number of participating spins. The angle brackets $\langle i,j \rangle$ denote that the sum is only over connected bits on the Chimera graph.

In this work, we shall treat the process of finding problem instances, i.e., sets of $\{J_{ij}\}$ values, over any predetermined set of allowed values in a way that maximizes the hardness of the problem (however hardness is defined), as an optimization problem in itself. The figure of merit—or cost function—for this optimization problem is any faithful characteristic of the inherent hardness of the instance. We will call this figure of merit the *time to solution*, or TTS for short.

Here, we shall use as the TTS of a problem instance the definition for classical thermal hardness that was introduced in Ref. [23], namely, the characteristic number of steps it takes for a parallel tempering (PT) algorithm to equilibrate. In a PT algorithm, one simulates N_T realizations of an N-spin system, with temperatures $T_1 < T_2 < \cdots < T_{N_T}$, where Metropolis updates occur independently for each copy. Each copy attempts to swap temperatures with its temperature neighbors, with probabilities satisfying detailed balance [47]. The resulting temperature random walk of each system copy allows a global traversal of the configuration space, as well as detailed exploration of local minima (i.e., at the lower temperatures). An accurate PT simulation takes time longer than the temperature “mixing” time, τ [48,49]. The time τ can be thought of as an equilibration time; the time for each copy to explore the entire temperature mesh. Thus large τ instances take longer to equilibrate, motivating the definition of the mixing time τ as the *classical hardness*.

Furthermore, we shall utilize the strong correlation found between the PT mixing time τ and the hardness of other algorithms (consistently with the requirement of intrinsic hardness [23]). Specifically, we shall use as TTS the runtime clocked by the Hamze–de Freitas and Selby algorithm (HFS) [50,51], which has proved to be much faster yet strongly correlated with the classical hardness, τ [23]. Our aim in this work is the *maximization* of the TTS cost function where the variables over which the maximization is done are the coupling strengths $\{J_{ij}\}$ of the underlying graph.

A. Random adaptive optimization (RAO)

Heuristic optimization algorithms, such as Metropolis and simulated annealing, aim to find the global minimum (equivalently, maximum) of a cost function by changing the state of the system at each step. Changes are accepted when the cost moves in the required direction, but also, often crucially, still accept changes in the "wrong" direction with a certain probability so as to reduce the chances of becoming stuck in local minima. This acceptance probability may be determined by defining a simulated "temperature" parameter, β (inspired by thermal annealing). We follow this approach with the cost being the TTS (which is assumed to be an indicator of inherent hardness), and the "state" of the system a particular configuration of the J_{ij}.

By picking some random "seed" instance, and modifying a subset of the J_{ij} (e.g., flipping the sign of a random edge), the resulting instance may be harder to solve as determined by the TTS cost. If this is the case, we accept the modification. Repeating this process will necessarily drive the system to harder and harder instances. If the TTS is lowered by such a modification, it may still be accepted, so as to avoid getting trapped in local minima. We utilize a Boltzmann-type acceptance probability, $e^{-\beta|\Delta\mathrm{TTS}|}$ [52], where the choice of β defining this distribution will depend on the solver being used to determine the TTS, and the manner in which one updates β during the algorithm (if at all), will depend on the methodology one wishes to pursue (e.g., Metropolis, simulated annealing, etc.).

We outline our algorithm in its most basic (unoptimized) form in Algorithm 1, which generates hard signed (i.e., $J_{i,j} = \pm 1$) instances, though we wish to stress that the general technique can be applied under much more diverse settings. In fact, we expect that allowing the coupling constants, J_{ij}, to take on a wider range of values will in general result in harder instances being generated (compared to the $J_{ij} = \pm 1$ case). However, this is not necessarily indicative of a greater efficiency of the RAO algorithm, as higher range (random) instances are known to be harder to solve, compared to $J_{ij} = \pm 1$ instances [15]. Additionally, as an example of this diversity, we have also utilized a "reversed" version of the algorithm, tweaked to *minimize* the TTS in order to generate particularly easy instances as well. Moreover, we apply this method to instances with planted solutions in Sec. IV. In the next section we illustrate the effectiveness of our technique.

Algorithm 1. Random adaptive optimization (RAO).

```
1: procedure GENERATEHARDPROBLEM
2:     Generate random $\pm 1$ seed instance and get TTS
3:     for step = 1 to NSTEP do
4:         Pick a random edge and flip sign
5:         Get new TTS
6:         if TTS increases then
7:             Accept Change
8:         else
9:             Accept with probability $e^{-\beta|\Delta\mathrm{TTS}|}$
10:        end if
11:        Update $\beta$ if required
12:    end for
13: end procedure
```

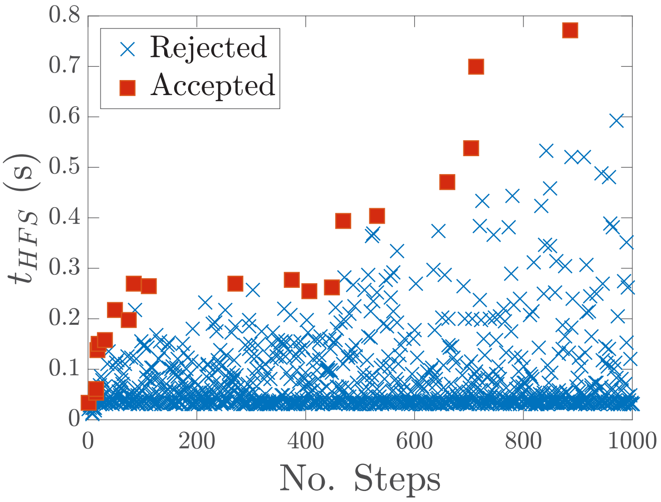

FIG. 1. Performance of the RAO algorithm for a single instance. A single run of the algorithm over 1000 steps, set up to maximize t_{HFS} of a 512-bit Chimera instance with $J_{ij} = \pm 1$. Updates consist of flipping the value of randomly picked edges. Squares (red) show successful update attempts; crosses (blue) are rejected updates.

