Patch-planting spin-glass solution for benchmarking

Wenlong Wang,[1] Salvatore Mandrà,[2,3] and Helmut G. Katzgraber[1,4,5]
[1]*Department of Physics and Astronomy, Texas A&M University, College Station, Texas 77843-4242, USA*
[2]*NASA Ames Research Center Quantum Artificial Intelligence Laboratory (QuAIL), Mail Stop 269-1, Moffett Field, California 94035, USA*
[3]*Stinger Ghaffarian Technologies Inc., 7701 Greenbelt Road, Suite 400, Greenbelt, Maryland 20770, USA*
[4]*1QB Information Technologies (1QBit), Vancouver, British Columbia, Canada V6B 4W4*
[5]*Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, New Mexico 87501, USA*
(Received 8 June 2017; published 29 August 2017)

We introduce an algorithm to generate (not solve) spin-glass instances with planted solutions of arbitrary size and structure. First, a set of small problem patches with open boundaries is solved either exactly or with a heuristic, and then the individual patches are stitched together to create a large problem with a known planted solution. Because in these problems frustration is typically smaller than in random problems, we first assess the typical computational complexity of the individual patches using population annealing Monte Carlo, and introduce an approach that allows one to fine-tune the typical computational complexity of the patch-planted system. The scaling of the typical computational complexity of these planted instances with various numbers of patches and patch sizes is investigated and compared to random instances.

DOI: 10.1103/PhysRevE.96.023312

I. INTRODUCTION

Many optimization problems belong to the NP-hard complexity class, for which it is believed that no algorithms exist to solve them in polynomial time. Spin-glass problems without biases and on nonplanar topologies, such as the Edward-Anderson (EA) model [1], represent a subclass of the NP-hard class. Because spin glasses are the simplest models with both disorder and frustration that fall into the NP-hard class, they represent the ideal model systems to benchmark algorithms, as well as novel computing architectures. A number of heuristics, as well as exhaustive search methods, have been designed and developed to minimize spin-glass Hamiltonians as efficiently as possible. These method also include simulated annealing [2], parallel tempering Monte Carlo [3–6], population annealing Monte Carlo [7–10], and genetic algorithms [11,12], as well as branch-and-cut [13] algorithms, to name a few. Many of these optimization algorithms use only local updates during the minimization procedure. However, in many cases, the use of cluster algorithms with nonlocal updates can greatly enhance the searching process when the energy landscape has many metastable states with small overlap [14–16]. In the last two decades, quantum heuristics have been proposed as an alternative to classical heuristics, due to their potential to exploit quantum superposition and quantum tunneling effects. Among quantum approaches, adiabatic quantum optimization (AQO) is widely used [17–30] and likely the method most amenable to hardware implementations [31]. Current state-of-the-art AQO hardware is manufactured by D-Wave System Inc., whose latest chip allows for the quantum optimization of problems of approximately up to 2000 variables. However, whether AQO can be more efficient than classical algorithms for certain problems is still controversial [32–34].

Given the importance of comparing optimization techniques across disciplines, it is necessary to have benchmark problems that are (1) representative of the hardness of a typical NP-hard problem, (2) scalable for large systems, and for which (3) the ground state is known *a priori*. While it is easy to fulfill criteria (1) and (2), it is challenging to have large problems with known solutions.

There have been previous approaches to plant solutions for benchmarking purposes. For example, Ref. [35] used an approach based on constraint satisfaction problems. Although these problems are tunable in hardness, there is little control when selecting the coupler values between the individual variables. For analog machines with finite precision, such as the D-Wave quantum annealers, this could be an unnecessary restriction. Other approaches [36] start from a random coupler configuration and then stochastically update the values of the couplers with a penalty that directly correlates to the time-to-solution of a given solver. However, this approach has two shortcomings: Fist, it assumes that the typical computational hardness [37] of a problem for a given algorithm will carry over to other optimization techniques. Second, for extremely large problems, the stochastic approach will take sizable resources to thermalize and thus will not be practical.

The method we propose here and which we call "patch planting" (see Fig. 1), where we solve small problems (patches) with open boundaries and then stitch these together to plant an arbitrarily large solution to an instance, does not suffer from these shortcomings: First, arbitrarily large problems can be generated. Second, by assessing the typical complexity using the entropic family size of population annealing Monte Carlo, a metric that characterizes the landscape of the problem and not the algorithmic complexity, we do not depend on the behavior of a particular algorithm when assessing the typical time to solution for a particular instance. Finally, the method poses no restrictions to coupler values, biases (field terms), or lattice topologies.

The paper is structured as follows. In Sec. II we introduce the benchmark problem, as well the patch-planting algorithm. In Sec. III we use simulated annealing, population annealing Monte Carlo, as well as experiments on the D-Wave 2X quantum annealer to illustrate how patch planting can produce computational problems that are typically hard. Concluding remarks are presented in Sec. IV.

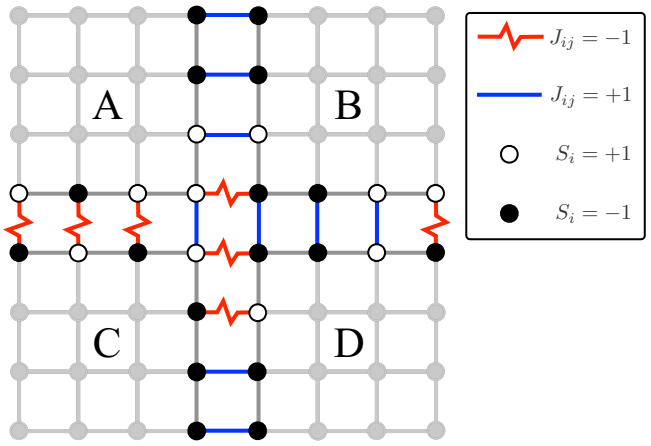

FIG. 1. Schematic diagram of the patch-planting method for a two-dimensional lattice. First, the ground state for each (easily solvable) patch (A–D) is computed using free boundary conditions. Then a ground state configuration of each patch is selected. Edge spins between the patches are represented by empty and full circles. Couplers between the edge spins between adjacent patches are added under the condition that all interactions are satisfied, i.e., if two spins have the same value (i.e., both full or both empty circles), a ferromagnetic coupler (straight blue line) is added. If, however, two adjacent spins are different, then an antiferromagnetic coupler (red wiggly line) is added. The direct product of the ground state of the patches A–D is then the ground state of the large planted system.

