Ising machines as hardware solvers of combinatorial optimization problems

Naeimeh Mohseni[1,2,3], *Peter L. McMahon*[4] ✉ *and Tim Byrnes*[1,5,6,7,8] ✉

Abstract | Ising machines are hardware solvers that aim to find the absolute or approximate ground states of the Ising model. The Ising model is of fundamental computational interest because any problem in the complexity class NP can be formulated as an Ising problem with only polynomial overhead, and thus a scalable Ising machine that outperforms existing standard digital computers could have a huge impact for practical applications. We survey the status of various approaches to constructing Ising machines and explain their underlying operational principles. The types of Ising machines considered here include classical thermal annealers based on technologies such as spintronics, optics, memristors and digital hardware accelerators; dynamical systems solvers implemented with optics and electronics; and superconducting-circuit quantum annealers. We compare and contrast their performance using standard metrics such as the ground-state success probability and time-to-solution, give their scaling relations with problem size, and discuss their strengths and weaknesses.

[1]*State Key Laboratory of Precision Spectroscopy, School of Physical and Material Sciences, East China Normal University, Shanghai, China.*

[2]*Max-Planck-Institut für die Physik des Lichts, Erlangen, Germany.*

[3]*Department of Physics, Erlangen-Nuremberg, Erlangen, Germany.*

[4]*School of Applied and Engineering Physics, Cornell University, Ithaca, NY, USA.*

[5]*New York University Shanghai, Shanghai, China.*

[6]*NYU-ECNU Institute of Physics at NYU Shanghai, Shanghai, China.*

[7]*National Institute of Information and Communications Technology, Tokyo, Japan.*

[8]*Department of Physics, New York University, New York, NY, USA.*

✉*e-mail: pmcmahon@cornell.edu; tim.byrnes@nyu.edu*

https://doi.org/10.1038/s42254-022-00440-8

Conventional computers have particular difficulties in solving hard combinatorial optimization problems. Such problems typically involve finding an optimal configuration, defined by a cost function, among a very large number of potential candidate configurations. Examples of such problems include the travelling salesman problem, Boolean satisfiability (SAT) problems and MaxCut, to name a few. In a practical setting, such combinatorial optimization problems are of relevance to applications such as planning, logistics, manufacturing, financial portfolio management, computer vision, artificial intelligence, machine learning, bioinformatics, drug design and a variety of chemical and physical materials problems[1–4].

In many cases, such combinatorial optimization problems are instances of non-deterministic polynomial-time-complete (NP-complete) problems, which represent the hardest problems within the NP class. However, if there were a way of solving any combinatorial optimization problem in the NP-complete class with an improvement over conventional computing methods, the impact for a large number of practical applications would be enormous. This is because a well-known result states that it is possible to map any problem in NP to an NP-complete problem in polynomial time[5,6].

To give an example of such an NP-complete problem, consider MaxCut (FIG. 1a). One starts with a graph, in which some of the vertices are connected via edges (that is, links). The aim is to group the vertices into two types such that the number of edges between the two groups is as large as possible. MaxCut is of direct relevance to problems such as circuit design[7,8], machine learning[9] and computer vision[10,11], and therefore even without any mapping is an important problem in its own right. A brute-force solution of MaxCut requires checking every possible grouping of vertices; the number of such groupings is exponential in the number of vertices. The MaxCut problem can be recast in physics language as a spin glass problem (FIG. 1b). To do so, a binary-valued spin $\sigma_i \in \pm 1$ is put on each vertex, and the interaction constant $J_{ij} = 1$ between the connected vertices and 0 otherwise. The value of spin then encodes which group a vertex is in, and lowers the overall energy for connected spins i and j if they are in different groups (that is, if $\sigma_i\sigma_j = -1$). This can be written as an Ising Hamiltonian

$$H_{\mathrm{P}} = \sum_{i,j=1}^{N} J_{ij}\sigma_i\sigma_j + \sum_{i=1}^{N} h_i\sigma_i, \qquad (1)$$

where N is the number of vertices or spins. A linear h_i term is included for generality, although for MaxCut it is not required. Finding the minimum energy of equation (1) is then equivalent to solving MaxCut. We note that the Ising Hamiltonian (1) can be related to a quadratic unconstrained binary optimization (QUBO) problem under a simple change of variables $\sigma_i = 1 - 2x_i$, $x_i \in \{0,1\}$, and hence they can be regarded as equivalent problems.

Such optimization problems are commonly solved on large-scale high-performance classical computers, using variants of Monte Carlo methods. With the demise of Moore's law, it is of interest whether alternative methods — perhaps based on unconventional methods

Key points

- Dedicated hardware solvers for the Ising model are of great interest, owing to their many potential practical applications and the end of Moore's law, which motivate alternative computational approaches.
- Three main computing methods that Ising machines use are classical annealing, quantum annealing and dynamical system evolution. A single machine can operate on the basis of multiple computing approaches.
- Today, Ising hardware based on classical digital technologies is the best performing for common benchmark problems. However, the performance is problem-dependent, and alternative methods can perform well for particular classes of problems.
- For particular crafted problem instances, quantum approaches have been observed to have superior performance over classical algorithms, motivating quantum hardware approaches and quantum-inspired classical algorithms.
- Hybrid quantum–classical and digital–analogue algorithms are promising for future development; they may harness the complementary advantages of both.

of computing — could be used to solve such optimization problems. Alternatives to Turing's concept of a deterministic digital computing machine[12] have a long history, particularly the analogue computers used for physical simulators to investigate complex problems[13]. In an analogue computer, the computation is performed using coupled physical systems that evolve continuously according to their physical dynamics. They are implemented by analogue electronics or mechanical systems, for example. Analogue computers were used predominantly in the first half of the twentieth century when digital computing speeds were insufficient, and they continued to be used for several decades for specialized applications such as flight simulation, although even these applications have been now rendered obsolete.

Interest in the field of quantum simulation[14–16] in many ways mirrors this development of classical analogue computers and led to a resurgence of interest in realizing analogue simulators of the Ising model. Quantum simulation was in fact one of the early motivations for realizing a quantum computer, based on Richard Feynman's conjecture that a quantum computer could simulate quantum systems more efficiently than classical computers[14] — proven over a decade later[17]. Although a large-scale, fault-tolerant quantum computer is still a challenging goal, technologically, advances in the manipulations of many-body quantum systems using cold atoms, ions or artificial qubits potentially allow for a way of simulating complex quantum systems without requiring the full controllability of a quantum computer[15,16,18–22]. This insight led to the idea that Ising models might be realizable using a quantum simulation approach[23,24], in which alternative models of computation could be used to find the ground state more efficiently. The first large-scale physical implementation of the quantum approach was a 128-qubit quantum annealer realized by D-Wave Systems, followed by larger-scale systems[25,26]. Today, there are numerous approaches, incorporating a variety of techniques (both classical and quantum), which will be described and compared in this Review.