III. RESULTS

A. Engineering of hard spin-glass instances

For our work, we used extensively the version of the HFS algorithm created by Selby [51,53], and have taken the average wallclock time for the TTS [54] as our cost function. We adopt the notation t_{HFS} for this quantity (see Appendix B for specific implementation details) [55].

The performance of one typical realization of our algorithm on a 512 bit Chimera-type instance is depicted in Fig. 1. Remarkably, the final instance is just 20 successful steps away from the initial, and the final TTS is about 25 times the initial instances TTS. Though there does not seem to necessarily be a typical (or standard) output for the algorithm, the occurrence of plateaus is found to be fairly common. Indeed in Fig. 1 we see that the instance remains at a plateau of about 0.25 s between 100 and 400 attempted flips. These plateaus can occasionally halt the optimization of the cost function, and as such it is important to carefully choose an appropriate simulated temperature.

In Fig. 2, we statistically quantify the merit of the generated hard instances by comparing the final TTS to the mean initial TTS (i.e., typical TTS of a random instance), after 0 (blue), 500 (red), and 2500 (yellow, with red outline) adaptive steps on 150 instances. One notices immediately a clear separation in hardness classification from the completely random instances (blue), and the other two groups, even after a fairly modest number of update attempts (i.e., 500). The instances after 2500 steps are on average about three times harder than those after 500 steps, which are themselves about an order of magnitude above the random instances.

To gain a reverse effect, namely, easier than random instances, we have also run our algorithm by "reversing" the acceptance criterion (i.e., step 6 in Algorithm 1) such that it favors instances with shorter TTS values. This allowed for

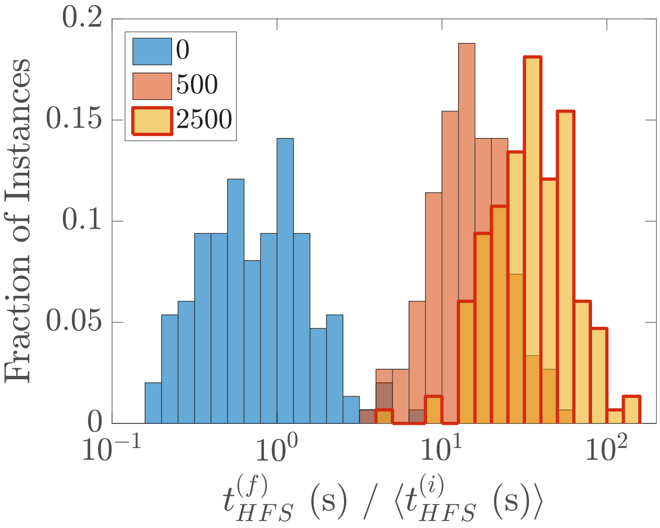

FIG. 2. Histogram of the ratio of final to mean initial t_{HFS} for 150 random signed instances (512 bit Chimera graph). As in Fig. 1 we adapted the instances by flipping random edge values, attempting to maximize t_{HFS}. This plot is a normalized histogram of the ratio final TTS, $t^{(f)}_{HFS}$, to mean initial TTS, $\langle t^{(i)}_{HFS}\rangle$, after 0 (blue), 500 (red), and 2500 (yellow, with red outline) algorithmic steps.

lowering the TTS on the DW2 Chimera by a factor of about 10 from randomly chosen instances (see next subsection).

B. Inherent (thermal) hardness of the generated instances

It is crucially important to demonstrate that the generated instances are not only difficult to solve with respect to the solver with the help of which they are obtained, but that the instances are *inherently* difficult, i.e., they possess inherent degrees of hardness. In what follows we illustrate precisely that by measuring the thermodynamical complexity of the instances, generated using our RAO algorithm.

To that aim, we have used t_{HFS} as both a minimizing and maximizing cost function, to generate 100 signed ($J_{ij} = \pm 1$) 500-bit DW2 Chimera instances in each of four different hardness groups, or generations, classified by $t_{HFS} \in [0.8,1.2] \times 10^{-4+k}$ s, where $k = 1,2,3,4$ [56]. The $k = 1$ instances are about an order of magnitude easier compared to random instances. The hardest instance we have found on the studied graph, after analyzing 780 instances, was found to be ~250 times harder than a typical random instance, with a runtime of $t_{HFS} \approx 6.0$ s on a 3.5 GHz single core CPU, which to our knowledge is the most difficult randomly generated HFS instance on the DW2 graph to date. We shall denote this instance by $k_{\max}$.

Our first task is assessing how difficult it is for state-of-the-art classical thermal algorithms such as PT to solve the generated instances. This question can be quantitatively answered by computing the mixing time, τ, for the temperature random walk of the algorithm [42,47–49] and examine its correlation with "hardness group" k.

Our computation of τ follows exactly the procedure detailed in Ref. [23]. We simulated 120 system copies consisting of four independent parallel-temperature chains, with 30 temperatures each. Given the similar system size, we also use the same temperature grid of Ref. [23]. The elementary Monte Carlo step (EMCS) consisted of 10 full-lattice Metropolis sweeps, independently performed in each system copy, followed by one parallel tempering temperature-exchange sweep. A computation of τ was considered satisfactory if two conditions were met. (i) The system copy (out of the 120 possibilities) that spends the least time in the hot-half region (i.e., the 15 highest temperatures) spends at least 20% of the total simulation time there. In other words, no system copy got permanently trapped in the cold-half region. (ii) The total simulation time was at least 20τ long. τ is given in units of Metropolis sweeps.

We performed three independent simulations of different lengths: 10^6 EMCS (i.e., 10^7 Metropolis sweeps), 10^7 EMCS, and 10^8 EMCS. The shortest runs (10^6 EMCS), were enough to compute τ for the 200 problem instances that belong to the first and second hardness groups ($k = 1,2$). The 10^7 EMCS run sufficed to compute τ for most (but not all) of the third-generation instances. The 10^8 EMCS run was enough to compute τ for all the third-generation instances, and for 86 out of 100 problem instances belonging to the fourth generation. As a cross-check, we compared the GS energy found with parallel tempering with the one found with the HFS code. Agreement was reached in all cases (even in cases where the computation of τ was not successful).