II. PATCH PLANTING

The patch-planting heuristic can be described via the following steps:

(1) Find the ground state of patches using *free boundary conditions*.

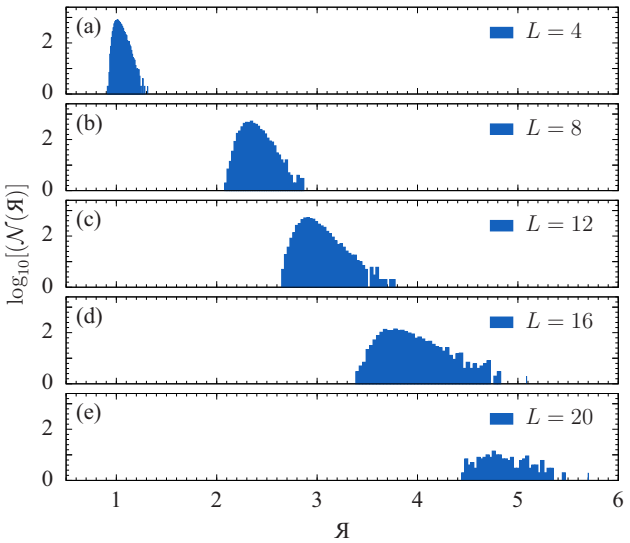

FIG. 2. Distribution of $\mathcal{R}$ [defined in Eq. (3)] for various system sizes [$L = 4$ (a), 8 (b), 12 (c), 16 (d), and 20 (e)] in three space dimensions. For increasing system size, the typical complexity of the problems grows, which is mirrored by the distributions of $\mathcal{R}$ shifting to the right.

(2) For each patch, choose an arbitrary ground-state configuration.

(3) Connect the patches with couplers between the free boundary spins ensuring that all couplings are satisfied.

Note that the patches can be chosen arbitrarily, as long as they can be glued together to form the desired problem or topology with the edges to be stitched together having free boundaries. In addition, the individual patches can be solved with any available optimization technique. As demonstrated below, it is important to solve as large patches as possible, because this will result in problems of comparable computational complexity to purely random problems. In some cases, the breakup of a problem might result in a patch that can be solved exactly, i.e., in polynomial time. Finally, when stitching the patches together, as shown in Fig. 1, it is important to "satisfy" the interaction between two spins of different patches. This means that the coupler has to be chosen as to minimize the dimer's energy. Knowing the minimizing configuration of the individual patches and assigning the stitching couplers as to satisfy the interactions between spins of neighboring patches then results in a larger planted solution [38].

As described in Sec. III in more detail, the typical computational complexity of the patched problem can be tuned by either changing the patch size (the larger, the harder) or using hard patches (the harder the patch, the harder the compound problem), e.g., by measuring the entropic family size via population annealing Monte Carlo. This metric can be measured with little numerical effort and gives a good representation of the typical computational complexity of a problem. Therefore, by postselecting individual patches, problems of different typical computational complexity can be generated.

Note that in the description of the patch-planting procedure no details of the problem to be studied have been mentioned, because the approach is agnostic to the choice of couplers and topologies. We thus emphasize that the patch-planting approach can be used for problems of *arbitrary* topology and for an arbitrary set of coupler values and biases. As such, solutions for arbitrary problems can be planted. This is of much importance when attempting to generate problem sets with particular features, such as synthetic application problems, that are known to have a specific nonrandom structure, or problems where the minimum energy gap is fixed (and large) to mitigate the effects of noise on analog optimization machines [39,40].

III. EXPERIMENTS

A. Benchmark problem

To test the properties of patch planting, we use the Edward-Anderson (EA) Ising spin-glass model [1], initially in three space dimensions. Later, we perform experiments on the D-Wave 2X quantum annealer using the native topology of the machine [41]. The EA spin glass is defined by the following Hamiltonian:

$$H = -\sum_{ij} J_{ij} S_i S_j - \sum_i h_i S_i \,, \tag{1}$$

where $S_i \in \{\pm 1\}$ are Ising spins and the first sum is over spin-spin interactions. For a three-dimensional lattice, the sum

TABLE I. Simulation parameters for the three-dimensional EA model experiments using population annealing Monte Carlo. Here L_0 is the patch size, L is the linear system size, R is the number of replicas used in the simulation, $T_0 = 1/\beta_0$ is the lowest temperature simulated, N_T is the number of temperature steps (evenly spaced in β) in the annealing schedule, BC is the type of boundary condition [either periodic boundary condition (PBC) or free boundary condition (FBC)], and $N_{\rm sa}$ is the number of disorder realizations studied. For each replica, $N_S = 10$ Monte Carlo sweeps are performed at each temperature during the anneal. Data for PBC with $L = 8$ are taken from Ref. [9].