In the classical realm, one of the main drawbacks of analogue computers compared with digital computers is that they are more susceptible to error, owing to the analogue storage of information. Nevertheless, analogue computers can have several advantages over digital computers. First, the operation of the analogue computer is typically highly parallelized. For a system consisting of many coupled systems that encode information, each system evolves in parallel, in contrast to digital computers in which parallelization is performed across multiple processors. Second, there is no additional overhead arising from the implementation of digital logic. In many cases the time evolution of a physical system is continuous, but in a simulation on a digital computer it is discretized and evolved in a step-wise sequence, which requires additional resources not required in analogue simulation. Analogue computing is in many ways analogous to the way the brain operates: there is no predefined algorithm, and its operation is inherently massively parallel and asynchronous. This 'natural computing' approach has intrigued researchers for decades, both from the point of view of improvements over current computing, and for understanding how biological systems compute.

In this Review, we survey hardware devices that have been developed with the aim of solving the Ising model; we call such devices 'Ising machines'. An important caveat is what exactly we mean by solving the Ising model. In many applications, suboptimal but still good solutions are acceptable in practice; hence we consider primarily heuristic and approximate solvers. We focus on discussing their underlying operating principles[27] and introduce the types of technologies that have been used to implement them. The technologies include variations of classical thermal annealers, quantum annealers, and dynamical system-based solvers including the coherent Ising machine, which have attracted interest in the past decade. We also describe other types of computing devices such as those based on hybrid quantum–classical systems. We discuss the performance of the investigated devices, focusing on the scaling with regard to the size of the Ising problem.

Operating principles of Ising machines

Classical thermal annealing

One of the fundamental concepts that is encountered in connection to solving the Ising model — and optimization problems in general — is annealing. Inspired by concepts in statistical mechanics, the configurations corresponding to the lowest values of a cost function (or energy, in the context of physics) are found by gradually lowering the effective temperature of a system. The basic observation is that at thermal equilibrium, a classical physical system follows statistics according to a Boltzmann (or Gibbs) distribution

$$p_n = \frac{\exp\left(-\frac{E_n}{k_B T}\right)}{Z}, \tag{2}$$

where $Z = \sum_n \exp(-\frac{E_n}{k_B T})$, k_B is the Boltzmann constant, T is thermodynamic temperature and E_n is the energy of each of the 2^N spin configurations of the Ising model (1), labelled by $n = \{\sigma_1, \ldots, \sigma_N\}$. The lowest-energy states appear with higher probability, and the probability of obtaining the ground state, that is, the desired solution

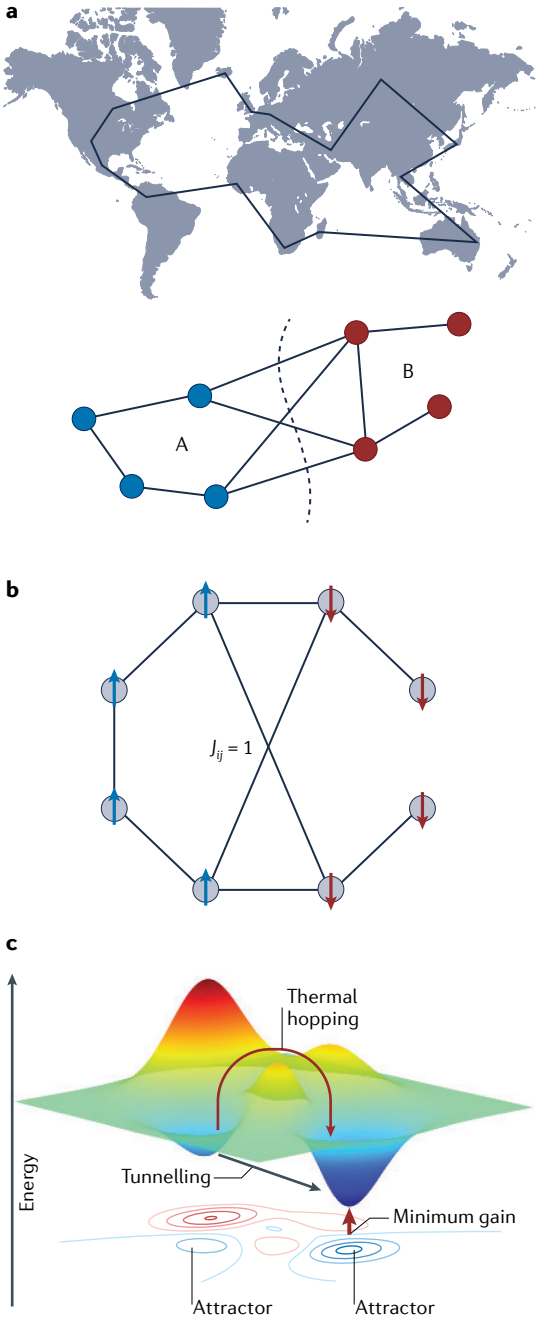

Fig. 1 | **Combinatorial problems, the Ising model, and its energy landscape. a** | Examples of combinatorial optimization problems. In the travelling salesman problem (top), the aim is to find the shortest possible route that visits each city exactly once and returns to the origin city. The travelling salesman problem can be mapped onto the Ising model by encoding the information of the city and its route ordering as a spin variable. The total number of spins required is the square of the number of cities. For MaxCut (bottom), the the optimal division is indicated by the dashed line. **b** | An example of an eight-spin Ising model, equivalent to the MaxCut problem in part **a**. On each node is a two-valued spin (arrow). Edges correspond to assigning an coupling between spins i and j of $J_{ij} = 1$. **c** | Schematic energy landscape of the Ising model and some mechanisms used in Ising machines to overcome local minima: thermal excitations used in classical thermal annealing, quantum tunnelling in quantum annealing, the minimum gain principle in coherent Ising machines, and attractors in dynamical system evolution.

of the Ising model, increases as the temperature is lowered. To produce such a state at thermal equilibrium, the system evolves according to a master equation, typically of the form

$$\frac{\mathrm{d}p_n}{\mathrm{d}t} = -w_{nm}p_n + w_{mn}p_m, \tag{3}$$

where p_n is the probability of being in the nth energy state and w_{nm} is the rate for transition from the nth to the mth state. The rates are taken such that in the limit that time $t \to \infty$ the probability distribution follows (2). Evolving equations (3) long enough guarantees obtaining the low-energy solutions for a sufficiently low temperature. The main problem is that particular energy landscapes require extremely long times before thermal equilibrium is reached, owing to the possibility of getting trapped in a local minimum (FIG. 1c). The solution to this problem is to gradually lower the temperature or anneal the system such that at each temperature the system has a chance to equilibrate. By reducing the temperature with an inverse logarithmic dependence on time, one is guaranteed to obtain the ground state[28].

