

Quadratic Unconstrained Binary Optimization via Quantum-Inspired Annealing


Joseph Bowles,^{1,*} Alexandre Dauphin,¹ Patrick Huembeli,² José Martínez,³ and Antonio Acín^{1,4}

¹*ICFO—Institut de Ciències Fotoniques, The Barcelona Institute of Science and Technology, Castelldefels, Barcelona 08860, Spain*

²*Institute of Physics, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne CH-1015, Switzerland*

³*Quside Technologies SL, Carrer d'Esteve Terradas, 1, Castelldefels, Barcelona 08860, Spain*

⁴*ICREA—Institut Català de Recerca i Estudis Avançats, Lluís Companys 23, Barcelona 08010, Spain*

 (Received 25 October 2021; revised 4 March 2022; accepted 2 August 2022; published 7 September 2022)

We present a classical algorithm to find approximate solutions to instances of quadratic unconstrained binary optimization. The algorithm can be seen as an analog of quantum annealing under the restriction of a product-state space, where the dynamical evolution in quantum annealing is replaced with a gradient-descent-based method. This formulation is able to quickly find high-quality solutions to large-scale problem instances and can naturally be accelerated by dedicated hardware such as graphics processing units. We benchmark our approach for large-scale problem instances with tunable hardness and planted solutions. We find that our algorithm offers a similar performance to current state-of-the-art approaches within a comparably simple gradient-based and nonstochastic setting.

DOI: [10.1103/PhysRevApplied.18.034016](https://doi.org/10.1103/PhysRevApplied.18.034016)

I. INTRODUCTION

Combinatorial optimization is a class of optimization problems that has applications in nearly all areas of industry and society [1]. Such problems involve searching for an optimal object amongst an often enormous but finite range of potential candidates and are notoriously difficult to solve. One type of combinatorial optimization, called *quadratic unconstrained binary optimization* (QUBO) [2,3], involves searching for a bit string that minimizes a quadratic function of its elements. QUBO problems have recently attracted considerable attention, largely because they can be naturally tackled by quantum computers by first mapping the problem to the energy minimization of a classical Ising model and then promoting this system to a quantum Ising model [4–11]. By exploiting phenomena such as superposition and entanglement, the hope is that these quantum algorithms provide faster or higher-quality solutions than their classical counterparts. Much of the focus of showing a quantum advantage versus classical optimization has consequently been focused around the class of QUBO problems; for example, the D-Wave quantum computer [12,13] exclusively solves this type of optimization problem.

At the same time, the interest in QUBO problems has inspired new classical algorithms [14–20] and corresponding optimization devices [21–26]. Since these algorithms can be run on digital logic, they can often handle

orders of magnitude more variables than current quantum computers and, as such, will serve as valuable classical benchmarks as quantum computers increase in size. Furthermore, since it is known that many hard optimization problems can be mapped to the QUBO setting [27–31], these classical solvers may also lead to improvements in classical-optimization heuristics in general. An important question is thus: what types of classical algorithm are best suited to solving large-scale QUBO problems? A standard approach is to use algorithms such as simulated annealing or population annealing [32], since these algorithms are naturally suited to the discrete nature of QUBO problems. Although they perform well in general, the fast implementation of these algorithms for large-scale problems is limited by hardware (for example, Fujitsu's digital annealer [33] is currently limited to 8192 variables due to the limitations of the CPU cache) and it is not clear how best to parallelize these algorithms since in their standard versions they rely on sequential parameter updates. A more recent method involves classically simulating the dynamical evolution of a physical system. This is generally done using continuous degrees of freedom (such as position and momentum) the energy of which is related to the corresponding Ising Hamiltonian. Two recently introduced methods, Toshiba's simulated bifurcation (SB) [14,15] and the simulated coherent Ising machine (SIM CIM) [19], are of this form. The original SB algorithm [14] is based on a simulation of classical nonlinear Hamiltonian system of oscillators and has later been developed into two modified versions of the algorithm [15]. The SIM CIM

*bowles.physics@gmail.com

[19] is a classical simulation of the optical neural network called the coherent Ising machine [10]. Importantly, both of these methods are suited to solving large-scale QUBO problems of tens or even hundreds of thousands of variables, owing to the possibility of graphics-processing-unit (GPU) acceleration of the computationally demanding part of these algorithms. Another interesting approach is to map the QUBO problem to the optimization of a classical neural network. This has been recently investigated via the use of graph neural networks [34], autoregressive neural networks [35], and neural-network quantum states [36,37].

In this work, we explore an alternative quantum-inspired classical algorithm for QUBO problems. The algorithm is inspired by quantum annealing [9,38] and is called *local quantum annealing* (LQA). In a standard quantum annealing algorithm, one evolves a multiqubit quantum state through a Schrödinger evolution that is generated by a time-dependent Hamiltonian, where the ground-state solution of the final Hamiltonian encodes the solution to the problem. In LQA, we use the same Hamiltonian to define a time-dependent cost function that we iteratively optimize via a momentum-accelerated gradient-descent-based approach. In order to make the optimization tractable, the cost function is optimized over a subset of product quantum states that is guaranteed to contain the problem solution. In this way, the system stays in a low-energy (product) state throughout the optimization, which can be seen as an approximation of the annealing process. Since we use a gradient-descent approach on the energy landscape, the method is not, however, equivalent to simulating a type of adiabatic quantum evolution in which the system stays in the global minimum at all times, as is the case for quantum annealing. Surprisingly, however, for small systems the method is often able to find the global minimum via the route defined by the gradient-descent procedure and for larger systems gives a method for finding good approximate solutions in very short time.

We note that this approach is reminiscent of, but not equivalent to, those studied in Ref. [39], which propose to update the parameters via a dynamical physical evolution defined by the energy of the system. We comment more on this in Sec. II. Similarly to the SB and the SIM CIM, the computationally expensive step in the algorithm corresponds to a matrix-vector multiplication, which can be accelerated by dedicated hardware such as GPUs and field-programmable gate arrays (FPGAs) and so the algorithm is naturally parallelizable and requires approximately the same resources per optimization step as the SB and the SIM CIM. Unlike simulated bifurcation, however, our approach is purely gradient based, and unlike the SIM CIM, it does not consume randomness during the optimization. We note that our approach shares some similarities with the molecular dynamics part of the hybrid quantum annealing algorithm presented in Ref. [40], where it is possible that the recursive form of the leapfrog algorithm

used there plays a similar role to gradient descent in our approach.