L_0	L	R	T_0	N_T	BC	$N_{\rm sa}$
4	4	4×10^3	0.2	101	FBC	345 600
4	8	10^4	0.2	101	FBC	5000
4	12	5×10^4	0.2	201	FBC	5120
4	16	2×10^5	0.2	301	FBC	1877
4	20	10^6	0.2	401	FBC	194
5	5	10^4	0.2	101	FBC	345 600
5	10	10^5	0.2	201	FBC	5000
6	6	2×10^4	0.2	101	FBC	41 472
6	12	10^5	0.2	201	FBC	1752
8	8	5×10^4	0.2	201	FBC	23 358
8	16	10^6	0.2	301	FBC	624
10	10	10^6	0.2	301	FBC	8000
10	20	2×10^6	0.2	401	FBC	260
8	8	10^5	0.2	101	PBC	5099
12	12	10^6	0.2	201	PBC	3812

is over nearest neighbors on a cubic lattice. For simplicity, all the local magnetic fields are set to zero, i.e., $h_i = 0$. We do emphasize, however, that patch planting also works with external biases. The spin-spin interactions J_{ij} are chosen from a normal distribution with zero mean and unit variance. A set of the couplings defines an "instance."

Given the hardware limitations of the D-Wave quantum chips, instances for the D-Wave 2X have been created by planting and patching together K_{44} unit cells following the two-dimensional logical structure of the Chimera graph. The couplers are randomly drawn from the Sidon set $\{\pm5, \pm6, \pm7\}$ [40]. In both cases, we use free boundary conditions (FBCs) for the patches to plant larger instances. We also compare our patched instances with free boundary conditions to random instances with periodic boundary conditions (PBCs).

B. Simulation details

We use the entropic family size of population annealing Monte Carlo ρ_s [10] to characterize the hardness of the instances. All simulation parameters for the three-dimensional Edwards-Anderson model are listed in Table I. For the Chimera graph studies on the D-Wave 2X machine, we find the ground state of the patches using $R = 2 \times 10^5$ population members, $N_T = 301$ temperature steps, $N_S = 10$ Monte Carlo sweeps, and $T_0 = 0.1$ the lowest temperature of the anneal. The simulation for random problems are done with the same parameters, except $R = 10^6$.

Experiments on the D-Wave 2X quantum annealer have been performed using a chip with $N = 1097$ working qubits. For the Chimera graph, we used all available qubits and patched the system using either two, three, or four patches, respectively. That is, if the system has 12×12 K_{44} cells of eight qubits each, we divide the whole lattice into two patches of 6×12 K_{44} cells, three patches of 4×12 K_{44} cells, or four patches of 3×12 K_{44} cells. For the experiments, we used an annealing time of 20 μs, 100 gauges, and 1000 readouts for each gauge.

C. Correlation between typical hardness and the entropic family size

The first crucial step in investigating the hardness of instances is to find a good metric that reliably characterizes the typical problem complexity, yet is easy to measure with little computational cost. One approach would be to use the success probability of simulated annealing as a proxy. However, even for medium-size systems, this metric becomes unreliable and computationally expensive. Another possibility consists in using specialized classical algorithms [36], such as the Hamze–de Freitas–Selby heuristic [42,43]. However, in this case the typical computational complexity depends on a chosen algorithm and not on the intrinsic properties of the problem's energy landscape. The latter can be mapped out well for random problems using parallel tempering Monte Carlo [39], however, at sizable computational cost for large patches. Therefore, in this work we infer the typical hardness of instances through the entropic family size ρ_s of population annealing Monte Carlo.

Population annealing (PA) Monte Carlo [7,44] is closely related to simulated annealing (SA), except that it uses a population of R replicas and the population is resampled at each temperature anneal step to maintain thermal equilibrium. At each simulation step, replicas are duplicated accordingly to the ratio between the Boltzmann factors computed after and before updating the temperature. This means that replicas with lower (higher) energy tend to be duplicated (eliminated), ensuring the correct representation of the Boltzmann distribution. Therefore, PA improves the probability to find the lowest energy state over SA by more efficiently sampling phase space. We choose to normalize our replicas so that the population size stays approximately the same. Similar to SA, Metropolis sweeps are applied to each replica at the new temperature. At low temperatures, most of the original population is eliminated in the resampling steps and the final population is a descendant from a small subset of the initial population. Let n_i be the fraction of the population from family i in the initial population, then

$$\rho_s = \lim_{R\to\infty} R \times e^{\sum_i n_i \log n_i}. \tag{2}$$

Here ρ_s represents the characteristic survival family size. The larger ρ_s is, the less surviving families, i.e., the more rugged the energy landscape. Moreover, ρ_s correlates strongly with the integrated autocorrelation time of parallel tempering, which is also a proxy towards the roughness of the energy landscape [10]. Note that ρ_s converges quickly in population size and is easily estimated with simulations. See Ref. [10] for more details on population annealing. Because ρ_s is approximately log-normal distributed (see Fig. 2), let us define the logarithm of ρ_s as

$$Я = \log_{10}(\rho_s). \tag{3}$$

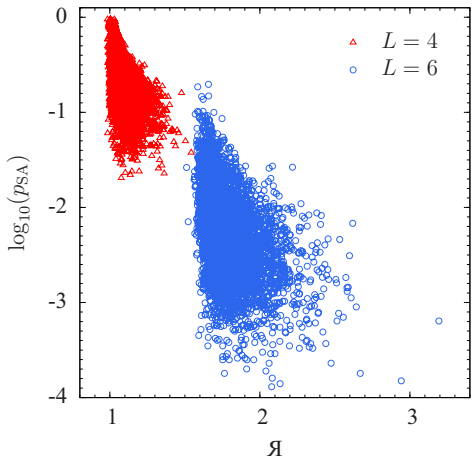

FIG. 3. Correlation of the probability to find the ground state of simulated annealing $p_{\rm SA}$ and the log of entropic family size of population annealing Я for three-dimensional systems with $L = 4$ and $L = 6$ at $\beta = 5$ (data taken from Refs. [45] and [9]). When Я is larger, it is also more difficult to find the ground state, i.e., $p_{\rm SA}$ is smaller. Note that $p_{\rm SA}$ drops very rapidly as L increases, while it is easier to measure Я.