As a classical computer algorithm, simulated annealing (SA) remains one of the most popular algorithms that can be applied to optimization problems. On a classical digital computer, it is preferable to perform an equivalent stochastic sampling approach, rather than run equation (3) directly, owing to the exponential resources required. As such, typically one uses a Monte Carlo algorithm employing the Metropolis–Hastings algorithm[29,30], such that the desired Boltzmann distribution is obtained. Substantial improvement over SA is obtained by more sophisticated classical algorithms, such as parallel tempering[31,32], population annealing[33] and isoenergetic cluster moves[34], to mention a few. For both parallel tempering and population annealing, multiple copies of the system are prepared in random initial states. For parallel tempering, each copy has a different temperature parameter. The temperature is increased for the copies that perform poorly and is decreased for the ones that perform successfully. In population annealing, poorly performing copies are probabilistically removed and those that perform successfully are replicated, while reducing the temperature[35,36].

Simulated annealing has also been implemented in dedicated hardware using digital hardware accelerators and analogue natural computing approaches, providing the chance to exploit the parallelization of such hardware. For analogue computation, numerous physical implementations of Ising and related models have been realized or proposed, including magnetic devices[37–45], optics[46–48], memristors[49,50], spin-switches[51], quantum dots[52], single atoms[53], microdroplets[54] and Bose–Einstein condensates[24,55] (FIG. 2). For example, stochastic magnet tunnel junctions can act as probabilistic bits, which thermally fluctuate between either parallel or antiparallel mutual orientations of magnetic domains[37,38,40] (FIG. 2a).

An arbitrary Ising interaction between the coupled bits is realized by measuring the orientation of the bits and adjusting the barrier energy between the two orientations. This set-up was used to factor integers by an adiabatic procedure[39]. In another approach, memristors have been used to perform an analogue matrix multiplication of the

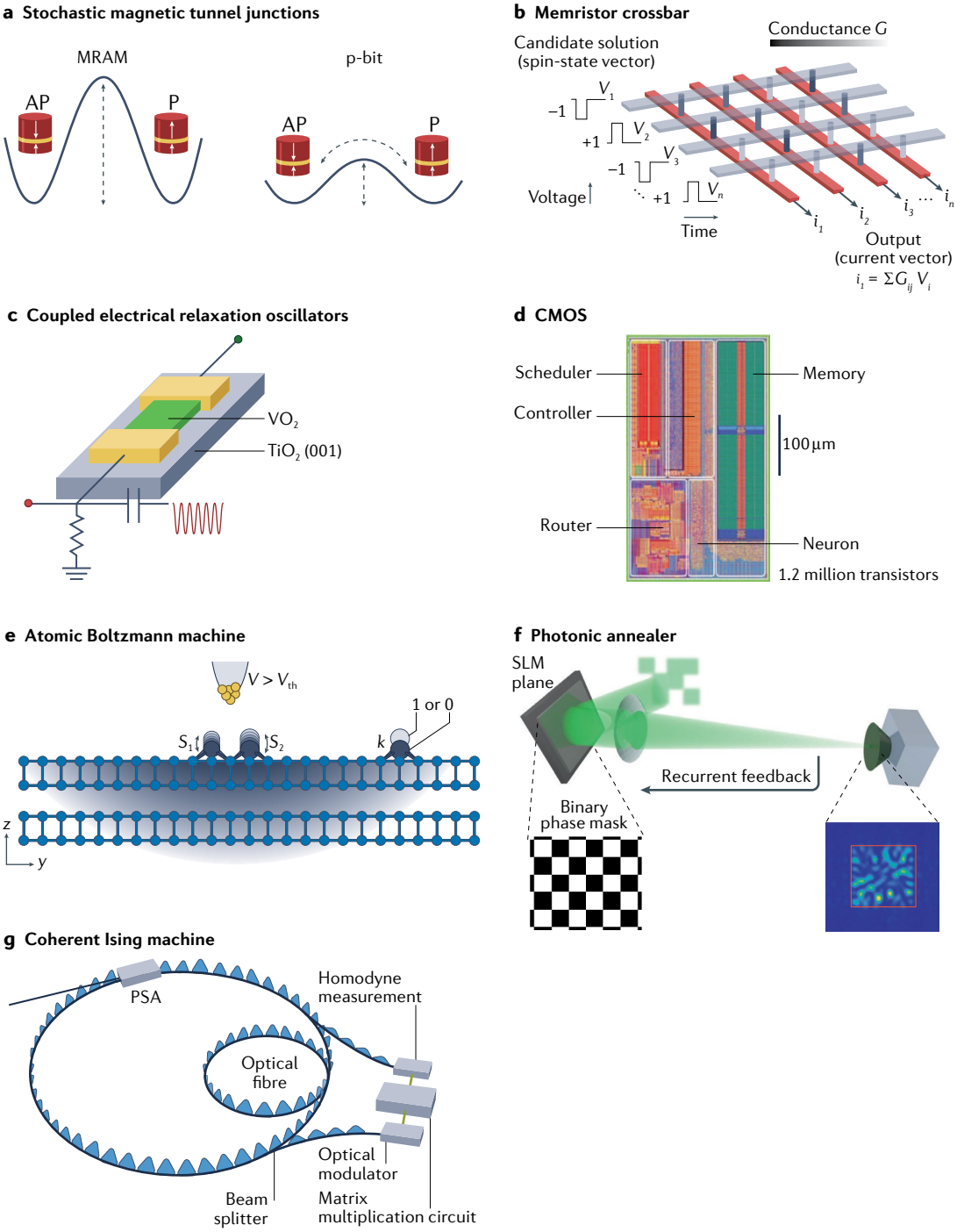

Fig. 2 | **Example technologies used to realize various types of Ising machines. a** | Stochastic magnetic tunnel junctions such as a probabilistic bit (p-bit; right) have a lower energy barrier ΔE between parallel (P) and antiparallel (AP) orientations of the magnetic layers, compared with conventional magnetoresistive random-access memory (MRAM; left). **b** | Memristor crossbar array to perform matrix-vector multiplication[49]. **c** | Metal–insulator VO_2 system to realize coupled electrical oscillators. **d** | Complementary metal–oxide–semiconductor (CMOS) chip with 28-nm transistors, which realizes a 1-million-spin Boltzmann machine. **e** | In a Boltzmann machine realized using Co atoms (grey dots) on the surface of black phosphorus (blue dots) interacting with a scanning tunnelling microscope, spins s_1 and s_2 switch stochastically whereas spin k remains fixed. The gate voltage V is above the threshold V_{th} for stochastic switching. **f** | Spatial light modulator (SLM)-based photonic annealer. **g** | Coherent Ising machine measurement–feedback loop. PSA, phase-sensitive amplifier. Part **a** is adapted with permission from REF.[39]. Part **c** is adapted with permission from REF.[100]. Part **d** is adapted with permission from REF.[69]. Part **e** is adapted with permission from REF.[53]. Part **f** is adapted with permission from REF.[46]. Part **g** is adapted with permission from REF.[223].