We benchmark our algorithm against the three alternative versions of the SB algorithm [15] and the SIM CIM [19]. Since all these algorithms require the same computational resources per optimization step, this makes for a fair and uncontroversial comparison in terms of the solution quality per optimization step. We focus most of our benchmarking around recently developed methods of planted solutions [41–43], which provide constructions of QUBO problems with tunable hardness the global optimal solution of which is known. In our opinion, this provides a more complete picture than benchmarks that focus on a single problem or an arbitrary sets of problems the hardness of which is unknown. We find that LQA provides comparable solution quality to the SB and the SIM CIM over all problem instances and thus opens up an alternative route to large-scale QUBO optimization via a purely gradient-based deterministic algorithm. We also believe that the relative simplicity of our approach paves the way for a number of potential improvements or extensions that we discuss at the end of the paper.

II. METHODS

Formally, a QUBO optimization problem is one of the form

$$\min_{\sigma \in \{0,1\}^n} \sigma^T Q \sigma + \sigma^T \mathbf{a}, \quad (1)$$

where Q is a $n \times n$ real symmetric matrix such that $Q_{ii} = 0 \forall i$, \mathbf{a} is an $n \times 1$ real vector and σ is a $n \times 1$ binary vector. By defining the ± 1 -valued variables $s_i = 2\sigma_i - 1$ and the corresponding vector \mathbf{s} , the problem is equivalent (up to a problem-dependent constant) to

$$\min_{\mathbf{s} \in \{+1,-1\}^n} \mathbf{s}^T J \mathbf{s} + \mathbf{s}^T \mathbf{b}, \quad (2)$$

where $J = Q/4$, $\mathbf{b} = (\mathbf{a} + Q\mathbf{1})/2$, and $\mathbf{1}$ is the vector of ones. Note that the second term in Eq. (2) can be incorporated into the first at the expense of adding one extra variable with fixed value $+1$. From here on, we therefore consider problems in the above form with $\mathbf{b} = 0$, so that our problem becomes

$$\min_{\mathbf{s} \in \{+1,-1\}^n} \mathbf{s}^T J \mathbf{s}. \quad (3)$$

In this form, the optimization is equivalent to minimizing the energy of a classical Ising Hamiltonian J with no bias field, where the variables s_i are interpreted as classical spin values. In practice, many relevant problems from industry can be mapped into Ising optimizations of this form [44].

The minimum given in Eq. (3) is equivalently obtained as the ground state of the quantum Ising Hamiltonian

$$H_z = \sum_{ij} J_{ij} \sigma_z^{(i)} \sigma_z^{(j)}, \quad (4)$$

where $\sigma_z^{(i)}$ denotes the Pauli Z matrix applied to the i th qubit of an n qubit system. That is, one way to find the minimum given in Eq. (2) is via

$$\min_{|\psi\rangle \in \mathbb{C}^{2^n}} \langle \psi | H_z | \psi \rangle, \quad (5)$$

where $|\psi\rangle$ is a normalized quantum state. Since H_z is diagonal in the Z basis, this minimum energy will be attained by one (or more) of the Z -basis product states. In a quantum annealing algorithm, one attempts to solve Eq. (5) by considering a time-dependent Hamiltonian such as

$$H(t) = tH_z\gamma - (1-t)H_x, \quad (6)$$

where

$$H_x = \sum_i \sigma_x^{(i)}. \quad (7)$$

Here, $\gamma > 0$ controls the relative strength of the H_z contribution to the energy. The quantum state of the system is initially prepared in the state $|+\rangle^{\otimes n}$, which is the ground state of $H(0)$, and the system is evolved by changing the Hamiltonian from $t = 0$ to $t = 1$. If this evolution is done slowly enough, the adiabatic theorem guarantees that the state will stay in the ground state throughout the evolution and the global solution to the problem will therefore be obtained.

Our algorithm is inspired by the quantum annealing approach. Since a classical simulation of quantum annealing is likely impossible due to the exponential memory requirements needed to store the quantum state vector, we do not consider the set of all quantum states but restrict ourselves to product states of the form

$$|\theta\rangle = |\theta_1\rangle \otimes |\theta_2\rangle \otimes \cdots \otimes |\theta_n\rangle, \quad (8)$$

where

$$|\theta_i\rangle = \cos \frac{\theta_i}{2} |+\rangle + \sin \frac{\theta_i}{2} |-\rangle. \quad (9)$$

We therefore have

$$\langle \theta_i | \sigma_z | \theta_i \rangle = \sin \theta_i, \quad \langle \theta_i | \sigma_x | \theta_i \rangle = \cos \theta_i. \quad (10)$$

Considering these states, the minimization in Eq. (5) is equivalent to

$$\min_{\theta} \langle \theta | H_z | \theta \rangle = \min_{\theta} \sum_{ij} J_{ij} \sin \theta_i \sin \theta_j. \quad (11)$$

In analogy to the quantum annealing algorithm, we define a time-dependent cost function corresponding to the energy of the system at time t :

$$\begin{aligned} \mathcal{C}(t, \theta) &= \langle \theta | tH_z\gamma - (1-t)H_x | \theta \rangle \\ &= t\gamma \sum_{ij} J_{ij} \sin \theta_i \sin \theta_j - (1-t) \sum_i \cos \theta_i. \end{aligned} \quad (12)$$

In our algorithm, we further parametrize the values θ_i as $\theta_i = \pi/(2) \tanh(w_i)$, where $w_i \in \mathbb{R}$ so that $w_i \rightarrow \pm\infty \implies |\theta_i\rangle \rightarrow |0\rangle, |1\rangle$. The cost set out in Eq. (12) therefore becomes

$$\mathcal{C}(t, \mathbf{w}) = t\gamma \mathbf{z}^T \mathbf{J} \mathbf{z} - (1-t) \mathbf{x}^T \cdot \mathbf{1}, \quad (13)$$

with $\mathbf{z} = (\sin(\frac{\pi}{2} \tanh w_1), \dots, \sin(\frac{\pi}{2} \tanh w_n))$ and $\mathbf{x} = (\cos(\frac{\pi}{2} \tanh w_1), \dots, \cos(\frac{\pi}{2} \tanh w_n))$. The gradient of Eq. (13) with respect to the parameters \mathbf{w} is

$$\nabla_{\mathbf{w}} \mathcal{C}(\mathbf{w}, t) = \frac{\pi}{2} [t\gamma(2\mathbf{J}\mathbf{z}) \circ \mathbf{x} + (1-t)\mathbf{z}] \circ a(\mathbf{w}), \quad (14)$$

where \circ denotes element-wise multiplication of vectors and $a(\cdot) = 1 - \tanh^2(\cdot)$ is the derivative of the tanh function and $a(\cdot)$ acts element-wise.