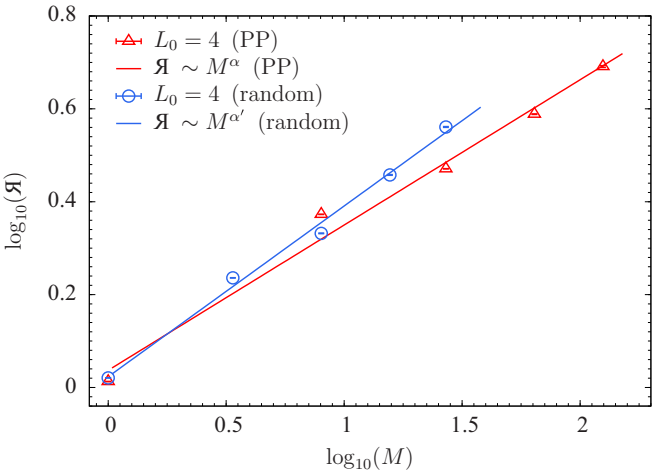

FIG. 4. Scaling of the logarithm of the entropic family size Я = $\log_{10}\rho_s$ by varying the number of patches M (triangles, labeled with "PP"). The line is a power-law fit of the form $\text{Я}(L_0)M^{\alpha}$, where $\text{Я}(L_0)$ is Я of a single patch. From the fit, we obtain $\alpha = 0.31(3)$. We also compare to random problems on a three-dimensional lattice (circles). In this case a power-law fit results in $\alpha' = 0.37(4)$, i.e., the two classes scale similarly, yet with two different exponents.

Figure 3 shows the correlation between the probability to find the ground state for SA, $p_{\rm SA}$ at inverse temperature $\beta = 1/T = 5$ and Я (data taken from Refs. [9,45]). As expected, the probability of success decreases by increasing Я. Indeed, SA struggles more to find the ground state when the energy landscape is more rugged. Therefore, Я represents a good metric to estimate the typical hardness of optimization problems. In this work, because we study large patched system sizes in three dimensions, we have used Я at $\beta = 3$, which is still a low temperature compared to the spin-glass transition temperature for this model [46]. For the Chimera graph, where there is no phase transition, we have used Я at a considerably lower temperature $\beta = 10$.

D. Results in three space dimensions

We first focus on the scaling properties of Я for patch-planted instances by either varying the patch sizes L_0 or the system size L. In addition, we also demonstrate that harder patches can be used to patch harder instances.

Let $M = (L/L_0)^3$ the number of patches of size L_0. For random instances, ρ_s grows exponentially with L [10]. Because one would expect that ρ_s for a problem of size L by patching M patches of size L_0 cannot be larger than the product of ρ_s of the individual patches, the patched instance complexity is bounded,

$$\text{Я}(M, L_0) \leqslant M\text{Я}(L_0), \tag{4}$$

where $\text{Я}(M, L_0)$ is Я of the patched instance of M patches of size L_0 and $\text{Я}(1, L_0) \equiv \text{Я}(L_0)$ is Я of a patch. In Fig. 4 we show the scaling of Я by varying the number of patches M and a power-law fit of the form

$$\text{Я}(M, L_0) = \text{Я}(L_0)M^{\alpha}, \tag{5}$$

where $0 < \alpha < 1$. Я scales sublinearly with M with an exponent $\alpha = 0.31(3)$. This proves that the patch-planted instances become *harder* by increasing the system size via the number of patches. Therefore, it is guaranteed that, for a sufficiently large number of patches, patch-planted instances can become arbitrarily hard in the thermodynamic limit. Figure 5 shows the scaling of the exponent α by increasing the size of the patches L_0, while keeping the number of patches fixed to $M = 8$. As one can see from the figure, α remains roughly constant for a wide range of L_0 values, implying that α is a characteristic constant for patch-planted problems. It is interesting to compare the scaling with random instances by defining an effective number of blocks as $M = (L/L_0)^3$, also shown in Fig. 4. We find that both random and patch-planted instances have a similar scaling form, although the random class has a larger exponent $\alpha' = 0.37(4)$, as expected. Therefore, ρ_s for patched instances

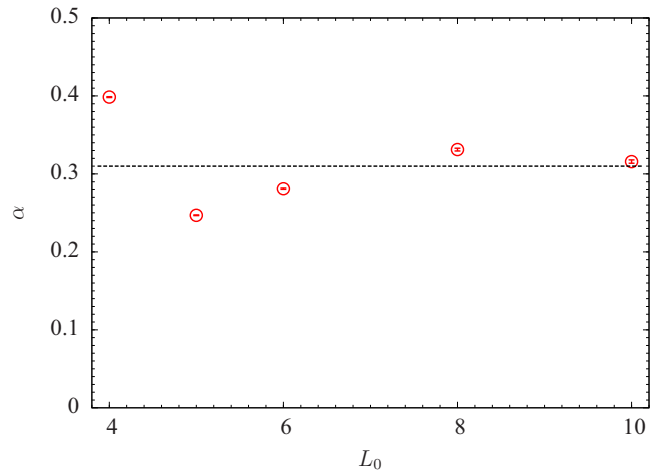

FIG. 5. Scaling exponent α ($\text{Я} \sim M^{\alpha}$; see Fig. 4), by varying the patch size L_0 but keeping the number of patches fixed to $M = 8$.