Specifically notable are 32 of the instances of the hardest ($k = 4$) HFS group, which were found to have $\tau > 10^7$. In Ref. [23] it was estimated that only 2 out of every 10^4 random instances of this type have $\tau > 10^7$. This observation helps to quantify the efficiency of our algorithm; a highly optimized PT algorithm screening random instances of this type would require ~65 CPU h [57] to find one with $\tau > 10^7$. Equivalently for (unoptimized) RAO, we estimate less than 5 h to generate one such instance [56].

Having classified (most of) these 400 instances by the PT mixing time, we analyze their energy landscapes by computing the overlap between their GS configurations and their dominant first-excited states. The overlap between two spin assignments, $\bar{s}_1,\bar{s}_2$, is defined as $1 - 2h(\bar{s}_1,\bar{s}_2)/N$, where h is the Hamming distance. This analysis is summarized in Fig. 3. The trend in the figure is rather clear: a large τ (and k) correlates strongly with a smaller overlap (i.e., large Hamming distance) between the ground and dominant first-excited states; that is, the larger τ, the more difficult the problems are in a *thermodynamical* sense. Interestingly, the easiest HFS instances, $k = 1$, which have been generated by minimization of TTS have been found to not correlate as well with the other data groups. We examine this in more detail in the next subsection.

C. Algorithmic scaling

To establish the inherent hardness of the instances generated by the RAO algorithm, we have directly compared their time-to-solution t_{HFS} to the PT mixing time, τ. This is depicted in Fig. 4. Despite the apparent fluctuations, we observe an agreement between these two vastly different solvers, which can be quantified by the dependence $\tau \sim t^{1.4}_{HFS}$ (as measured by the median data point for the $k = 2,3,4$ groups). This correlation has in fact allowed us to generate 14 instances (out of the 100 of the $k = 4$ group) with $\tau > 5\times10^7$ EMCS,

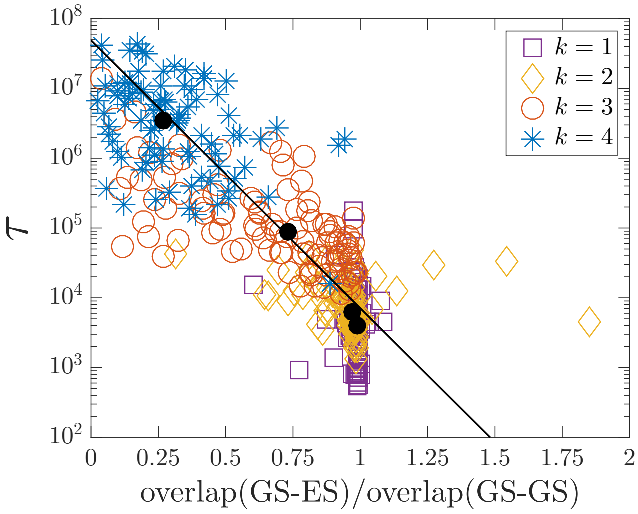

FIG. 3. Mixing time τ vs ratio of typical GS-ES overlaps over GS-GS overlaps. For each instance for which PT computed τ successfully, we compute the median overlap (see text) between the ground and dominant first-excited states, normalizing by the median ground state to ground state overlap. Also plotted is the median data point for each group (filled black circle), where going from bottom right to upper left is in order from $k = 1$ to $k = 4$. Linear fit is on the median data point for $k = 2,3,4$. τ is given in units of Metropolis sweeps.

which using straightforward "mining" would have required the generation and subsequent analysis of more than 2×10^6 randomly generated instances (as found by Ref. [23])—about 100 times more costly in terms of computational resources [56,57]. Comparing this to the numerics quoted in the previous subsection (regarding generating $\tau > 10^7$ instances), we see RAO becomes even more beneficial over conventional methods as the problem difficulty bar is raised, and is very likely to be the *only* way to obtain a large number of such temperature-chaotic instances.

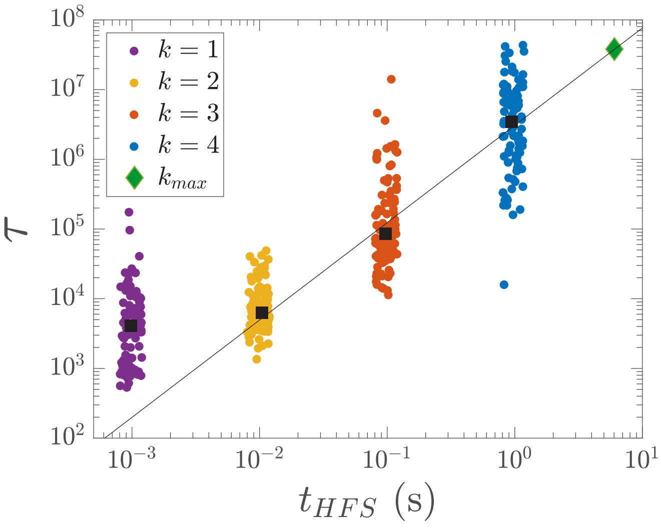

FIG. 4. PT hardness (mixing time) vs HFS hardness. The 400 instances generated as documented in the main text, examined on parallel tempering (PT). The linear fit of slope 1.40 is obtained from a least-squares fitting on the median data point (black squares) of each data group, ignoring the $k = 1$ group. We also include, indicated by a green diamond marker, the hardest instance ($k_{\max}$) found using our adaptive algorithm, extrapolated using the linear fit to obtain the corresponding value for $\tau \approx 3.7\times10^7$. τ is given in units of Metropolis sweeps.