In order to obtain an approximate solution to the QUBO problem, one may use a gradient-descent routine on $\mathcal{C}(t, \mathbf{w})$. More precisely, in each step i of the algorithm, one updates the parameters \mathbf{w} according to the gradient of $\mathcal{C}(t, \mathbf{w})$, where $t = i/N$ and N is the total number of steps. A spin configuration can be obtained at any time via $\mathbf{s} = \text{sign}(\mathbf{w})$. Here, one may choose from a plethora of strategies that exist for gradient-based optimization, such as momentum assistance and adaptive step sizes. For example, the standard momentum-assisted gradient-descent technique uses an additional velocity vector \mathbf{v} (typically initialized as the zero vector) and a momentum parameter $\mu \in [0, 1]$ and corresponds to the parameter update (at time t)

$$\mathbf{v} \leftarrow \mu \mathbf{v} - \eta \nabla \mathcal{C}_{\mathbf{w}}(\mathbf{w}, t) \quad (15)$$

$$\mathbf{w} \leftarrow \mathbf{w} + \mathbf{v}, \quad (16)$$

where η is the step size of the gradient descent. Momentum can help to accelerate the gradient descent in flat regions of optimization and is widely used in the training of machine-learning models. Another common technique is the ADAM update [45], which uses the momentum technique and further adapts the step size after each update. In the following pseudocode, we denote a generic update as $\mathbf{w} \leftarrow g(\mathbf{w}, \nabla \mathcal{C}_{\mathbf{w}}(\mathbf{w}, t))$, where g may incorporate additional information such as momentum and step size:

The effect of this algorithm is to continuously push the parameters toward a local minimum of the time-evolving cost function. One may therefore hope to mimic

```

Data:  $J \in \mathbb{R}^{n \times n}$ : symmetric Ising matrix;
          $\mathbf{w}_0 \in \mathbb{R}^n$ : initial weights;  $N$ : total steps
initialization  $\mathbf{w} = \mathbf{w}_0$ ;
for  $i = 1, \dots, N$  do
  |  $\mathbf{w} \leftarrow g(\mathbf{w}, \nabla \mathcal{C}_{\mathbf{w}}(\mathbf{w}, i/N))$ 
end
return  $\text{sign}(\mathbf{w})$ 

```

Algorithm 1. Local quantum annealing

the evolution of a quantum annealing algorithm, in which the system stays in the instantaneous ground state of the time-dependent Hamiltonian throughout the optimization. For this reason, it is best to initialize the parameters close to zero, since the ground state of $\mathcal{C}(\mathbf{w}, 0)$ is given by $|+\rangle^{\otimes n}$.

A couple of important points are in order here. First, note that the computationally expensive part of the algorithm is the matrix multiplication $J\mathbf{z}$ that appears in the gradient calculation in Eq. (14). This can be accelerated by dedicated hardware such as GPUs or FPGAs, which for large problems, results in a tremendous performance boost with respect to using a CPU. Second, we perform only a single parameter update for each value of t , rather than waiting for convergence to a local minimum before stepping t , which requires a much larger optimization time and typically leads to similar results. Even for large optimization times, the method is not guaranteed to converge to the globally optimal solution, since the optimization is performed over a much smaller space of product quantum states and there is no corresponding guarantee that following a locally optimal minimum throughout the optimization will lead to the ground state of the final Hamiltonian (due to, e.g., a first-order phase transition). Nevertheless, one may expect that the behavior approximates the quantum annealing behavior to some extent.

In Fig. 1, we plot the evolution of the cost, given by $\mathcal{C}(\mathbf{w}, t)$ for a simple problem involving 20 spins. Here, we also investigate the use of momentum-assisted gradient descent. Although the system does not stay in the global minimum throughout the optimization (the lower dashed curve, computed heuristically through the minimization of the ansatz with the basin-hopping algorithm [46]), the momentum-assisted approach eventually finds the ground state of the system via a different path. The pure gradient-descent algorithm has difficulty leaving its initial parameters $|+\rangle^{\otimes n}$ and more frequently does not converge to the global solution. This can be attributed to the fact that the initial local minimum becomes a saddle point from which it is very slow to escape without momentum assistance. We therefore suspect that momentum assistance is vital in achieving good performance, which we find to generally be the case.

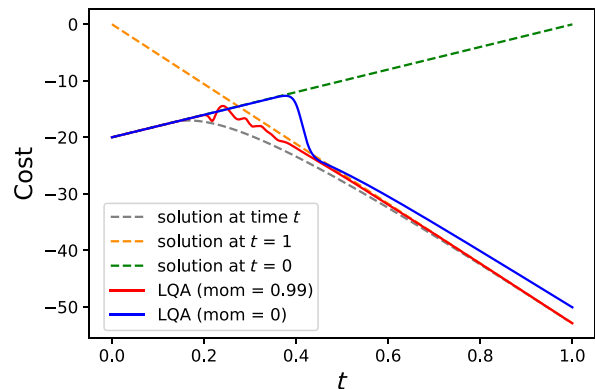


FIG. 1. A plot of the cost-function value as a function of t for a fully connected 20-spin problem, where the weights J_{ij} are uniformly drawn from the interval $[-1, 1]$. The dashed straight lines show the cost of the global solutions for the times $t = 0$ and $t = 1$, evaluated at different values of t (i.e., the energy crossing of the two ground states if the two Hamiltonian H_x and H_z were to commute). The lower dashed line shows the minimum of the cost function of the product-state ansatz in Eq. (8), obtained via an intensive basin-hopping algorithm. Here, we consider a vanilla gradient-descent approach (mom = 0) and a momentum-assisted approach (mom = 0.99). One sees that the use of momentum is particularly effective at helping the system leave the initial minimum, since this point becomes a saddle point for some value of t . This helps the momentum-assisted approach stay closer to the global minimum throughout the optimization and to reach the global minimum.

III. RESULTS

Here, we present a number of benchmarking results. We compare our algorithm against the three variants of simulated bifurcation—the original algorithm (SB) [14], ballistic simulated bifurcation (SBB) [15], and discrete simulated bifurcation (SBD) [15]—and the SIM CIM [19]. All algorithms are implemented in PyTorch on a standard laptop CPU. For LQA, the ADAM gradient-descent method [45] is adopted for the parameter updates.

We focus on three types of problem. The first is the K_{2000} Max-Cut problem [47] (see Fig. 2). This is a benchmark introduced in Ref. [10] that has been tested on both simulated bifurcation and the SIM CIM, which exploits the Max-Cut problem mapping to QUBO (see, e.g., Ref. [44]). Here, larger values correspond to higher-quality solutions. We find that LQA achieves the highest mean value over 100 optimization trials of 5000 steps each, with the SBD achieving the best value over all trials. The distribution of final values varies significantly over the algorithms and LQA and the SBB both feature a small variance with a strong peak in a single solution. We note that this behavior is sensitive to hyperparameter choice (one can achieve a larger variance at the cost of a lower mean value by adjusting the initial step size) and thus does not seem to be a generic feature of LQA (for more information, see Appendix B).