Ising matrix to evaluate the energy; using the intrinsic hardware noise, they performed a highly parallelized implementation of a Ising model annealer[49] (FIG. 2b). In optical systems, the Ising model was realized by encoding the spins using the phase of the light, and a recurrent feedback network was used to produce the Ising couplings[46–48] (FIG. 2f). This system converges towards the Boltzmann distribution (2), with the primary advantage being the fast parallelized spin updates.

For digital-electronic approaches, hardware accelerators using CMOS application-specific integrated circuits (ASICs)[56–60] and field-programmable gate arrays (FPGAs)[61–65] have been investigated to solve the Ising model as a type of domain-specific computing. For example, complementary metal–oxide–semiconductor (CMOS) circuits have been used to implement 2×10^4 Ising spins; each spin interacts with up to five local spins[56]. Random thermal effects were introduced by either introducing random spin-flips during calculation of spin values or applying a low supply voltage to the memory cells, which also introduces randomness at the level of the hardware. In another approach, an 8,192-spin Ising machine with full connectivity was realized, based on a digital-CMOS-chip implementation of SA, where spin updates are performed in parallel[57,58,61,62]. The parallelization allows for a large speed-up in comparison to a serial implementation of SA. We note that in the context of machine-learning accelerators[66], hardware implementations of Boltzmann machines have been investigated with CMOS ASICs[67–69] (FIG. 2d), FPGAs[70–75] and graphics processing units (GPUs)[76,77]. The similarity of the underlying energy model of Boltzmann-machine hardware accelerators suggests that such technologies could be adapted to act as Ising solvers. For instance, an FPGA implementation of the restricted Boltzmann machine's stochastic sampling algorithm to solve the Ising problem has been demonstrated[64]. In this case, the problem is mapped to a bipartite version and each group of spins is updated by applying parallel SA[60,78]. The inherent parallelism of this architecture allows parallel sampling, which provides substantial improvement over SA.

Dynamical system solvers

In a thermal annealer, at any given point of time during the evolution, the system is ideally in a state that is at thermal equilibrium, following the Boltzmann distribution. Likewise, as discussed below, in a quantum annealer the system ideally remains in the ground state of the instantaneous Hamiltonian. To ensure these conditions, annealing must proceed sufficiently slowly to maximize the probability that the minimum energy state of the Ising model is obtained. In contrast to such annealing-based approaches, alternative strategies exist, in which the system evolution is much faster than thermal equilibriation and adiabatic timescales. In such dynamical system approaches, the state of the system is driven towards the lowest energy state of the Ising model. An early example of such a dynamical solver used electronic circuits to realize equation (1)[79,80]. In this section, we explain three types of dynamical system solvers: coupled oscillators, coherent Ising machines and chaotic systems.

Oscillator-based computing. Dating from the 1950s, the 'parametron' computer is a pioneering type of analogue computer, based on coupled oscillators[81–83]. The state information, such as the configuration of an Ising spin, is represented by the phase of an oscillator. In the presence of a nonlinearity, an oscillator with resonant frequency ω_0 can be phase-locked with a pump frequency $2\omega_0$ with two possible stable phases, 0 or π, that represent the digital information. In its original conception, information processing in the parametron computer occurs as a sequence of logical gates. However, it was also shown that a computation can be performed in a more parallel, natural computing approach (as reviewed in REFS[84,85]). Such coupled oscillators can be used for solving combinatorial optimization problems, such as the Ising model.

The basic idea of oscillator-based computing can be captured by the Kuramoto model, which describes a system of oscillators mutually coupled by an interaction[86–88]. Consider N oscillators, labelled by index i, that oscillate with frequency ω_0. Denote the phase of the ith oscillator by ϕ_i. Mutually coupling the oscillators, the dynamical system can be described by

$$\frac{\mathrm{d}\phi_i}{\mathrm{d}t} = \omega_0 + K \sum_j J_{ij} \sin(\phi_i - \phi_j) + K h_i \sin(\phi_i - \omega_0 t), \qquad (4)$$

where K is a coupling parameter that controls the overall contribution of the Ising dynamics. In the rotating frame, the $\sin(\phi_i - \phi_j)$ factor has two steady-state solutions $\frac{\mathrm{d}\phi_i}{\mathrm{d}t} = 0$, where the phases are either in or out of phase. The $\sin(\phi_i - \omega_0 t)$ term is also stable when $\phi_i - \omega_0 t = 0, \pi$. Thus, the system converges to a particular configuration of phases; in a simulation of the Ising model a spin readout can be performed from this configuration. For the case of constant J_{ij} and $h_i = 0$, the Kuramoto model can be analytically solved to show a dynamical phase transition between unsynchronized and synchronized oscillators, for particular interaction strengths. This type of dynamics has been applied to numerous artificial intelligence problems, such as image processing, pattern recognition and generation[84,85].

Such oscillators, implemented as a system of coupled LC circuits, have been proposed as a means to solve the graph colouring problem[89,90]. The aim is to colour the vertices of a graph with k colours such that no adjacent vertices have the same colour; for $k \geq 3$ this is an NP-complete problem. For $k = 2$, the oscillator scheme is able to correctly find solutions, but for the more difficult $k = 3$ case the scheme only succeeded for a subset of problem instances[89]. The approach was theoretically further developed, including explicitly extending to the case of solving Ising problems and analysing various possible physical implementations[91–96]. Oscillator networks have been experimentally demonstrated with systems such as bulk analogue electronic oscillators[93,97–99], the VO_2 insulator-to-metal transition[100–102] (FIG. 2c), spin oscillators[103,104] and integrated CMOS electronic oscillators[105–107]. These systems have been used to solve problems such as graph colouring, maximum independent set and the Ising model. In several of these studies, the network was enabled to find low-energy solutions

of the Ising model by adding noise and turning on the interactions smoothly.

A related approach called memcomputing uses networks of Boolean logic gates as a dynamical system solver. These have conceptual similarities with oscillator-based Ising machines, even if they are not explicitly constructed from networks of oscillators[108,109]. Such approaches have been applied successfully to frustrated-loop Ising model instances[110,111].

Coherent Ising machine. Oscillator-based Ising machines of a particular class, dubbed 'coherent Ising machines' (CIMs), are naturally suited to being implemented with optical oscillators[112–121] (FIG. 2g). Each Ising spin σ_i in a CIM is encoded in the phase ϕ_i of light in an optical mode. To enforce binary spin values $\phi_i = 0,\pi$, CIMs use degenerate optical parametric oscillators (DOPOs), which are a form of parametric oscillator in which phase-sensitive gain yields oscillations either in-phase or out-of-phase with respect to the oscillator's pump light[113,122]. Each DOPO represents a single spin and is part of a network of DOPOs that are coupled together such that the coupling between a pair of DOPOs is proportional to the Ising spin–spin coupling J_{ij}. Several ways to realize couplings have been proposed[24,113–116], but for the experimental demonstrations performed thus far, the details of the coupling scheme are not crucial for understanding the Ising-solving capability of each CIM implementation.