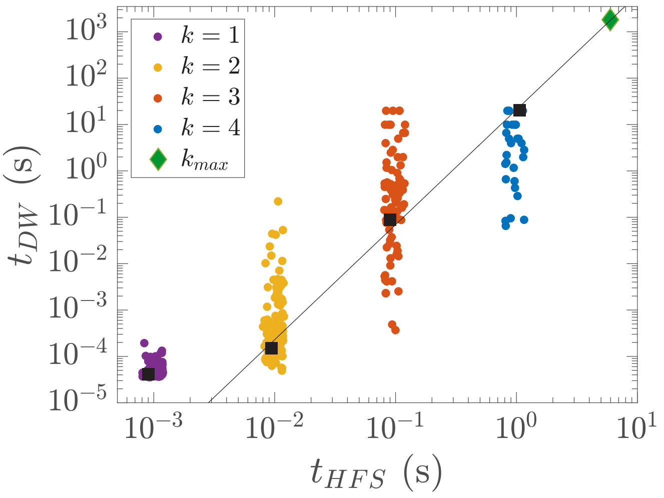

FIG. 5. DW2 hardness vs HFS hardness. We plot the D-Wave TTS, t_{DW}, against t_{HFS} for the 400 problem instances as explained in the main text. Note that there are fewer blue data points ($k = 4$), compared to the others. This is because many of these instances were not solved once by the D-Wave machine in 660 000 attempts and so are left off of this graph. The linear fit of slope 2.48 is obtained from a least squares fitting on the lower quartile data point (black squares) of each data group, excluding the $k = 1$ group. We also include, indicated by a green diamond marker, the hardest instance ($k_{\max}$) found using our adaptive algorithm, extrapolated using the linear fit to obtain the corresponding value for $t_{DW} \approx 10^3$ s, i.e., require $\sim10^8$ attempts.

We perform a similar analysis using the D-Wave Two QA optimizer as the comparison platform. We define TTS as measured by DW2, t_{DW}, as the anneal time divided by the probability of successfully finding the GS. To establish probabilities of success, we ran each of the 400 instances on the D-Wave processor for roughly 650 000 anneals with individual anneal times in the range 20–40 μs. For the hardest instances according to HFS ($k = 4$), about 75% of the instances were not solved even once by DW2. For this reason we use the lower quartile as a representative data point in Fig. 5.

Here too, as with the PT comparison, large variations in the data are observed. Nonetheless, we see a strong correlation between the two solvers on average (as we expect), with $t_{DW} \sim t_{HFS}^{2.48}$. That DW2 scales unfavorably with HFS hardness as compared to how the scaling of PT mixing time suggests that the QA chip may be detrimentally affected by "classical causes" such as thermal hardness of instances.

The hardest HFS instance found, $k_{\max}$, neatly demonstrates the capabilities of the RAO algorithm applied to this particular graph. Instances of this type we estimate to have $\tau \approx 3.7\times10^7$, and $t_{DW} \approx 10^3$ s (that is, would require $\sim10^8$ DW2 anneals).

As mentioned above, the HFS "easiest" instances ($k = 1$) do not seem to correlate with the other data groups (as we also see in Figs. 3 and 4). In fact the reason here is trivial; since t_{DW} has a minimum equal to the median annealing time, 30 μs $\approx 3\times10^{-5}$ s in this work, the easiest instances accumulate at this value, as is seen in Fig. 5. We believe there may be an equivalent scenario occurring for PT, i.e., practical lower bound on τ (though, note, quantifying such a bound is nontrivial).

These above correlation analyses all suggest that instances generated using our RAO algorithm are indeed inherently more difficult optimization problems, compared with randomly generated instances. In particular, RAO problems are more akin to rare events (i.e., the instances displaying the strongest temperature chaos in a large set of randomly chosen problems). For example, the hardest RAO instances, indexed here by $k = 4$ (equivalently, $\tau_{HFS} \approx 10^0$ s), are in general harder for the D-Wave Two quantum annealer, as well as for parallel tempering algorithms. Moreover, we see that (Fig. 3) these instances are thermodynamically more difficult, with larger Hamming distances between the ground and dominant first-excited states.

IV. HARD BENCHMARKS WITH KNOWN GROUND-STATE CONFIGURATIONS

The generation of instances with random couplings does not allow us in general to know the GS energy of the instances with certainty—an important feature when carrying out comparison tests. Therefore, this section is devoted to another adaptive technique, building on work from Ref. [16], which generates hard instances, but also crucially allows for knowledge of the GS energy, without having to resort to exact solvers.

We apply our method to Ising-type instances with *planted solutions*—an idea borrowed from constraint satisfaction (SAT) problems [16,58,59]. Instances of this type are constructed around some arbitrary solution, by splitting the full graph up into smaller subgraphs, i.e., the Hamiltonian is written as a sum of small subgraph Ising Hamiltonians, $H = \sum_{j=1}^{M} H_j$. The coupling values of each sub-Ising Hamiltonian are chosen so that the planted solution is a simultaneous GS of all of the H_j, and therefore is also a GS of the total Hamiltonian H. This knowledge circumvents the need for exact (provable) solvers, which rapidly become too expensive computationally as the number of variables grows, and as such is very suitable for benchmarking. In what follows, we shall choose our subgraph Hamiltonians to be randomly placed *frustrated* cycles, or loops, along the edges of the hardware graph [16] such that no configuration of the variables simultaneously minimizes all terms in the cost function (see Fig. 1 of [16] for examples of Hamiltonian loops on the DW2 graph). This frustration is known to often cause classical algorithms to get stuck in local minima, since the global minimum of the problem satisfies only a fraction of the Ising couplings and/or local fields [44,45].

Unlike the signed $J_{ij} = \pm 1$ problems studied above, planted-solution problems allow for the computation of certain measures of frustration (the reader is referred to Ref. [16] for further details, and results pertaining to this approach in the context of benchmarking of experimental quantum annealers). By combining the planted solution technique with the RAO method, one can generate instances which are harder than is possible to generate by simply using randomly placed loops on the graph, with the added benefit of still knowing the GS energy.

We initialize the setup as in the above, i.e., we first pick a random planted solution, and place M random sub-Hamiltonian loops (either frustrated or not) on the graph which satisfy the solution. This method allows us to easily calculate the GS energy as a sum of the individual loop energies with respect to the planted solution. At variance with the RAO method, update attempts now instead of involving single random edges, involve Hamiltonian loops. We remove a random loop from the instance and add a new random loop, making sure to keep track of the GS energy, and making sure the new loop respects the planted solution.