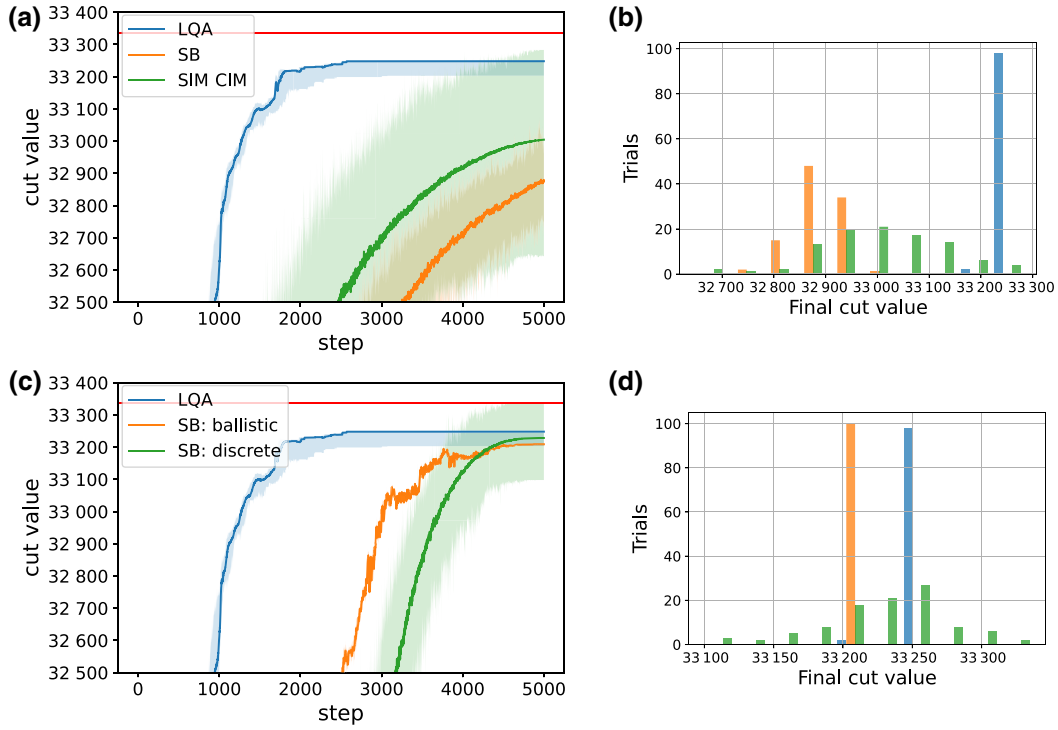


FIG. 2. Benchmarking using the K_{2000} Max-Cut problem introduced in Ref. [19]. This problem is equivalent to solving a 2000-spin fully connected Ising problem with $J_{ij} = \pm 1$ chosen to be uniformly random. Here, we compare LQA (a),(b) to the original simulated bifurcation algorithm [14] and the simulated coherent Ising machine [19] and (c),(d) to the more recent ballistic and discrete adaptations of the simulated bifurcation algorithm [15]. We perform 100 trials for each algorithm, each with 5000 optimization steps. The thick lines show the average values over the 100 trials as a function of the optimization step. The shaded regions correspond to the max and min cut values over all trials. The solid red line corresponds to the best-known cut value of 33 337 [15]. The histograms plot the final obtained cut values at the end of the optimization over the 100 trials. For LQA, we choose an initial step size of 1 and set $\gamma = 0.1$. For the SB, SBB, and SBD, the step size is set to 0.5, 1.25, and 1.25 respectively, with other parameters being set as recommended in Ref. [15]. For the SIM CIM, a step size of 1 is chosen and the noise parameter A_n is set to 0.25. In all algorithms, the initial parameters are set as $0.1 \cdot \text{RAND}$, where RAND is a random list with elements in the range $[-1, 1]$.

Next, we consider performance with respect to problems with planted solutions (see Fig. 3). To generate these problems, we make use of the CHOOK software package [43]. We consider two such classes: the first are the two-dimensional (2D) “tile-planting” problems [Figs. 3(a) and 3(b)] introduced in Ref. [41]. These problems are defined on a 2D lattice and although, in principle, solvable in polynomial time due to their planar nature, they are often challenging for heuristic solvers. Here, we consider 1024 spin problems that are constructed from the C2 and C3 tiles (for more details, see Ref. [41]). We generate 11 problem instances of this type, where the probability of using a C3 tile is increased linearly from 0 to 1 over the problem instances. It has been observed that this results in progressively harder problems, which is reproduced in our results. For these problems, the best results are obtained by the SBB, with LQA giving similar results to the SB.

We also study the class of Wishart planted solutions [Figs. 3(c) and 3(d)] defined in Ref. [42], which generate fully connected problems with an easy-hard-easy transition. Here, we consider 500-spin problems. We find a

very similar performance amongst all the algorithms, with the SBB and LQA performing the best in the hard central problem region. The SBD performs almost identically to the SIM CIM, so we do not show these results here for clarity. The SB has some problems with stability for the initial problem instance, which we suspect could be remedied with appropriate hyperparameter tuning. For problem instances at the latter easy tail of the sequence, it is known that the probability that local optimization methods will find the ground-state solution by chance increases significantly [42]; the relatively few trials in which the algorithms find the global minimum in these cases may thus be more an indication of luck for this particular batch of trials rather than a feature of the algorithm itself.

IV. DISCUSSION AND CONCLUSIONS

Our method is reminiscent of, but different to, the approaches suggested in Refs. [39,48], which use a similar product-state ansatz and Hamiltonian to mimic the effects of quantum annealing. These approaches are based