Modelling DOPOs as classical oscillators, the time evolution of a CIM can be modelled by the following system of coupled differential equations in the rotating frame[113,117]:

$$\frac{da_i}{dt} = -\gamma a_i + r a_i^* - \kappa |a_i|^2 a_i - g \sum_j J_{ij} a_j - g h_i + n_i, \qquad (5)$$

where each a_i is a complex number representing the optical field in the ith mode, γ is the decay rate of the photons from each mode, r is the amplification provided by DOPO gain, κ is the coefficient of nonlinear loss due to OPO gain saturation, g is a coupling constant determining the strength of the Ising interactions, and n_i are Langevin noise operators associated with the photon decay and nonlinear gain. The notation * denotes complex conjugation. The effect of the Ising terms $-g\sum_j J_{ij} a_j - g h_i$ can be thought of as additional loss terms that act on the ith mode.

The key differences between these equations of motion and those for Kuramoto oscillators (4) are that the oscillator amplitudes are explicitly considered, in addition to their phases, and there are loss and gain terms; it is these terms that are responsible for DOPOs having an oscillation threshold. For a single DOPO, the loss and gain terms result in the DOPO being bistable: above threshold, a DOPO oscillates either exactly in-phase or exactly out-of-phase. Because the Ising terms can be interpreted as spin-configuration-dependent loss, one can interpret the DOPO network as having a collective-oscillation threshold that is lowest when the Ising terms are smallest, and hence when the represented spin configuration has minimum energy. If the CIM is operated such that the gain is slowly increased from 0 (where the DOPO network is below threshold) to ever-higher values — that is, r is not a constant, but rather a monotonically increasing function of time — then, in the absence of noise n_i, the DOPO configuration with the lowest loss should oscillate first (see the minimum gain principle illustrated in FIG. 1c) and the solution to the Ising problem can be read out by measuring the phases of the light from each DOPO. An important point to note is how slowly r can be increased and have the CIM still oscillate in the ground state for a length of time sufficient to allow measurement: even in the complete absence of noise (which is not experimentally realistic, but can be programmed in a computer simulation), the CIM does not find the exact solution to arbitrary Ising problems in polynomial time. An important technicality that arises in the CIM model (5), and in other oscillator-based Ising machines in which the oscillators have both amplitude and phase degrees of freedom (as opposed to just phase), is that if the amplitudes $|a_i|$ of the oscillators are not equal, then the system tends to minimize the energy of an Ising instance with a different J_{ij} matrix from the desired one. This phenomenon is sometimes referred to as a broken mapping due to amplitude heterogeneity. An intuitive fix is to add a feedback mechanism that forces the amplitudes $|a_i|$ to be equal, as has been studied for XY machines[123] and Ising machines[118].

The classical description of a CIM (equation (5)) is sufficient to explain the results obtained so far in experimental demonstrations[114–116,119,124–126], because these experiments have used DOPOs with relatively large round-trip (photon) loss. However, with sufficiently low loss, each DOPO can generate an appreciable amount of quadrature squeezing, and in this regime the CIM's dynamics are more faithfully modelled quantum mechanically[117]. An interpretation for CIM operation that arises in the quantum-mechanical formulation is that each DOPO begins in a squeezed state that is approximable by a coherent superposition of in-phase and out-of-phase coherent states $|\alpha\rangle + |-\alpha\rangle$, so the below-threshold state of the CIM is one in which every spin configuration is represented in superposition. When the CIM goes through threshold, one of the configurations is selected. It is an open question to what extent quantumness of the DOPO network may improve (or impair) the computational performance of a CIM[120]. A quantum model for a machine conceptually similar to a quantum-regime CIM, in which superpositions $|\alpha\rangle + |-\alpha\rangle$ are also formed, has been studied[127]. The machine acts as an adiabatic quantum computer when the pump rate (the equivalent of r in the CIM model) is increased from 0 sufficiently slowly. This theoretical connection suggests that insights into the solution mechanisms of quantum annealers might be helpful for understanding CIMs, especially CIMs in which the coupling between DOPOs is conservative rather than dissipative, and vice versa.

Besides the CIM, there have been proposals and demonstrations of several types of optical and optoelectronic Ising and Ising-like machines in addition to those cited in the subsection on thermal annealers: systems based on coupled lasers[128–131], optoelectronics[132], exciton-polaritons[92,133–136] and electromechanical systems[137,138].

Chaos in dynamical system solvers. In an ergodic system, the dynamics are such that the system visits all parts of configuration space. This is an attractive idea in the context of solving the Ising model, since in many approaches getting trapping in local minima is the cause of the exponential slowdown. Numerical studies studying thermal relaxation have showed that the process is strongly non-ergodic, and does not visit all parts of configurational space[139]. Several studies have suggested that modifying the dynamics to include chaos would yield an improvement in performance[118,140–142].

Limit-cycle-free dynamical systems have been designed with fixed-point attractors that are the solutions of a given optimization problem[140]; in particular, *k*-SAT, which is — like the Ising problem — an NP-complete decision problem with an NP-hard optimization version. The formulation of the dynamical system involved both state variables s_i corresponding to the variables in the *k*-SAT problem (analogous to spin variables for an Ising problem) and auxiliary variables a_i. The dynamical system has an appealing theoretical property: it avoids becoming stuck in local minima of the *k*-SAT cost function. However, this comes at a price: the auxiliary variables grow exponentially in time. As a result, an analogue hardware implementation of the dynamical system requires an exponentially growing amount of energy to operate (a prototype CMOS demonstration[143] for problems with up to 50 variables artificially capped the signals representing the auxiliary variables at 1 V). In addition, a digital hardware implementation that integrates the differential equations requires exponentially small timesteps because the differential equations become stiff[140,141].

Numerical simulations indicate that the dynamical system undergoes a transient period of chaos when solving difficult instances of the *k*-SAT problem, but not when solving easy instances[140]. It was therefore suggested that chaos might be unavoidable in approaches to solving hard optimization problems. A discrete-map optimization algorithm[144] applied to solving both *k*-SAT and Ising problems also exhibits chaotic dynamics. The general approach in REF.[140] for designing a limit-cycle-free dynamical system that avoids being trapped in *k*-SAT local minima through the use of auxiliary variables has been adopted for Ising solving[118], and has been implemented and benchmarked with an FPGA[142].