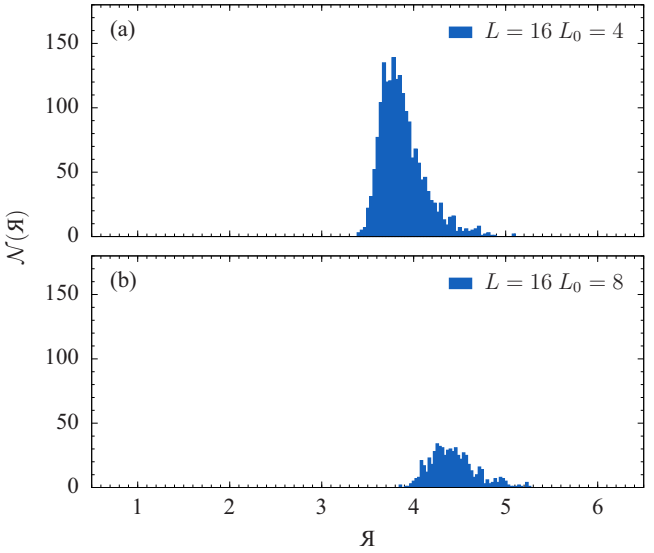

FIG. 6. Comparison of the distribution of Я for $L = 16$ in three space dimensions, but with different patch sizes $L_0 = 4$ (a) and 8 (b). There is a noticeable shift in the distributions of Я. Therefore, to patch harder instances, one should use as large patch sizes as possible.

also approximately scales exponentially with system size L, as is the case for random instances. Note that α and α' likely depend on the characteristics of the problem to be studied.

One may also expect to have some benefit by using either larger or harder patches. Indeed, in both cases, this results in having a larger value of Я. In Fig. 6 we show the effects of having larger patches by analyzing the distribution of Я at fixed size of the system, $L = 16$, using two different patch sizes, $L_0 = 4$ [Fig. 6(a)] and $L_0 = 8$ [Fig. 6(b)]. As one can see, patched instances are consistently harder by using larger patches for a fixed system size. Similarly, in Fig. 7 we show the distribution of Я by patching instances with $M = 8$ patches of

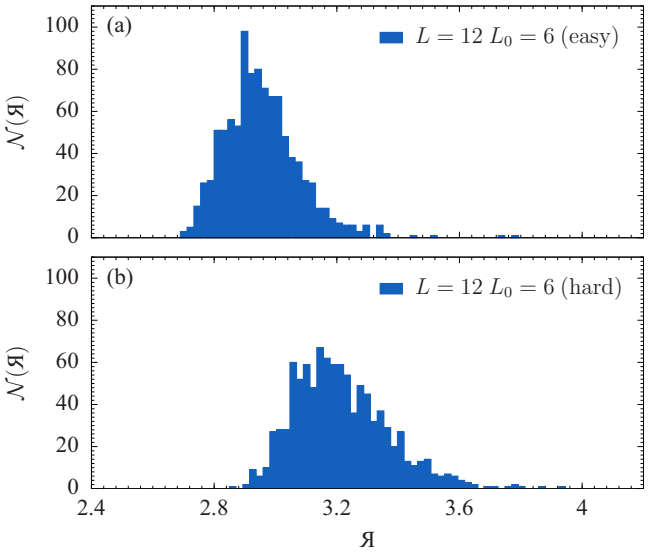

FIG. 7. Distributions of Я for $M = 8$ ($L = 12$) patches of size $L_0 = 6$ using either easy patches (a) or hard patches (b). As expected, the distribution shifts to the right when harder patches are used.

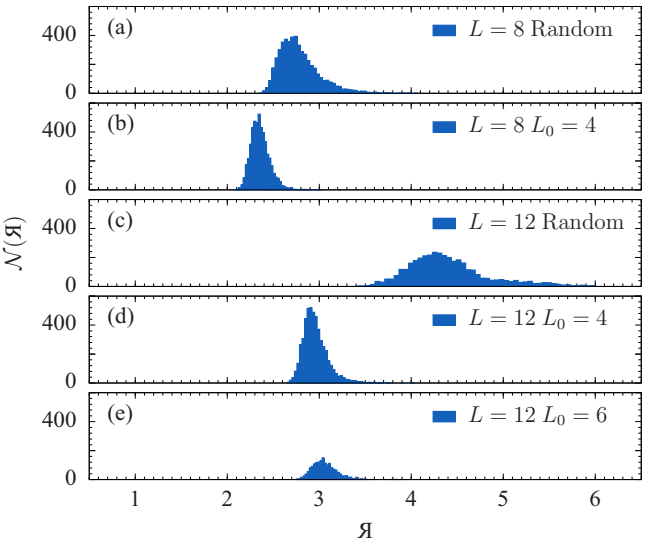

FIG. 8. Comparison of the typical complexity of the patched instances with random instances. The patched instances shown in panels (b), (d), and (e) are generally computationally easier than their random counterparts shown in panels (a) and (c), for a given system size. Note the overlap between the distributions, i.e., by mining the data one can obtain very hard patched problems.

size $L_0 = 6$ by either using *easy* [Fig. 7(a)] or *hard* [Fig. 7(b)] patches. We defined *easy* patches as the 8000 patches with the smallest Я and hard patches as the 8000 patches with the