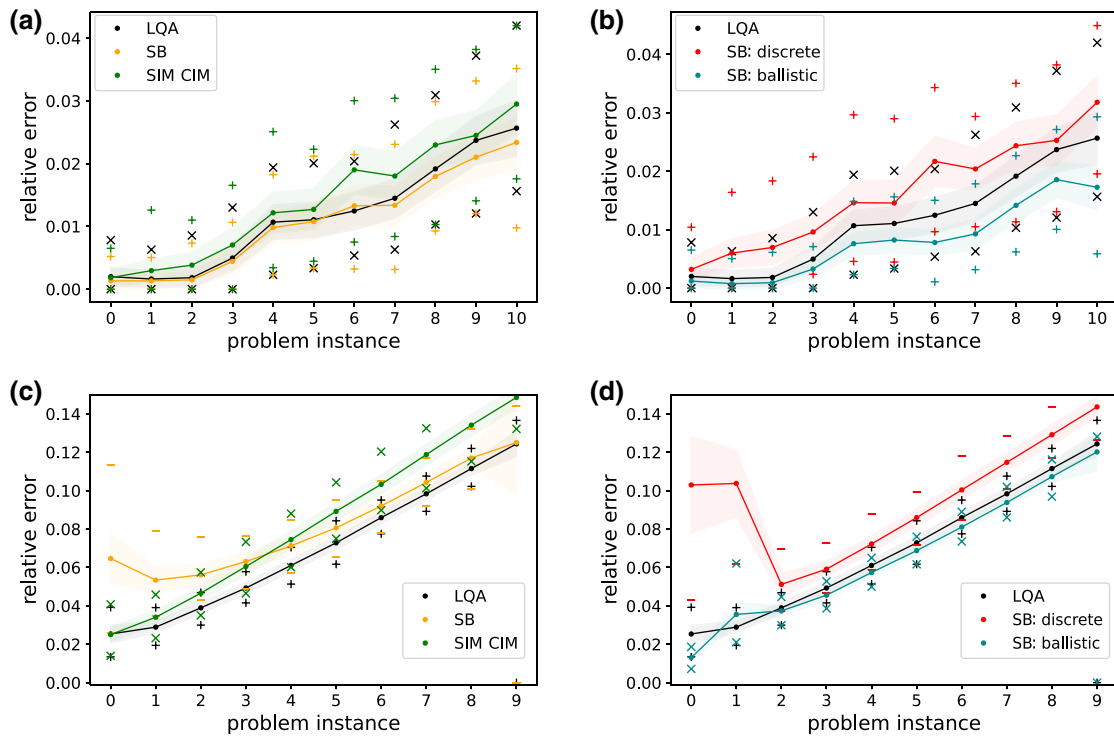


FIG. 3. (a),(b) Benchmark results for the tile-planting problems defined in Ref. [41], for 11 problem instances (x axis) with increasing hardness. The number of spins in each problem is 1024. For each instance, 500 trials with 500 steps are performed. The solid lines plot the average relative error to the known global minimum (defined as $|\frac{C-C_0}{C_0}|$, where C_0 is the global minimum and C is the value obtained in a given trial) over the 500 trials. The upper and lower colored markers denote the maximum and minimum values obtained over the 500 trials. The shaded regions correspond to one standard deviation around the mean. (c),(d) Analogous benchmarking results for the Wishart problems defined in Ref. [42], which feature an easy-hard-easy problem-hardness transition (note that the global minimum is obtained for the last instance for some algorithms). Here, ten problem instances are chosen and 500 trials are performed, each with 500 optimization steps. The initial step size for LQA is set to 2 for both benchmarks. For the SIM CIM, the step size is set to 1 and A_n to 0.25 for both benchmarks. For tile planting, the step sizes for the SB, SBB, and SBD are set to 0.5, 1.25, and 1.25. For the Wishart planting, the step sizes for the SB, SBB, and SBD depend on the problem instance due to instabilities in the optimization for the first two problem instances; for a table with the precise step sizes, see Appendix A. The parameter initialization is set as in Fig. 2.

on a dynamical (physical) evolution of the parameters θ under the Hamiltonian given in Eq. (6). In its most basic form, our method uses a simple gradient-descent update, which results in a different (unphysical) trajectory of the parameters θ . Under the momentum-assisted update in Eq. (15) and in the continuous-time limit (i.e., for infinitesimal step size), it is known that a physical interpretation of our approach is possible [49]: namely, as a dynamical evolution of a system in a viscous medium in which the momentum parameter plays the role of mass. This evolution is not equivalent to that of Refs. [39,48] due to the effects of damping implied by the viscosity. Furthermore, since our parameter updates are done at the level of the variables \mathbf{w} and not θ , the variables on which this physical evolution is understood are not the same. We find that these differences can result in significant differences in solution quality for large problems. For the case of more complex parameter updates, such as the ADAM [45] update that we use for our benchmarking, it is not clear if the parameter evolution can be understood physically. In any case, we

note that, as is typical with gradient-descent methods, the solution quality can be quite sensitive to the choice of step size and it is the case that large step sizes often outperform small step sizes, even for long optimization times. Given the considerations, our method could also be viewed as a type of gradient-descent-based graduated optimization [50–53] (also called the “continuation method”).

Our method also shares some similarities with the works of Hopfield and Tank [54], who have also considered time-dependent cost functions [55] inspired by the functioning of neurons in the brain. The fact that we view the individual systems as spins rather than neuron voltages results in a different parametrization and therefore a different optimization surface. The form of the time dependence in Ref. [55] [see Eq. (12)] is also different to ours: whereas we use an annealing-inspired time dependence, the time dependence in Ref. [55] is not fixed in advance but is a function of the current state of the variables. This likely results in significantly different optimization dynamics

The performance of the tested algorithms is similar throughout all benchmarks, with no algorithm clearly outperforming another. This is perhaps not surprising, since although they are all designed from quite different starting points, the parameter updates all make use of the same matrix-vector product between the coupling matrix J and a parameter-dependent vector that encodes the solution, which becomes more dominant as the optimization progresses. For LQA, this is the product Jz in Eq. (14). It is therefore unclear whether one should expect any large difference in performance between any of these algorithms, since they may all feature basins of attraction to similar solution qualities despite their differing parameter updates. We would argue, however, that LQA may be the most versatile of the options. First, it is purely gradient-descent based; it can therefore make use of myriad of tools from machine learning for gradient-based optimization, as well as being used as a subroutine in any other gradient-based algorithm. The use of ADAM in our tests makes it quite stable to initial step-size variation, which to some extent removes the burden of setting this hyperparameter. However, we currently do not have a systematic method to chose the parameter γ and so more work is needed here. Although not done here, second-order derivatives of the cost could be calculated analytically via Eq. (14). Thus, methods that make use of second-order information could be incorporated exactly and may give a performance boost.

We believe that the main value of the algorithm, however, is the form of the time-dependent cost function, since this results in significantly better solutions than using a static Hamiltonian. It would be interesting to investigate if this approach could be improved. For example, although we use a simple linear annealing schedule here, nonlinear schedules may give better results. On this note, there have been a number of works that use alternative transverse Hamiltonians in order to improve the performance of quantum annealing [56,57]. It would be interesting to investigate whether the different cost landscapes implied by these Hamiltonians could lead to improvements. Finally, it would be interesting to see if the product ansatz in Eq. (8) could be expanded to include a wider range of quantum states without significantly sacrificing the speed of the algorithm. Here, previous works connecting QUBO

optimization to neural network quantum states [36,58] and matrix product state [59] may be valuable.

An implementation of our algorithm in PyTorch is available as a PYTHON package [60].