Algorithm 2. Loop adaptive optimization (LAO).

```
procedure GenerateFrustratedProblem
    Generate (random) solution
    Place M random loops on graph, each respecting the
  planted solution
    Calculate GS energy
    for step + 1 to NSTEP do
        Remove random loop from current instance
        Pick new random loop and add, respecting planted
  solution
        Get new TTS
        if TTS increases then
            Accept change, update GS energy
        else
            Accept with probability $e^{-\beta|\Delta\mathrm{TTS}|}$
            Update GS energy if accepted
        end if
    end for
end procedure
```

One now has many different parameters affecting the performance, e.g., the total number of loops in the instance, the ratio of different sized loops (e.g., one can use a mix of size 4 and 6 loops, etc.), and different (positive) weights on the loops. One can also scrutinize the position of each loop to try to maximize, e.g., the frustration (note that randomly adding loops can have the affect so as to cancel out frustration). Also, of course, similar comments about adjusting the algorithm as mentioned in the RAO section still apply here.

In Fig. 6 we perform one such version of our LAO algorithm on 100 instances, and compare the final TTS to the typical TTS a random instance. While there is a general increase in problem difficulty, from that of a random instance, it is by far less than the equivalent figure for random signed instances (see Fig. 2). Note however that planted-solution problems may be tuned in numerous ways (see previous paragraph) to provide varying degrees of hardness, thereby altering the structure of the problem (in fact, they may be tuned to be harder than random signed instances [16]). Thus the choice of parameters, which we have not optimized here, may heavily affect the performance of LAO. Nevertheless, we have demonstrated the successful application of our main algorithm to instances with planted solutions, allowing for the generation of instances that are about an order of magnitude more difficult.

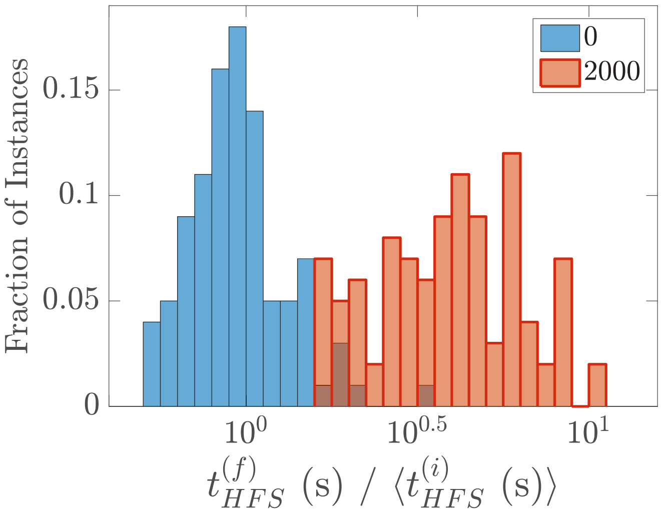

FIG. 6. Histogram of the ratio of final to mean initial t_{HFS} for 100 planted-solution instances (504-bit Chimera graph). We compare the LAO algorithm after 2000 steps (red) to the 100 random initial (blue) planted-solution instances, each containing 350 random loops. Updates consisted of adding and removing random loops, where size 4 (6) loops have a probability of 0.1 (0.9) being chosen, with integer loop weight chosen randomly in range [1,5]. We plot the histogram of the ratio final HFS TTS, $t^{(f)}_{HFS}$, after 0 (blue) and 2000 (red) LAO steps, to the mean initial TTS, $\langle t^{(i)}_{HFS}\rangle$. The mean TTS ratio after the 2000 steps is ≈4.3, and the maximum final TTS is ≈0.15 s.

V. SUMMARY AND CONCLUSIONS

We developed a technique to practically engineer extremely hard optimization problems to address the challenge of generating proper benchmarks for the testing of experimental quantum annealers. This was accomplished by treating the generation of hard problem instances as an optimization problem and subsequently devising a heuristic optimization algorithm to solve it. We demonstrated that one can successfully engineer Ising-type optimization problems with varying degrees of difficulty, defined by some suitable choice of problem hardness. To establish and confirm the inherent hardness of the instances, we measured the correlation between various independent measures of problem difficulty, in particular, from parallel tempering configurations we computed a Hamming distance measure between the ground and main first-excited states.

We illustrated the ability to generate signed (i.e., $J_{ij} = \pm 1$), 512-bit instances for the D-Wave Two Chimera which are greater in difficulty (as measured by the TTS of a very successful classical HFS solver) by more than two orders of magnitude compared to randomly generated instances. As designed, these instances were found to be more difficult both for the DW2 processor and for parallel tempering algorithms to solve. We have further shown that our technique is significantly faster than straightforward mining for hard instances which requires the generation and subsequent costly analysis of very many random instances.

Since in designing benchmark tests it is often desirable, and necessary, to know with certainty the GS energy of the instances used, we have devised an adaptive technique which also allows one to generate hard instances for which a GS is known, based on problems with planted solutions [16]. While in this case the method is somewhat less effective, it nonetheless allows one to easily generate problem instances that are rare to find by random generation of instances.

The generation of hard instances is one of the key tools to understand some fundamental, but unanswered questions in spin-glass theory as well as in the field of quantum annealing. For example, what makes certain problems hard? What are the most reliable ways to classify problem difficulty? Is there a marked difference between quantum and classical hardness? The techniques presented in this work may be further utilized to the end of observing the elusive quantum speedup (provided that there could be one). The adaptive generation of hard instances may be further leveraged to systematically study the properties (geometric, thermodynamical, or otherwise) of the resultant instances, paving the way towards the systematic generation of inherently hard spin-glass instances.

By simultaneously updating an instance, one may further try to use as a figure of merit that is to be maximized the ratio of the "classical TTS" to quantum (e.g., experimental annealer) TTS. This may hopefully allow for the engineering of instances which are classically hard but quantum easy. These may consequently be studied so as to enhance our understanding of the differences between quantum and classical hardness.

ACKNOWLEDGMENTS

We thank Ehsan Khatami for useful comments and suggestions. This work was partially supported by MINECO (Spain) through Grants No. FIS2012-35719-C02, No. FIS2015-65078-C2-1-P. Computation for the work described in this paper was partially supported by the University of Southern California's Center for High-Performance Computing (http://hpcc.usc.edu).