Quantum approaches

Quantum annealing. Quantum annealing (QA)[4,145–147] is a heuristic algorithm based on the quantum adiabatic theorem as proposed in REF.[148] and has been studied in the context of the Ising model[149]. In this algorithm, the system is initially prepared in the known ground state of a Hamiltonian H_0. A common choice for this initial Hamiltonian is

$$H_0 = -\sum_{i=1}^{N} \sigma_i^x, \tag{6}$$

where N denotes the number of qubits, σ_i^x denotes the Pauli x operator on the ith qubit, and the ground state is the uniform superposition of all possible configurations $|+\rangle^{\otimes N}$, where $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$. The Hamiltonian is gradually re-weighted to the desired problem Hamiltonian H_P according to

$$H = (1-\lambda(t))H_0 + \lambda(t)H_\mathrm{P}, \tag{7}$$

where $\lambda(t) \in [0,1]$ is the annealing schedule. The annealing process can be viewed as H_0 introducing quantum fluctuations originating from the non-commutability of H_P and H_0. These fluctuations are gradually reduced to reach the low-energy configuration of the classical energy function H_P. Based on the quantum adiabatic theorem, for a sufficiently slow sweep, the system remains in its instantaneous ground state throughout the evolution[145,150]. The sweep time for which adiabaticity can be achieved is proportional to a negative power of the minimum energy gap between two lowest-energy levels during the sweep[151–154].

The use of quantum fluctuations in QA has been hypothesized as a potential resource for a speed-up over classical methods. Quantum tunnelling allows the system to pass through energy barriers (FIG. 1c). However, despite several decades of investigation, the computational role of coherent tunnelling in providing speed-up is not completely understood[145,155,156]. Part of the reason for this is the difficulty of simulating QA on classical computers due to the large computational overhead. The only quantum hardware that has so far been able to directly test QA with a large number of qubits is that developed by D-Wave Systems. Although this technology still suffers from limitations such as the presence of decoherence, control errors and limited connectivity, several studies have shown that quantum effects do play a role in the D-Wave machine[157–159]. For problem instances that possess tunnelling barriers, QA and quantum-inspired classical algorithms that mimic tunnelling[157] have been shown to have an advantage over SA.

Hybrid quantum–classical algorithms. The aim of variational quantum algorithms[160] is to solve classical and quantum optimization problems by combining a parametrized quantum circuit with a classical optimizer to obtain the variational parameters. The parametrized quantum circuit can be thought of as preparing a variational quantum state, which is optimized to give the lowest-energy state of a given Hamiltonian. These algorithms are believed to be strong candidates to achieve a practical quantum advantage on noisy intermediate-scale quantum (NISQ) devices[161]. In the context of combinatorial optimization problems, the quantum approximate optimization algorithm (QAOA)[162] has particularly attracted a lot of interest, partially as a result of the existence of theoretical guarantees on the approximation ratio that it can achieve for certain classes of optimization problems[162,163].

The QAOA algorithm can be viewed as a Trotterized version of QA with a parametrized annealing pathway[164]. The system is initially prepared in $|+\rangle^{\otimes N}$, the ground state of the Hamiltonian (6). The parametrized quantum circuit transfers the initial state to the ground state of the target problem Hamiltonian (in the ideal case) by

alternately applying the unitary operator corresponding to the problem Hamiltonian $e^{-i\gamma_j H_P}$ and the unitary operator $e^{-i\beta_j H_0}$. This sequence generates the following quantum variational state

$$|\psi(\boldsymbol{\beta},\boldsymbol{\gamma})\rangle = e^{-i\beta_p H_0} e^{-i\gamma_p H_P} \cdots e^{-i\beta_1 H_0} e^{-i\gamma_1 H_P} |+\rangle^{\otimes N}, \quad (8)$$

where $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \cdots, \gamma_p) \in [0, 2\pi]^p$ and $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_p) \in [0, \pi]^p$ are $2p$ variational parameters and p determines the circuit depth. Next, a classical optimizer is applied to find the optimal $\boldsymbol{\beta},\boldsymbol{\gamma}$ that optimizes the energy expectation $E(\boldsymbol{\beta},\boldsymbol{\gamma}) = \langle\psi(\boldsymbol{\beta},\boldsymbol{\gamma})|H_P|\psi(\boldsymbol{\beta},\boldsymbol{\gamma})\rangle$ by updating the variational parameters iteratively. Various approaches have been applied for this classical optimization step such as brute-force grid search[162], gradient descent methods[165] and machine learning[166]. A key feature of QAOA is that the computational power increases with p (REFS[164,167]) in contrast with QA, in which the performance does not always improve with annealing time[162]. Under reasonable complexity-theoretic assumptions, QAOA with $p = 1$ cannot be efficiently simulated with classical computers[168], or it implies that P = NP. This result has led to speculations that QAOA may be able to demonstrate a quantum computational advantage in the context of an optimization problems on near-term quantum computers[168]. However, the class of problems that can be solved efficiently with shallow circuits may not be representative for problems of practical interest. For example, for all-to-all connected Ising models and MaxCut, it has been shown that deep circuits may be required[169]. Therefore, benchmarking computational advantages of QAOA against classical algorithms requires going far beyond problems that can be solved with a shallow circuit and instead exploring the power of QAOA at intermediate depths. However, at current technological levels such circuits are prone to decoherence and gate errors[169].

QAOA has been demonstrated at the small scale on platforms such as superconducting qubits[169], photonics[170] and trapped ions[167]. So far, no large-scale (that is, $N > 50$) demonstrations of QAOA have been experimentally performed. We note that classical simulations showing expectation results for single-layer ($p = 1$) QAOA on problems with up to $N = 10^5$ spins have been performed[171], but as is the case with quantum annealers, it is expected that large-scale quantum hardware will be needed to properly evaluate the performance of QAOA in general.

Other quantum algorithms. Several other quantum algorithms have been proposed to solve combinatorial optimization problems[172]. These include using approaches based on amplitude amplification, and quantum simulated annealing (not to be confused with simulated quantum annealing below). In these approaches, the aim is to prepare the quantum Gibbs state, a superposition state with Boltzmann probabilities (2) as the amplitudes. The quantum Gibbs state is attained by performing a quantum walk such that after many iterations the desired coherent Gibbs state is obtained[173–175]. Then, in a similar way to thermal annealing, the temperature is gradually lowered to obtain a low-energy state.

Other classical algorithms

Quantum-inspired classical algorithms. Inspiration from quantum algorithms has led to proposals for new types of classical algorithms. Such quantum-inspired algorithms are run on conventional computing hardware or on digital hardware accelerators, and hence are classical approaches, but use concepts that originate from quantum mechanics in the algorithm. We briefly summarize several approaches in this direction.

In simulated quantum annealing (SQA), quantum Monte Carlo is applied to estimate the low-energy states of the QA Hamiltonian[176–178]. To perform the quantum Monte Carlo, a stoquastic QA Hamiltonian is mapped to a classical Hamiltonian by introducing an extra spatial dimension, corresponding to imaginary time. A stoquastic Hamiltonian is characterized by having only non-positive off-diagonal elements in the computational basis. The new Hamiltonian has equivalent equilibrium properties to the original QA Hamiltonian[179]. The mapping can be implemented either in discrete time by applying the Trotter–Suzuki decomposition, or in continuous time by applying a path integral[179]. Quantum Monte Carlo in the continuous time limit samples the equilibrium thermal state of a quantum system (as opposed to directly simulating its unitary time evolution) and can generate Boltzmann-distributed states (2). At sufficiently low temperatures, SQA can mimic tunnelling effects. SQA can also generate entangled ground states that occur during the adiabatic evolution. It can thus faithfully predict the performance of QA for stoquastic Hamiltonians.