ACKNOWLEDGMENTS

This work was supported by the Institute of Photonic Sciences (ICFO)–Quside Joint Laboratory in Quantum Processing, Fundacio Cellex, Fundacio Mir-Puig, Generalitat de Catalunya (SGR 1381, SGR 1341, Quantum Cataluña, and CERCA Program), European Research Council Advanced Grant NOvel Quantum simulators - connectIng Areas and CERQUTE, the Spanish Ministry of Economy and Competitiveness (MINECO) (Severo Ochoa CEX2019-000910-S, Plan National FIDEUA, and TRANQI, Retos QuSpin, FPI, QUANTERA Magnetic-Atom Quantum Simulator PCI2019-111828-2/10.13039/501100011033), the European Union Horizon 2020 program, Future and Emerging Technologies - Open Optical Topologic Logic (OPTO-Logic) (Grant No. 899794), and the National Science Centre, Poland (Symfonia Grant No. 2016/20/W/ST4/00314), Marie Skłodowska-Curie Grant Structured electric-dipole-based chirality No. 101029393, a fellowship granted by the Caixa Foundation (ID 100010434, fellowship code LCF/BQ/PR20/11770012), and the AXA Chair in Quantum Information Science.

Note added.—Recently, we became aware of an independent work [61] that also suggests using a gradient approach that is similar to ours.

APPENDIX A: PARAMETER SETTING FOR WISHART-PLANTING BENCHMARK

The step sizes used for the Wishart-planting benchmarking of Fig. 3 for each algorithm are displayed in Table I. The step sizes are set close to the largest values possible before instabilities cause the solution quality to deteriorate.

APPENDIX B: EFFECT OF STEP SIZE ON K_{2000} BENCHMARK

Here, we investigate the effect of varying the initial step size η for the K_{2000} benchmark of Fig. 2. The

TABLE I. The step-size settings used for Fig. 3.

	Instance									
	0	1	2	3	4	5	6	7	8	9
LQA	2	2	2	2	2	2	2	2	2	2
SB	0.5	1	1	1	1	1	1	1	1	1
SBB	1.0	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
SBD	0.1	0.1	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
SIM CIM	1	1	1	1	1	1	1	1	1	1

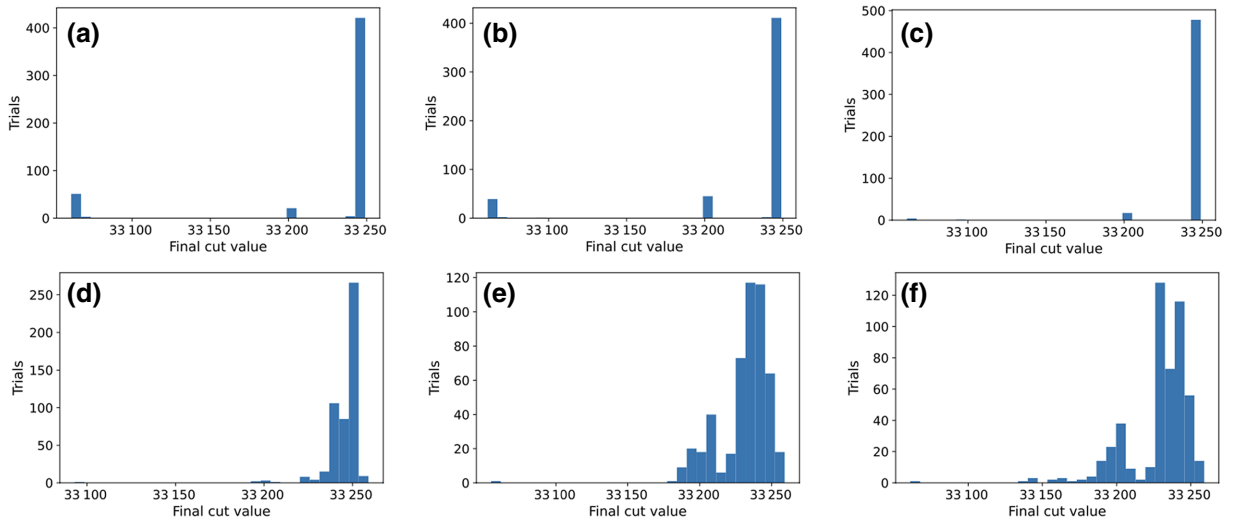


FIG. 4. Histogram plots of the final cut value obtained for the K_{2000} benchmark using LQA: step = (a) 0.1, (b) 0.5, (c) 1.0, (d) 1.5, (e) 2.0, and (f) 2.1. Five hundred trials are performed using the same parameters as for Fig. 2 but with the varying initial step sizes used in the ADAM gradient-descent update.

results are presented in Fig. 4 for the initial step sizes [0.1, 0.5, 1.0, 1.5, 2.0, 2.1]. For step sizes above 2.1, the optimization becomes unstable and the results are very poor. From the results, one sees that a larger step size results in a larger variety of solutions being obtained, at the expense of a lower average cut value. However, a solution with a cut value of 33 259 is obtained only for large step sizes. We therefore suspect that in order to obtain large values, a good strategy is to set the initial step size to its largest stable value and perform many trials until a high-quality solution is found.

APPENDIX C: TIME-TO-TARGET RESULT FOR K_{2000}

The plots in Fig. 2 suggest that although LQA does not achieve the highest cut value over all trials, it may be faster at achieving large (but not maximal) cut values than other algorithms. To investigate this, we consider the time-to-target (TTT) metric (see, e.g., Ref. [15] and references therein), defined as the computation time to achieve a specific value with 99% probability. The TTT can be calculated as follows:

$$t_{\text{target}} = T_{\text{trial}} \frac{\log(1 - 0.99)}{\log(1 - P_T)}, \quad (\text{C1})$$

where T_{trial} is the computation time per trial and P_T is the probability of achieving the target value (or better) in a given trial, which is estimated by performing many trials. We consider two target cut values: (i) a target of 33 249, which corresponds to the maximum cut obtained by LQA in Fig. 2, and (ii) a target of 33 003, which is approximately equal to 99% of the best-known cut value of 33 337.