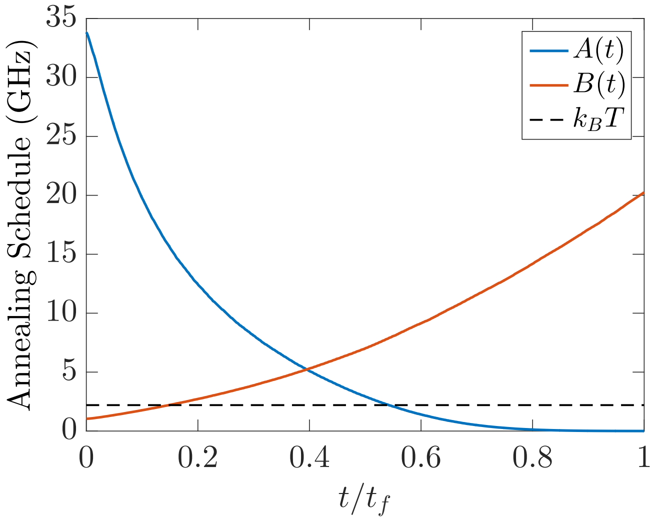

FIG. 7. Annealing schedule of the DW2 device. The annealing curves $A(t)$ and $B(t)$ are calculated using rf-SQUID models with independently calibrated qubit parameters. Units of $\hbar = 1$. The operating temperature of 17 mK is also shown.

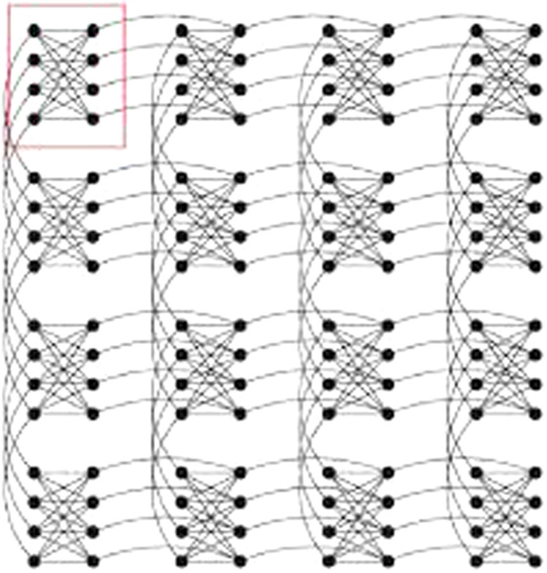

FIG. 8. 4×4 patch of the full 8×8 Chimera graph for the DW2 chip. Top left (red) shows a single $K_{4,4}$ bipartite graph. The qubits or spin variables occupy the vertices (circles) and the couplings J_{ij} are along the edges. Of the 512 qubits, 504 were operative in our experiments.

APPENDIX A: D-WAVE TWO ANNEALER

The D-Wave Two (DW2) is marketed by D-Wave Systems Inc. as a quantum annealer, which evolves a physical system of superconducting flux qubits according to the time-dependent Hamiltonian

$$H(t) = A(t)\sum_{i\in V}\sigma_i^x + B(t)H_p, \quad t \in [0,t_f], \tag{A1}$$

with H_p given in Eq. (1). The annealing schedules given by $A(t)$ and $B(t)$ are shown in Fig. 7. Our experiments used the DW2 device housed at the USC Information Sciences Institute, with an operating temperature ≈17 mK. The Chimera graph of the DW2 used in our work is shown in Fig. 8. Each unit cell is a balanced $K_{4,4}$ bipartite graph. In the ideal Chimera graph (of 512 qubits) the degree of each vertex is 6 (except for the corner unit cells). In the actual DW2 device we used, 504 qubits were functional.

APPENDIX B: IMPLEMENTATION DETAILS

All of our numerical results were obtained using Selby's (heuristic) version of HFS [51,53] (i.e., with input settings -S3 -m1), running on a single core of an Intel Xeon CPU E5-1650 v2 running at 3.50 GHz. With these settings, the code will halt only after it has found agreement in the lowest energy for Eq. (1), $n+1$ consecutive times (option -p set to n in Selby's code), over independent HFS sweeps [Selby's code technically solves quadratic unconstrained binary optimization (QUBO) problems, but the mapping from Ising problems of this sort is trivial].

We define the HFS time to solution, t_{HFS} in the main text, as $t_{HFS} := \lim_{n\to\infty} T_{\text{step}}/(n+1)$, where T_{step}, depending on n, is the physical wallclock run time of Selby's code, for a single QUBO instance (also see [54]). That is, T_{step} is the time taken to find agreement in the lowest energy $n+1$ times in a row, and as such we define this quantity as $T_{\text{step}} := \sum_{i=0}^{n} t_i$, where t_i is the time taken to find ith ($i > 0$) occurrence of the (presumed) minima, and t_0 the time to first detect this minima. In practice, to obtain a reasonable estimate of t_{HFS} one should take a "large" value for n, e.g., $n > 500$.

We face two practical issues: (1) Fixing some value for n, and adapting the instance under RAO, of course means that the wallclock runtime of each step T_{step} of RAO increases as the problem becomes harder, meaning for a large number of RAO steps, the algorithm may take a very long time to complete. (2) We noticed that in addition to t_{HFS} increasing, so do the differences, $|\Delta t_{HFS}|$, and, as such, the acceptance probability, $e^{-\beta|\Delta t_{HFS}|}$, may quickly become negligible. We provide a quick and easy solution to these two problems, by varying just one parameter, n. This is by no means an optimal solution, but it has enabled us to generate many hard instances with fewer computational resources compared to what would otherwise be required.

By defining a cutoff return time, $T_{\max}$ (we picked $T_{\max} = 3$ s), such that if $T_{\text{step}} > T_{\max}$, we reduce $n \to n/2$ (note we never let n be lower than 16, and initialize it with $n = 512$). This allows us to control the growth of T_{step}, and hence bound (at least somewhat) the total runtime of the RAO algorithm. The accuracy in the estimation of t_{HFS} of course decreases as n decreases; therefore, one should run the final instance (i.e., the instance after adapting it as per the RAO algorithm) with some large choice of n.