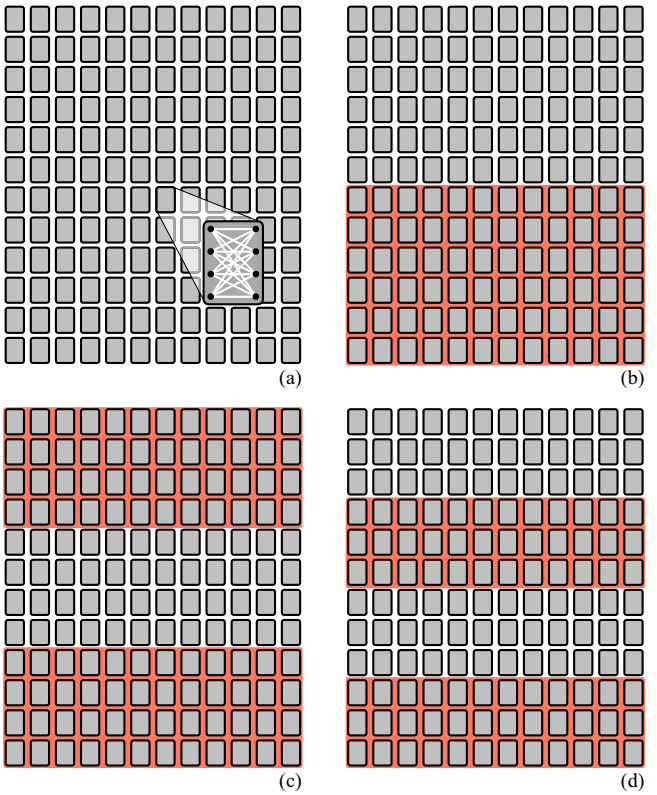

FIG. 9. Sketch of the different patch geometries used on the D-Wave 2X quantum annealer chip. Each gray block represents a K_{44} call with 8 sites. (a) A zoom of such cell. (a–d) The shading represents the different patches used from $M = 1$ (a) to $M = 4$ (d).

largest Я from the 41 472 patches randomly generated. From these, 1000 easy and 1000 hard instances are then generated.

Patch-planted instances generated using hard patches are consistently harder than patch-planted instances assembled from easy patches with the mean value of Я for both cases being 3.214(5) and 2.957(4), respectively. We note that the approach pioneered in Ref. [36] applied to the production of patches could be combined with patch planting to generate unusually hard planted problems. It is also interesting to compare the complexity of the patched instances with random instances. We show the distribution of Я for $L = 8$ [Figs. 8(a) and 8(b)] and $L = 12$ [Figs. 8(c)–8(e)] with different patch sizes and random instances. One can see that while the patched instances are generally easier than the random instances, they are not necessarily trivial. There is clear overlap between the distributions, i.e., by mining the data one can obtain problems of comparable typical complexity. Note also that the typical complexity grows with increasing patch size for a fixed system size.

Finally, we comment on the performance of parallel tempering (PT) on patched instances. Because population annealing and parallel tempering have a similar performance in both thermal sampling and optimization, and given that the entropic family size correlates strongly with the integrated autocorrelation time (characteristic measure of hardness of parallel tempering) [10], it is natural to expect the proposed patch-planted instances to be hard also for PT. To this end, it is noteworthy to mention recent results that analyze the performance of PT with isoenergetic cluster moves (ICMs) (see Ref. [16]) in solving patch-planted instances [47]. PT combined with ICMs has been found to be one of the best classical heuristics in solving hard optimization problems [33]. However, Ref. [47] clearly show that PT is not able to efficiently solve patch-planted instances (see Figs. 8–10).

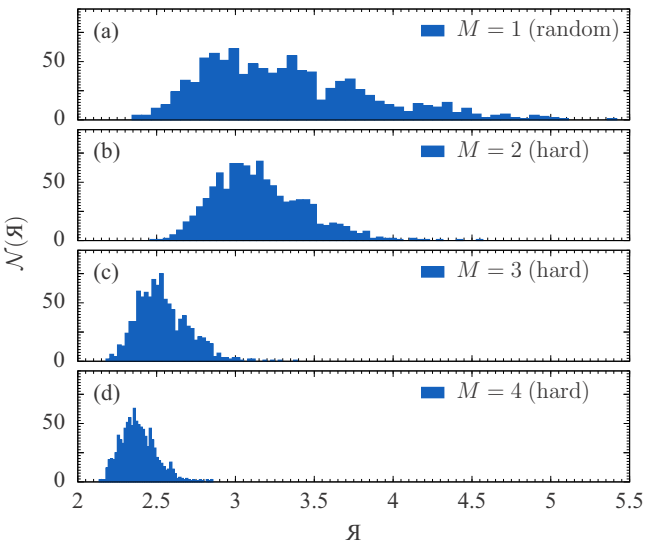

FIG. 10. Distributions of Я on a chimera graph with $N = 1097$ for random instances and patched instances with different number of patches M. There are 1000 instances each, and the patched instances were chosen from the hardest ones out of 10^4 instances in each class. Note that the patched problems with $M = 2$ (b) and random (a) are comparable. Panels (c) and (d) show that problems become easier for smaller patches, i.e., a larger number of patches.

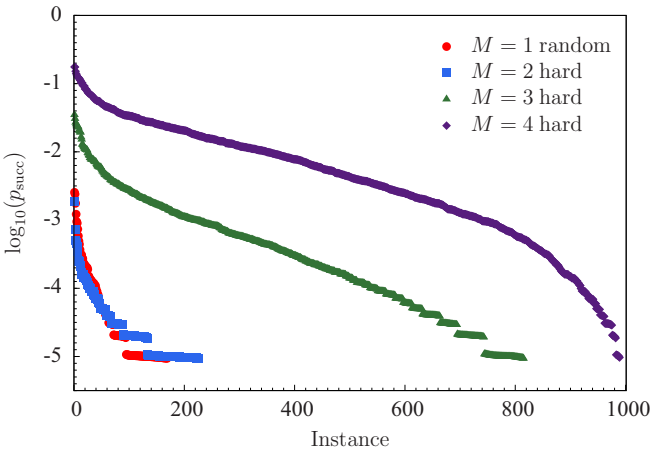

FIG. 11. Sorted probabilities to find the ground state $p_{\rm succ}$ from experiments with the D-Wave 2X quantum annealer for $N = 1097$ sites and different number of patches M. As in Fig. 10, random instances and instances with $M = 2$ are comparable.