For each of the algorithms considered, 150 trials are performed for $N = 1000, 2500, 5000,$ and $10\,000$ and four values of the corresponding P_T estimated. The highest P_T

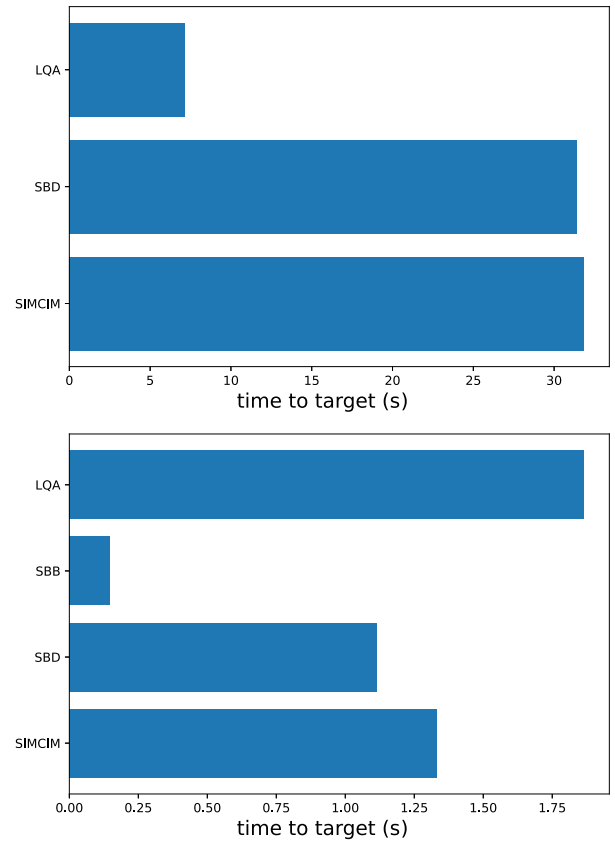


FIG. 5. Time-to-target results for the K_{2000} graph, for cut values 33 249 (top) and 33 003 (bottom). Only those algorithms that achieve the cut values in at least one trial are shown.

obtained from these four estimations is taken to calculate the TTT as above. The computations are performed using a Tesla K80 GPU. Figure 5 shows the results for the two target values, for those algorithms that achieve the target value at least once in one of the trials. For target (i), LQA is indeed significantly faster at reaching the target than all other algorithms. However, for the lower value of target (ii), LQA is outperformed by all algorithms except the SB. This suggests that in its current form, LQA is no faster in general at reaching large cut values than the other algorithms.

-
- [1] S. Yarkoni, E. Raponi, S. Schmitt, and T. Bäck, Quantum annealing for industry applications: Introduction and review, arXiv preprint [arXiv:2112.07491](https://arxiv.org/abs/2112.07491) (2021).
- [2] E. Boros and P. L. Hammer, The Max-Cut problem and quadratic 0–1 optimization; polyhedral aspects, relaxations and bounds, *Ann. Oper. Res.* **33**, 151 (1991).
- [3] F. Glover, G. Kochenberger, and Y. Du, Quantum bridge analytics I: A tutorial on formulating and using QUBO models, *4OR* **17**, 335 (2019).
- [4] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv preprint [arXiv:1411.4028](https://arxiv.org/abs/1411.4028) (2014).
- [5] P. Date, R. Patton, C. Schuman, and T. Potok, Efficiently embedding QUBO problems on adiabatic quantum computers, *Quantum Inf. Process.* **18**, 1 (2019).
- [6] M. Jünger, E. Lobe, P. Mutzel, G. Reinelt, F. Rendl, G. Rinaldi, and T. Stollenwerk, Performance of a quantum annealer for Ising ground state computations on chimera graphs, arXiv preprint [arXiv:1904.11965](https://arxiv.org/abs/1904.11965) (2019).
- [7] J. Lee, A. B. Magann, H. A. Rabitz, and C. Arenz, Towards favorable landscapes in quantum combinatorial optimization, arXiv preprint [arXiv:2105.01114](https://arxiv.org/abs/2105.01114) (2021).
- [8] 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).
- [9] T. Kadowaki and H. Nishimori, Quantum annealing in the transverse Ising model, *Phys. Rev. E* **58**, 5355 (1998).
- [10] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeki, K. Enbutsu, O. Tadanaga, H. Takenouchi, K. Aihara, K.-i. Kawarabayashi, K. Inoue, S. Utsunomiya, and H. Takesue, A coherent Ising machine for 2000-node optimization problems, *Science* **354**, 603 (2016).
- [11] S. Boettcher, Analysis of the relation between quadratic unconstrained binary optimization and the spin-glass ground-state problem, *Phys. Rev. Res.* **1**, 033142 (2019).
- [12] D-Wave systems documentation, <https://docs.dwavesys.com/docs/latest/index.html>.
- [13] K. Boothby, P. Bunyk, J. Raymond, and A. Roy, Next-generation topology of D-Wave quantum processors, arXiv preprint [arXiv:2003.00133](https://arxiv.org/abs/2003.00133) (2020).
- [14] H. Goto, K. Tatsumura, and A. R. Dixon, Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems, *Sci. Adv.* **5**, eaav2372 (2019).
- [15] H. Goto, K. Endo, M. Suzuki, Y. Sakai, T. Kanao, Y. Hamakawa, R. Hidaka, M. Yamasaki, and K. Tatsumura, High-performance combinatorial optimization based on classical mechanics, *Sci. Adv.* **7**, eabe7953 (2021).
- [16] T. Okuyama, T. Sonobe, K.-i. Kawarabayashi, and M. Yamaoka, Binary optimization by momentum annealing, *Phys. Rev. E* **100**, 012111 (2019).
- [17] S. Patel, L. Chen, P. Canozza, and S. Salahuddin, Ising model optimization problems on a FPGA accelerated restricted Boltzmann machine, arXiv preprint [arXiv:2008.04436](https://arxiv.org/abs/2008.04436) (2020).
- [18] Y. Haribara, H. Ishikawa, S. Utsunomiya, K. Aihara, and Y. Yamamoto, Performance evaluation of coherent Ising machines against classical neural networks, *Quantum Sci. Technol.* **2**, 044002 (2017).
- [19] E. S. Tiunov, A. E. Ulanov, and A. Lvovsky, Annealing by simulating the coherent Ising machine, *Opt. Express* **27**, 10288 (2019).
- [20] M. C. Strinati and C. Conti, Multidimensional hyperspin machine, arXiv preprint [arXiv:2203.16190](https://arxiv.org/abs/2203.16190) (2022).
- [21] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, A 20k-spin Ising chip to solve combinatorial optimization problems with CMOS annealing, *IEEE J. Solid-State Circuits* **51**, 303 (2015).
- [22] K. Yamamoto, W. Huang, S. Takamaeda-Yamazaki, M. Ikebe, T. Asai, and M. Motomura, in *Proceedings of the 8th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies* (Tsukuba Japan, 2017), p. 1.
- [23] Y. Kihara, M. Ito, T. Saito, M. Shiomura, S. Sakai, and J. Shirakashi, in *2017 IEEE 17th International Conference on Nanotechnology (IEEE-NANO)* (IEEE, Pittsburgh, USA, 2017), p. 256.
- [24] S. Tsukamoto, M. Takatsu, S. Matsubara, and H. Tamura, An accelerator architecture for combinatorial optimization problems, *Fujitsu Sci. Tech. J* **53**, 8 (2017).
- [25] K. Yamamoto, K. Ando, N. Mertig, T. Takemoto, M. Yamaoka, H. Teramoto, A. Sakai, S. Takamaeda-Yamazaki, and M. Motomura, in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)* (IEEE, San Francisco, USA, 2020), p. 138.
- [26] K. P. Kalinin, A. Amo, J. Bloch, and N. G. Berloff, Polaritonic XY-Ising machine, *Nanophotonics* **9**, 4127 (2020).
- [27] C. Bauckhage, E. Brito, K. Cvejovski, C. Ojeda, R. Sifa, and S. Wrobel, in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition* (Springer, Venice, Italy, 2017), p. 3.
- [28] P. Date, D. Arthur, and L. Pusey-Nazzaro, Qubo formulations for training machine learning models, *Sci. Rep.* **11**, 1 (2021).
- [29] C. S. Calude, M. J. Dinneen, and R. Hua, QUBO formulations for the graph isomorphism problem and related problems, *Theor. Comput. Sci.* **701**, 54 (2017).
- [30] C. Bauckhage, N. Piatkowski, R. Sifa, D. Hecker, and S. Wrobel, in *Lernen. Wissen. Daten. Analysen. (LWDA)* (Berlin, 2019), p. 54.
- [31] C. Papalitsas, T. Andronikos, K. Giannakis, G. Theocharopoulos, and S. Fanarioti, A QUBO model for the traveling salesman problem with time windows, *Algorithms* **12**, 224 (2019).