In addition, we let β depend linearly on n, so that as t_{HFS} increases (hence $|\Delta t_{HFS}|$ too), β is lowered, and as such we can control the acceptance probability (again, at least somewhat). Our particular choice used to generate the 100 hardest instances ($k = 4$) for the results section was $\beta = 6.5n$. The value 6.5 is not particularly special; our version of RAO seemed to perform best with this choice over a small trial of other values.

[1] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover Books on Computer Science (Dover Publications, Mineola, NY, 2013).

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

[4] R. Martoňák, G. E. Santoro, and E. Tosatti, Quantum annealing by the path-integral Monte Carlo method: The two-dimensional random Ising model, Phys. Rev. B **66**, 094203 (2002).

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

[6] J. Brooke, D. Bitko, T. F. Rosenbaum, and G. Aeppli, Quantum annealing of a disordered magnet, Science **284**, 779 (1999).

[7] 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).

[8] B. W. Reichardt, The quantum adiabatic optimization algorithm and local minima, in *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04 (ACM, New York, 2004), p. 502.

[9] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. S. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose, Quantum annealing with manufactured spins, Nature (London) **473**, 194 (2011).

[10] 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. Przybysz, and J. Whittaker, Architectural considerations in the design of a superconducting quantum annealing processor, IEEE Trans. Appl. Supercond. **24**, 1 (2014).

[11] A. P. Young, S. Knysh, and V. N. Smelyanskiy, Size Dependence of the Minimum Excitation Gap in the Quantum Adiabatic Algorithm, Phys. Rev. Lett. **101**, 170503 (2008).

[12] I. Hen and A. P. Young, Exponential complexity of the quantum adiabatic algorithm for certain satisfiability problems, Phys. Rev. E **84**, 061152 (2011).

[13] I. Hen, Excitation gap from optimized correlation functions in quantum Monte Carlo simulations, Phys. Rev. E **85**, 036705 (2012).

[14] E. Farhi, D. Gosset, I. Hen, A. W. Sandvik, P. Shor, A. P. Young, and F. Zamponi, Performance of the quantum adiabatic algorithm on random instances of two optimization problems on regular hypergraphs, Phys. Rev. A **86**, 052334 (2012).

[15] 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).

[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] T. Lanting, A. J. Przybysz, A. Yu. Smirnov, F. M. Spedalieri, M. H. Amin, A. J. Berkley, R. Harris, F. Altomare, S. Boixo, P. Bunyk, N. Dickson, C. Enderud, J. P. Hilton, E. Hoskinson, M. W. Johnson, E. Ladizinsky, N. Ladizinsky, R. Neufeld, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, S. Uchaikin, A. B. Wilson, and G. Rose, Entanglement in a Quantum Annealing Processor, Phys. Rev. X **4**, 021041 (2014).

[18] S. Boixo, V. N. Smelyanskiy, A. Shabani, S. V. Isakov, M. Dykman, V. S. Denchev, M. Amin, A. Smirnov, M. Mohseni, and H. Neven, Computational role of collective tunneling in a quantum annealer, arXiv:1411.4036.

[19] S. Boixo, T. F. Ronnow, 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).

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

[21] 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? arXiv:1512.02206.

[22] 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).

[23] V. Martin-Mayor and I. Hen, Unraveling quantum annealers using classical hardness, Sci. Rep. **5**, 15324 (2015).

[24] D. Venturelli, S. Mandrà, S. Knysh, B. O'Gorman, R. Biswas, and V. Smelyanskiy, Quantum Optimization of Fully Connected Spin Glasses, Phys. Rev. X **5**, 031040 (2015).

[25] F. Belletti, M. Cotallo, A. Cruz, L. A. Fernandez, A. Gordillo, A. Maiorano, F. Mantovani, E. Marinari, V. Martiń-Mayor, A. Muñoz-Sudupe, D. Navarro, S. Pérez-Gaviro, J. J. Ruiz-Lorenzo, S. F. Schifano, D. Sciretti, A. Tarancón, R. Tripiccione, and J. L. Velasco (Janus Collaboration), Simulating spin systems on IANUS, an FPGA-based computer, Comput. Phys. Commun. **178**, 208 (2008).

[26] F. Belletti, M. Guidetti, A. Maiorano, F. Mantovani, S. F. Schifano, R. Tripiccione, M. Cotallo, S. Perez-Gaviro, D. Sciretti, J. L. Velasco, A. Cruz, D. Navarro, A. Tarancon, L. A. Fernandez, V. Martin-Mayor, A. Muñoz-Sudupe, D. Yllanes, A. Gordillo-Guerrero, J. J. Ruiz-Lorenzo, E. Marinari, G. Parisi, M. Rossi, and G. Zanier (Janus Collaboration), Janus: An FPGA-based system for high-performance scientific computing, Comput. Sci. Eng. **11**, 48 (2009).

[27] M. Baity-Jesi, R. A. Baños, A. Cruz, L. A. Fernandez, J. M. Gil-Narvion, A. Gordillo-Guerrero, D. Iniguez, A. Maiorano, F. Mantovani, E. Marinari, V. Martin-Mayor, J. Monforte-Garcia, A. Muñoz Sudupe, D. Navarro, G. Parisi, S. Perez-Gaviro, M. Pivanti, F. Ricci-Tersenghi, J. J. Ruiz-Lorenzo, S. F. Schifano, B. Seoane, A. Tarancon, R. Tripiccione, and D. Yllanes (Janus Collaboration), Janus II: A new generation application-driven computer for spin-system simulations, Comput. Phys. Commun. **185**, 550 (2014).

[28] S. R. McKay, A. N. Berker, and S. Kirkpatrick, Spin-glass Behavior in Frustrated Ising Models with Chaotic Renormalization-group Trajectories, Phys. Rev. Lett. **48**, 767 (1982).

[29] A. J. Bray and M. A. Moore, Chaotic Nature of the Spin-glass Phase, Phys. Rev. Lett. **58**, 57 (1987).

[30] J. R. Banavar and A. J. Bray, Chaos in spin glasses: A renormalization-group study, Phys. Rev. B **35**, 8888 (1987).