E. Experiments on the D-Wave quantum annealer

We complement the numerical studies on three-dimensional spin glasses by experiments on the D-Wave 2X quantum annealer. For this purpose, we patch plant problems on the native topology of the machine and measure the probabilities to find the ground state $p_{\rm succ}$ over multiple runs. In addition, we compare to random problems and show correlation plots between the success probabilities and Я.

The topology of the machine with $N = 1097$ working qubits is cut into two, three, and four patches; see Fig. 9 for a graphical representation. For each experiment, we study 10^3 instances. For the patch-planted instances, we first generate 10^4 patch-planted problems from random patches and then use Я to select the 10^3 hardest ones. The distributions of Я for the problems studied is shown in Fig. 10. One can see that for an increasing number of patches M the problems become computationally easier. However, again by mining the data as done above results in hard problems. Figure 11 shows the

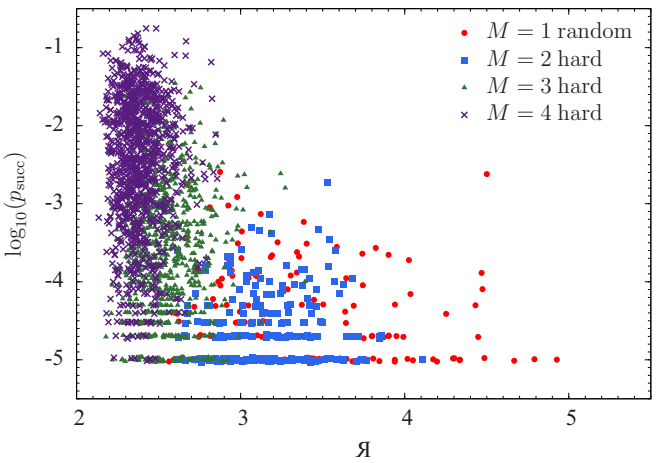

FIG. 12. Correlation of Я and $p_{\rm succ}$ for experiments on the D-Wave machine with $N = 1097$. Data for random instances, as well as instances with different numbers of patches M.

TABLE II. Statistics of the D-Wave 2X quantum annealer success probability p_{succ} in Fig. 11 for random instances and patched instances with different number of patches M. There are 10^3 instances each, and the patched instances were chosen from the hardest ones out of 10^4 instances in each class. p_{min}, p_{max}, and p_{ave} are the minimum, maximum, and average values of p_{succ}, respectively, and f is the fraction of instances with $p_{\text{succ}} = 0$.

	Random	$M = 2$	$M = 3$	$M = 4$
p_{min}	0	0	0	0
p_{max}	0.00240(32)	0.00137(59)	0.0326(32)	0.173(9)
p_{ave}	0.0000204(46)	0.0000129(24)	0.00115(10)	0.0127(7)
f	0.831(12)	0.775(13)	0.185(12)	0.011(3)

sorted success probabilities for the 1000 problems studied for different number of patches. One can see that problems with $M = 4$ are computationally much easier. It remains to be tested if changing the shape of the patch could make the problems harder. For example, the four patches could be chosen to be comprised of 6×6 K_{44} cells. Finally, Fig. 12 shows a correlation plot between success probabilities and Я. As can be seen, there is a good correlation between these two quantities, especially for larger patches. Experiments (not shown) suggest that the correlation becomes more pronounced for larger system sizes. With some data mining and only a 10-fold overhead, instances with two patches $M = 2$ have approximately the same complexity as the random ones, which are harder than instances with three patches and four patches. Statistics of the success probabilities are shown in Table II.

IV. SUMMARY

We have introduced the concept of patch planting to create planted solutions to Ising-type optimization problems for arbitrarily large systems. The method does not restrict the values of the couplers and works for any topology that can be decomposed into patches. We studied in detail the scaling of the typical complexity of the patched instances and compared this to random instances using population annealing Monte Carlo and the D-Wave 2X machine. From our results it is clear that one should use as large patches as possible to more faithfully reproduce the hardness of random problems. Patch planting is easy to implement and could be used to generate benchmark instances for future generations of quantum devices, as well as classical algorithms and any other novel hardware. The approach is generic in that solutions could also be planted for other paradigmatic optimization problems (e.g., the traveling salesman problem) with only minor modifications.

ACKNOWLEDGMENTS

We thank F. Hamze, J. Machta, and E. Brown for helpful discussions. H.G.K. would like to thank I. P. Freely and I. M. A. Wiener for inspiration at the early stages of the project. W.W. and H.G.K. acknowledge support from the National Science Foundation (Grant No. DMR-1151387). The research of H.G.K. is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via MIT Lincoln Laboratory Air Force Contract No. FA8721-05-C-0002. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purpose notwithstanding any copyright annotation thereon. We thank Texas A&M University for access to their Ada and Curie clusters.

[1] S. F. Edwards and P. W. Anderson, Theory of spin glasses, J. Phys. F: Met. Phys. **5**, 965 (1975).

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

[3] R. H. Swendsen and J.-S. Wang, Replica Monte Carlo Simulation of Spin-Glasses, Phys. Rev. Lett. **57**, 2607 (1986).

[4] C. Geyer, in *23rd Symposium on the Interface*, edited by E. M. Keramidas (Interface Foundation, Fairfax Station, VA, 1991), p. 156.