- [32] 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).
- [33] M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura, and H. G. Katzgraber, Physics-inspired optimization for quadratic unconstrained problems using a digital annealer, *Front. Phys.* **7**, 48 (2019).
- [34] M. J. Schuetz, J. K. Brubaker, and H. G. Katzgraber, Combinatorial optimization with physics-inspired graph neural networks, arXiv preprint [arXiv:2107.01188](https://arxiv.org/abs/2107.01188) (2021).
- [35] M. Hibat-Allah, E. M. Inack, R. Wiersema, R. G. Melko, and J. Carrasquilla, Variational neural annealing, arXiv preprint [arXiv:2101.10154](https://arxiv.org/abs/2101.10154) (2021).
- [36] J. Gomes, K. A. McKiernan, P. Eastman, and V. S. Pande, Classical quantum optimization with neural network quantum states, arXiv preprint [arXiv:1910.10675](https://arxiv.org/abs/1910.10675) (2019).
- [37] T. Zhao, G. Carleo, J. Stokes, and S. Veerapaneni, Natural evolution strategies and variational Monte Carlo, *Mach. Learn.: Sci. Technol.* **2**, 02LT01 (2020).
- [38] 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).
- [39] J. A. Smolin and G. Smith, Classical signature of quantum annealing, *Front. Phys.* **2**, 52 (2014).
- [40] H. Irie, H. Liang, T. Doi, S. Gongyo, and T. Hatsuda, Hybrid quantum annealing via molecular dynamics, *Sci. Rep.* **11**, 1 (2021).
- [41] D. Perera, F. Hamze, J. Raymond, M. Weigel, and H. G. Katzgraber, Computational hardness of spin-glass problems with tile-planted solutions, *Phys. Rev. E* **101**, 023316 (2020).
- [42] F. Hamze, J. Raymond, C. A. Pattison, K. Biswas, and H. G. Katzgraber, Wishart planted ensemble: A tunably rugged pairwise Ising model with a first-order phase transition, *Phys. Rev. E* **101**, 052102 (2020).
- [43] D. Perera, I. Akpabio, F. Hamze, S. Mandra, N. Rose, M. Aramon, and H. G. Katzgraber, CHOOK—a comprehensive suite for generating binary optimization problems with planted solutions, arXiv preprint [arXiv:2005.14344](https://arxiv.org/abs/2005.14344) (2020).
- [44] A. Lucas, Ising formulations of many NP problems, *Front. Phys.* **2**, 5 (2014).
- [45] D. P. Kingma and J. Ba, ADAM: A method for stochastic optimization, arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014).
- [46] D. J. Wales and J. P. K. Doye, Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms, *J. Phys. Chem. A* **101**, 5111 (1997).
- [47] <https://github.com/haribara/SA-complete-graph/releases/tag/WK2000/>, repository where the K2000 matrix can be found.
- [48] T. Hatomura and T. Mori, Shortcuts to adiabatic classical spin dynamics mimicking quantum annealing, *Phys. Rev. E* **98**, 032136 (2018).
- [49] N. Qian, On the momentum term in gradient descent learning algorithms, *Neural Netw.* **12**, 145 (1999).
- [50] E. Hazan, K. Y. Levy, and S. Shalev-Shwartz, in *International Conference on Machine Learning* (PMLR, New York, USA, 2016), p. 1833.
- [51] M. Gargiani, A. Zanelli, Q. Tran-Dinh, M. Diehl, and F. Hutter, Convergence analysis of homotopy-SGD for non-convex optimization, arXiv preprint [arXiv:2011.10298](https://arxiv.org/abs/2011.10298) (2020).
- [52] H. Mobahi and J. Fisher III, in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29 (2015).
- [53] Z.-Y. Liu and H. Qiao, GNCCP—graduated nonconvexity and concavity procedure, *IEEE Trans. Pattern Anal. Mach. Intell.* **36**, 1258 (2013).
- [54] J. J. Hopfield and D. W. Tank, Computing with neural circuits: A model, *Science* **233**, 625 (1986).
- [55] J. J. Hopfield and D. W. Tank, “Neural” computation of decisions in optimization problems, *Biol. Cybern.* **52**, 141 (1985).
- [56] H. Nishimori and K. Takada, Exponential enhancement of the efficiency of quantum annealing by non-stoquastic Hamiltonians, *Front. ICT* **4**, 2 (2017).
- [57] Y. Susa, Y. Yamashiro, M. Yamamoto, and H. Nishimori, Exponential speedup of quantum annealing by inhomogeneous driving of the transverse field, *J. Phys. Soc. Jpn* **87**, 023002 (2018).
- [58] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, *Science* **355**, 602 (2017).
- [59] B. Bauer, L. Wang, I. Pizorn, and M. Troyer, Entanglement as a resource in adiabatic quantum optimization, arXiv:1501.06914 (2015).
- [60] <https://github.com/josephbowles/localquantumannealing>.
- [61] M. T. Veszeli and G. Vattay, Mean field approximation for solving QUBO problems, arXiv preprint [arXiv:2106.03238](https://arxiv.org/abs/2106.03238) (2021).