[31] I. Kondor, On chaos in spin glasses, J. Phys. A **22**, L163 (1989).

[32] I. Kondor and Végsö, Sensitivity of spin-glass order to temperature changes, J. Phys. A **26**, L641 (1993).

[33] A. Billoire and E. Marinari, Evidence against temperature chaos in mean-field and realistic spin glasses, J. Phys. A **33**, L265 (2000).

[34] T. Rizzo, Against chaos in temperature in mean-field spin-glass models, J. Phys. A: Math. Gen. **34**, 5531 (2001).

[35] R. Mulet, A. Pagnani, and G. Parisi, Against temperature chaos in naive Thouless-Anderson-Palmer equations, Phys. Rev. B **63**, 184438 (2001).

[36] A. Billoire and E. Marinari, Overlap among states at different temperatures in the sk model, Europhys. Lett. **60**, 775 (2002).

[37] F. Krzakala and O. C. Martin, Chaotic temperature dependence in a model of spin glasses, Eur. Phys. J. **28**, 199 (2002).

[38] T. Rizzo and A. Crisanti, Chaos in Temperature in the Sherrington-Kirkpatrick Model, Phys. Rev. Lett. **90**, 137201 (2003).

[39] G. Parisi and T. Rizzo, Chaos in temperature in diluted mean-field spin-glass, J. Phys. A **43**, 235003 (2010).

[40] M. Sasaki, K. Hukushima, H. Yoshino, and H. Takayama, Temperature Chaos and Bond Chaos in Edwards-Anderson Ising Spin Glasses: Domain-wall Free-energy Measurements, Phys. Rev. Lett. **95**, 267203 (2005).

[41] H. G. Katzgraber and F. Krzakala, Temperature and Disorder Chaos in Three-dimensional Ising Spin Glasses, Phys. Rev. Lett. **98**, 017201 (2007).

[42] L. A. Fernandez, V. Martin-Mayor, G. Parisi, and B. Seoane, Temperature chaos in 3d Ising spin glasses is driven by rare events, Europhys. Lett. **103**, 67003 (2013).

[43] A. Billoire, Rare events analysis of temperature chaos in the Sherrington-Kirkpatrick model, J. Stat. Mech. (2014) P04016.

[44] K. H. Fischer and J. A. Hertz, *Spin Glasses* (University of Cambridge, Cambridge, UK, 1991).

[45] K. Binder and A. P. Young, Spin glasses: Experimental facts, theoretical concepts and open questions, Rev. Mod. Phys. **58**, 801 (1986).

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

[47] A. D. Sokal, Monte Carlo methods in statistical mechanics: Foundations and new algorithms, in *Functional Integration: Basics and Applications*, edited by C. DeWitt-Morette, P. Cartier, and A. Folacci (Plenum, New York, 1997).

[48] L. A. Fernandez, V. Martin-Mayor, S. Perez-Gaviro, A. Tarancon, and A. P. Young, Phase transition in the three dimensional Heisenberg spin glass: Finite-size scaling analysis, Phys. Rev. B **80**, 024422 (2009).

[49] R. Alvarez Baños, A. Cruz, L. A. Fernandez, J. M. Gil-Narvion, A. Gordillo-Guerrero, M. Guidetti, A. Maiorano, F. Mantovani, E. Marinari, V. Martin-Mayor, J. Monforte-Garcia, A. Muñoz Sudupe, D. Navarro, G. Parisi, S. Perez-Gaviro, J. J. Ruiz-Lorenzo, S. F. Schifano, B. Seoane, A. Tarancon, R. Tripiccione, and D. Yllanes (Janus Collaboration), Nature of the spin-glass phase at experimental length scales, J. Stat. Mech. (2010) P06026.

[50] F. Hamze and N. de Freitas, From fields to trees, in *UAI*, edited by D. M. Chickering and J. Y. Halpern (AUAI Press, Arlington, Virginia, 2004), p. 243.

[51] A. Selby, Efficient subgraph-based sampling of Ising-type models with frustration, arXiv:1409.3934.

[52] The "Boltzmann factor" $e^{-\beta|\Delta \mathrm{TTS}|}$ should not be confused with the standard one in statistical mechanics, $e^{-\beta H}$: in our case the TTS plays the role that the Hamiltonian would play in a standard problem.

[53] A. Selby, QUBO-Chimera, https://github.com/alex1770/QUBO-Chimera.

[54] For the HFS algorithm, the most natural choice of TTS, so as to quantify the computational difficulty, is simply the physical, average solve time of an instance; due to the nature of the algorithm, it does not easily lend itself to a "universal," system independent measure of TTS (e.g., number of algorithmic steps). The reader is referred to Refs. [16] and [51] for more details.

[55] If one were to use a solver which suffers from intrinsic control errors (e.g., an analog device, such as the DW annealers), i.e., encoding errors in the J_{ij}, one may have to perform some kind of averaging procedure to try to estimate the TTS more accurately (e.g., in the DW case, running over several different gauges [15]). The performance of our algorithm will of course be adversely affected in such a case.

[56] To generate the hardest instances, $k = 4$, we ran our algorithm 780 times with up to 1000 steps (that is, at least 100 of the 780 instances fell within the time interval [0.8,1.2] s). The total CPU time this took was approximately 400 h (though note that our code was not optimized). Also note that of these 780, about 120 have $t_{HFS} > 1.2$ s. The other k groups took significantly less time to generate.

[57] Screening instances on PT occurs in two phases. Phase 1 screens 76800 random instances, taking ~900 CPU h. Phase 2 then screens the 1024 (estimated) worst case instances for another 100 h. Given 76800 random 500+ bit instances we expect approximately 15 to have $\tau > 10^7$ (~65 h per instance).

[58] W. Barthel, A. K. Hartmann, M. Leone, F. Ricci-Tersenghi, M. Weigt, and R. Zecchina, Hiding Solutions in Random Satisfiability Problems: A Statistical Mechanics Approach, Phys. Rev. Lett. **88**, 188701 (2002).

[59] F. Krzakala and L. Zdeborová, Hiding Quiet Solutions in Random Constraint Satisfaction Problems, Phys. Rev. Lett. **102**, 238701 (2009).