[5] K. Hukushima and K. Nemoto, Exchange Monte Carlo method and application to spin glass simulations, J. Phys. Soc. Jpn. **65**, 1604 (1996).

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

[7] K. Hukushima and Y. Iba, in *The Monte Carlo Method in the Physical Sciences: Celebrating the 50th Anniversary of the Metropolis Algorithm*, edited by J. E. Gubernatis (AIP, New York, 2003), Vol. 690, p. 200.

[8] E. Zhou and X. Chen, in *Proceedings of the 2010 Winter Simulation Conference (WSC)* (Springer, Baltimore, 2010), p. 1211.

[9] W. Wang, J. Machta, and H. G. Katzgraber, Comparing Monte Carlo methods for finding ground states of Ising spin glasses: Population annealing, simulated annealing, and parallel tempering, Phys. Rev. E **92**, 013303 (2015).

[10] W. Wang, J. Machta, and H. G. Katzgraber, Population annealing: Theory and application in spin glasses, Phys. Rev. E **92**, 063307 (2015).

[11] K. F. Pal, The ground state energy of the Edwards-Anderson Ising spin glass with a hybrid genetic algorithm, Physica A **223**, 283 (1996).

[12] A. K. Hartmann and H. Rieger, *Optimization Algorithms in Physics* (Wiley-VCH, Berlin, 2001).

[13] C. De Simone, M. Diehl, M. Jünger, P. Mutzel, G. Reinelt, and G. Rinaldi, Exact ground states in spin glasses: New experimental results with a branch-and-cut algorithm, J. Stat. Phys. **80**, 487 (1995).

[14] J. Houdayer, A cluster Monte Carlo algorithm for 2-dimensional spin glasses, Eur. Phys. J. B. **22**, 479 (2001).

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

[16] Z. Zhu, C. Fang, and H. G. Katzgraber, borealis—A generalized global update algorithm for Boolean optimization problems, arXiv:1605.09399.

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

[18] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing: An Introduction* (Oxford University Press, New York, 2001).

[19] 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).

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

[21] J. Roland and N. J. Cerf, Quantum search by local adiabatic evolution, Phys. Rev. A **65**, 042308 (2002).

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

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

[24] 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).

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

[26] K. L. Pudenz, T. Albash, and D. A. Lidar, Quantum annealing correction for random Ising problems, Phys. Rev. A **91**, 042302 (2015).

[27] A. Perdomo-Ortiz, B. O'Gorman, J. Fluegemann, R. Biswas, and V. N. Smelyanskiy, Determination and correction of persistent biases in quantum annealers, arXiv:1503.05679 [quant-phys].

[28] W. Vinci, T. Albash, G. Paz-Silva, I. Hen, and D. A. Lidar, Quantum annealing correction with minor embedding, Phys. Rev. A **92**, 042310 (2015).

[29] S. Mandrà, G. G. Guerreschi, and A. Aspuru-Guzik, Adiabatic quantum optimization in the presence of discrete noise: Reducing the problem dimensionality, Phys. Rev. A **92**, 062320 (2015).

[30] 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).

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

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

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

[34] S. Mandrà, Z. Zhu, and H. G. Katzgraber, Exponentially Biased Ground-State Sampling of Quantum Annealing Machines with Transverse-Field Driving Hamiltonians, Phys. Rev. Lett. **118**, 070502 (2017).

[35] 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).

[36] J. Marshall, V. Martin-Mayor, and I. Hen, Practical engineering of hard spin-glass instances, Phys. Rev. A **94**, 012320 (2016).

[37] In practical applications of optimization techniques, the *worst-case complexity* of a problem is not representative for the behavior of a finite subset of instances. For example, a problem could fall into the NP-hard optimization class with only having one extremely hard instance and all others being solvable in polynomial time. While this extreme case is likely unrealistic, it does highlight the importance of the *typical computational complexity*, i.e., the median time needed to solve a finite set of problems for a given instance class.

[38] The idea of patch planting is inspired by patchwork approaches to compute ground states of Ising spin glasses [48]. However, in this case the goal is to plant a solution and not find one.

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

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

[41] P. Bunyk, E. Hoskinson, M. W. Johnson, E. Tolkacheva, F. Altomare, A. J. Berkley, R. Harris, J. P. Hilton, T. Lanting, and J. Whittaker, Architectural considerations in the design of a superconducting quantum annealing processor, IEEE Trans. Appl. Supercond. **24**, 1 (2014).

[42] F. Hamze and N. de Freitas, in Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (AUAI Press, Arlington, VA, 2004), p. 243.

[43] A. Selby, Efficient subgraph-based sampling of Ising-type models with frustration, arXiv:1409.3934 [cond-mat.stat-mech].

[44] J. Machta, Population annealing with weighted averages: A Monte Carlo method for rough free-energy landscapes, Phys. Rev. E **82**, 026704 (2010).

[45] W. Wang, J. Machta, and H. G. Katzgraber, Evidence against a mean-field description of short-range spin glasses revealed through thermal boundary conditions, Phys. Rev. B **90**, 184412 (2014).

[46] H. G. Katzgraber, M. Körner, and A. P. Young, Universality in three-dimensional Ising spin glasses: A Monte Carlo study, Phys. Rev. B **73**, 224432 (2006).

[47] H. Karimi, G. Rosenberg, and H. G. Katzgraber, Effective optimization using sample persistence: A case study on quantum annealers and various Monte Carlo optimization methods, arXiv:1706.07826.

[48] C. K. Thomas, O. L. White, and A. A. Middleton, Persistence and memory in patchwork dynamics for glassy models, Phys. Rev. B **77**, 092415 (2008).