Several other quantum-inspired classical algorithms based on dynamical system evolution have been proposed. In simulated CIMs, the equations modelling the CIM are simulated on a classical computer, and used as an algorithm to solve the Ising model. It has been shown that such a simulation has a speed-up compared to a physical implementation of a CIM applying FPGA[180] and GPU[181]. The key observation here is that such simulations are described by a set of coupled equations of the form (4) or (5), which scale with the number of spin variables N, rather than the configurational space 2^N. Thus, a simulation of the coupled-oscillator system is efficient.

Another approach is simulated bifurcation (SB), which is based on simulating adiabatic evolutions of classical nonlinear Hamiltonian dynamical systems. This algorithm is the classical counterpart of bifurcation-based adiabatic quantum computation[127]. Two branches of the bifurcation in each nonlinear oscillator represents two states of each Ising spin. In 2019, Toshiba developed an FPGA- and GPU-based SB machine showing excellent performance due in part to its high parallelizability[182,183]. The operational mechanism of the SB algorithm operates based on an adiabatic and ergodic search. Later, two other variants of SB were introduced, called the ballistic simulated bifurcation algorithm (bSB) and the discrete simulated bifurcation algorithm (dSB)[184], which far outperform the original SB in terms of both speed and solution accuracy. These new algorithms apply new approaches, such as a quasi-quantum tunnelling effect. Recently, a multi-chip architecture using a partitioned version of the

SB algorithm was implemented with FPGAs, showing that the method can handle large-scale Ising problems[185]. Both CIM simulations and the SB algorithm are parallelizable, by simultaneously updating at each time step N coupled-oscillator variables. In contrast, SA canonically involves sequential updates of spins, with simultaneous updates allowed only for isolated spins.

Yet another quantum-inspired algorithm involves tensor networks, which are a powerful framework that provides representations of complex quantum states based on their entanglement structure[186]. Tensor networks have been applied as an ansatz to solve optimization problems[186,187]. Such an approach was used in the context of dynamic portfolio optimization, which can be encoded as an Ising problem[188].

Machine-learning approaches. Meanwhile, owing to the synergy between machine learning and combinatorial optimization algorithms, a new era at the interface of both fields is growing to take the best of both and develop new methods to deal with combinatorial optimization problems. In particular, the emergence of methods that are more sample-efficient makes them more scalable to large-scale problems. Machine-learning algorithms can be applied to either boost the performance of traditional classical solvers and quantum algorithms[189–191] or can work as a stand-alone solver[187]. For example, some machine-learning algorithms have been applied to accelerate Monte Carlo simulations[190,191]. Deep-learning-based methods applying reinforcement learning[192,193], graph neural networks[194–196] and neural attention mechanism[197] have also been investigated as solvers for combinatorial optimization problems.

Computational complexity

How do the computing approaches discussed in this Review relate to computational complexity? What are the prospects for devising an Ising machine that can solve Ising problems efficiently (that is, in polynomial time)? Although the $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$ question remains an open problem, it is widely conjectured that $\mathrm{P} \neq \mathrm{NP}$, in other words, that certain problems in NP, including the Ising problem, are fundamentally more difficult to solve than those in P. Indeed, decades of work in computer science, physics, mathematics and operations research has failed to find a polynomial-time algorithm that solves any NP-complete problem. The explosion of interest in quantum computing since the 1990s was kicked off by the discovery that integer factorization could be performed in polynomial time on a quantum computer[198]. It is thus conjectured that the BQP complexity class — decision problems that a quantum computer can solve in polynomial time with an error probability of at most 1/3 — is a larger class than P, that is, $\mathrm{P} \subseteq \mathrm{BQP}$. The associated class for a probabilistic Turing machine, the BPP class, is meanwhile conjectured to be equivalent to P, that is, $\mathrm{P} = \mathrm{BPP}$. This conjecture in general remains unproven, but is true if a suitable pseudorandom number generator is available[199]. Although there is no proof that quantum or probabilistic computers cannot solve NP-complete problems such as the Ising problem in polynomial time, it is considered unlikely[200].

The complexity-class arguments above concern solving the Ising problem in the sense of being able to find the exact ground state for all possible problem instances (all possible J_{ij}, h_i in equation (1)). However, as mentioned in the Introduction, an approximate solution with an energy close to the true ground state is often acceptable for practical applications. Three approaches for solving combinatorial optimization problems, such as the Ising problem, are exact algorithms, approximation algorithms and heuristic algorithms. Exact algorithms are designed to find solutions in a way that guarantees that the returned solutions are exactly optimal. Approximation algorithms return solutions that are not necessarily optimal but are guaranteed to be within a certain distance of optimality. Heuristic algorithms return solutions without any guarantee on their quality; because of this lack of theoretical guarantee, the primary basis for trusting an heuristic algorithm is from previous empirical (benchmarking) results. Both approximation and heuristic algorithms tend to be practical to run on large problems.

One consideration for approximation algorithms is what solution quality (formally, approximation ratio) can be guaranteed. The MaxCut problem, and hence the Ising model, is approximable-hard (APX-hard). Consequently, assuming $\mathrm{P} \neq \mathrm{NP}$, there exists no polynomial-time approximation algorithm for the Ising problem that guarantees a solution arbitrarily close to the exact solution[201]. However, there does exist a polynomimal-time approximation algorithm for MaxCut that finds solutions a fixed distance from the optimal solution: the Goemans–Williamson algorithm is guaranteed to find solutions within about 12% of the optimal value[202]. It is also known that it is NP-hard to approximate MaxCut with solutions guaranteed to be closer than about 6% to the optimal[203], so it is expected (assuming $\mathrm{P} \neq \mathrm{NP}$) that no polynomial-time approximation algorithm that achieves this approximation closeness is possible. In many practical settings, it is desirable to find solutions to MaxCut problems that have distance from the optimal solutions better than ~12% or even ~6%, which motivates the use of heuristic algorithms to solve MaxCut in practice.

Most Ising machines are heuristic solvers — that is, they can be thought of as physical machines that realize heuristic optimization algorithms. As such, they typically do not provide any approximation-ratio guarantees. The potential advantages of Ising machines largely lie outside the realm of complexity theory: there is the possibility that Ising machines have a polynomially improved scaling or constant-prefactor advantage over existing heuristic algorithms running on conventional processors. In other words, it is generally expected that Ising machines, regardless of their underlying algorithm or practical hardware implementation, still require exponential runtimes to achieve near-optimal solutions, but the exponent or the constant factor in front of it may be smaller than for a conventional solver. A small difference in the exponent can make a large difference in runtimes for large problem sizes; the fast clock speeds of various physical implementations, which give rise to constant-factor improvements, could lead to significant practical speed-ups compared with conventional state-of-the-art solvers.

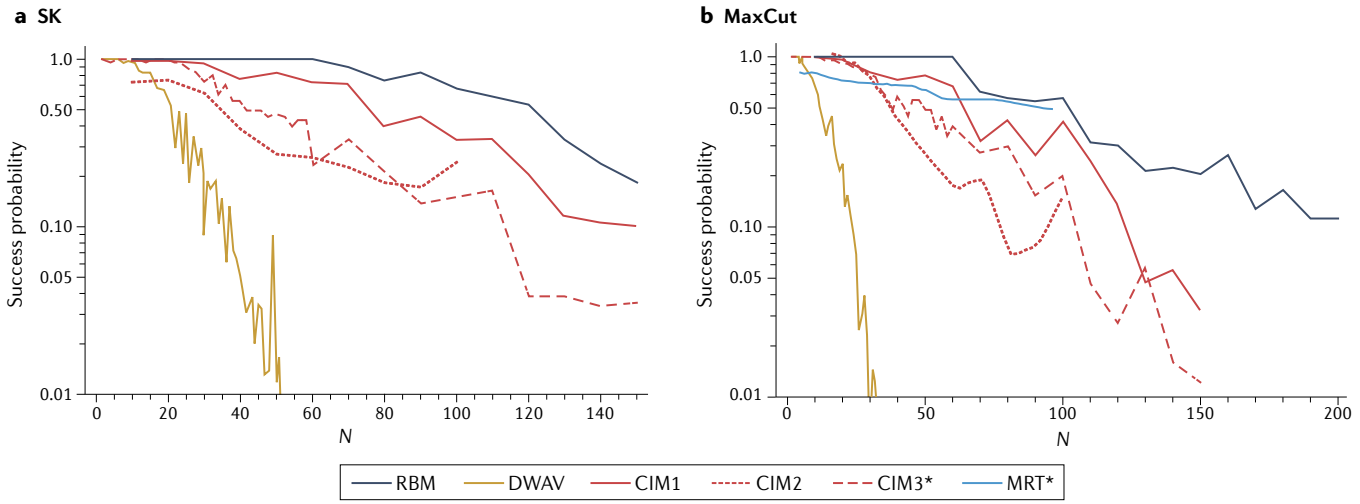

Fig. 3 | **Success probability comparison of Ising machines. a**,**b** | The probability of obtaining the ground state is shown for the Sherrington–Kirkpatrick (SK) problem (**a**) and dense MaxCut problem (**b**). For the SK problem, the coupling J_{ij} between spins i and j is chosen from ±1 with equal probability. The MaxCut problem is mapped onto the Ising model by setting J_{ij} to 0 and 1 with equal probability. In both cases, the external field $h_i = 0$. The labels for each line and their references are given in TABLE 1. Error bars on original data where present have been omitted for clarity. CIM1, CIM2, CIM3, DWAV and RBM are benchmarked on the same problem instances. The asterisks denote data reported for theoretical predictions rather than directly measured from a hardware implementation.

Computation performance comparisons

Because the utility of any Ising machine is in its ability to solve a given Ising problem both quickly and accurately, an important task is to benchmark performance and compare competing methods. We direct our attention particularly to various Ising solvers that have been experimentally tested for relatively large systems $N \geq 50$. We consider only large systems as it is difficult to extract any scaling relation for smaller systems. Choosing experimentally realized systems directs our focus onto technologies that are relatively near to maturity. For the figures of merit, we focus on two of the most commonly used quantifiers: the success probability and the time-to-solution. We first define each of these.

The success probability is defined as the probability that the exact ground state of the Ising model is obtained in a single run of the Ising machine. The success probability depends inherently on algorithmic parameters. For annealing methods under ideal conditions, longer annealing times generally result in higher success probabilities, and the success probability can be made arbitrarily close to 1 in the ideal case. However, practical considerations typically prohibit approaching unit success probability. For example, in quantum annealers, maintaining a quantum superposition requires the annealing time to be within the coherence time. In this sense, the success probability still has meaning, because it often involves a trade-off with practical considerations. For our comparison of success probabilities, we generally quote the best-performing value available in the literature.

One of the limitations of the success probability as a figure of merit is that it does not take into account of how long a single run of the Ising machine takes. An Ising machine typically performs multiple runs when attempting to solve a problem, and Ising machines are often optimally operated for a choice of run parameters for which the success probability for a single run is not maximized, but each run takes only a short time. The time-to-solution is another figure of merit that takes into account both the time to perform a single run on a given Ising machine and the success probability. If r runs of a particular scheme are performed, each having a success probability p_{suc} of obtaining the ground state, then the collective probability of getting at least one successful run is $1 - (1 - p_{\mathrm{suc}})^r$. For a given target collective probability, say 99%, the time-to-solution is then related to the success probability as

$$T_{\mathrm{sol}} = \tau \frac{\ln 0.01}{\ln(1 - p_{\mathrm{suc}})} \tag{9}$$

where τ is the time taken for each run. This measure takes into account the different clock speed of various approaches, and allows for various approaches to choose their optimal parameters such that the best performance of the machine can be extracted.

In FIG. 3 we show the performance of various Ising machines, quantified by the success probability for random instances of Sherrington–Kirkpatrick (SK) problems and dense MaxCut problem instances. We note that although the same types of models are used for the comparison in FIGS 3 and 4, the same problem instances were not necessarily used, as we have compiled results from different studies. Although the comparisons are not perfect, we hope that these figures nevertheless give a sense of state of the art of various approaches. In numerous works, the general scaling behaviour is observed to follow the relation

$$p_{\mathrm{suc}} \propto \mathrm{e}^{-bN}, \tag{10}$$

where b is a fitting parameter. Keeping in mind the interpretational caveat mentioned above about how the success probability can for some Ising-machine approaches

be made high at the expense of long runtimes in a way that is ultimately not useful, the results in FIG. 3 suggest that at current technological levels, SA-based approaches, such as the restricted Boltzmann machines (RBMs), implemented on digital hardware give the best scaling with N. One of the reasons for the high success probability of RBMs is the inherent parallelism of this architecture which allows parallel SA updating. However, we note that this figure does not include results from state-of-the-art dynamical systems algorithms such as the CIM with amplitude-heterogeneity correction[118,142] and SB[184]. Based on these algorithms' excellent performance on the G-set MaxCut instances, one may anticipate that they would be competitive with RBM-based solvers. It is notable that the quantum annealer has a particularly poor performance in comparison to other methods. This can be understood[119] as a consequence of the benchmarked D-Wave machine having quantum bit (qubit) connectivity given by a low-degree (Chimera) graph that cannot natively implement either the dense MaxCut or SK models (see TABLE 1). An embedding procedure that requires $\propto N^2$ physical qubits is used to realize the equivalent graph, and this puts the D-Wave annealer at a disadvantage compared with the other listed approaches, which feature all-to-all spin connections. It is for this reason that the success probability for the D-Wave machine has a relation which more resembles $p_{\mathrm{suc}} \propto \mathrm{e}^{-bN^2}$.

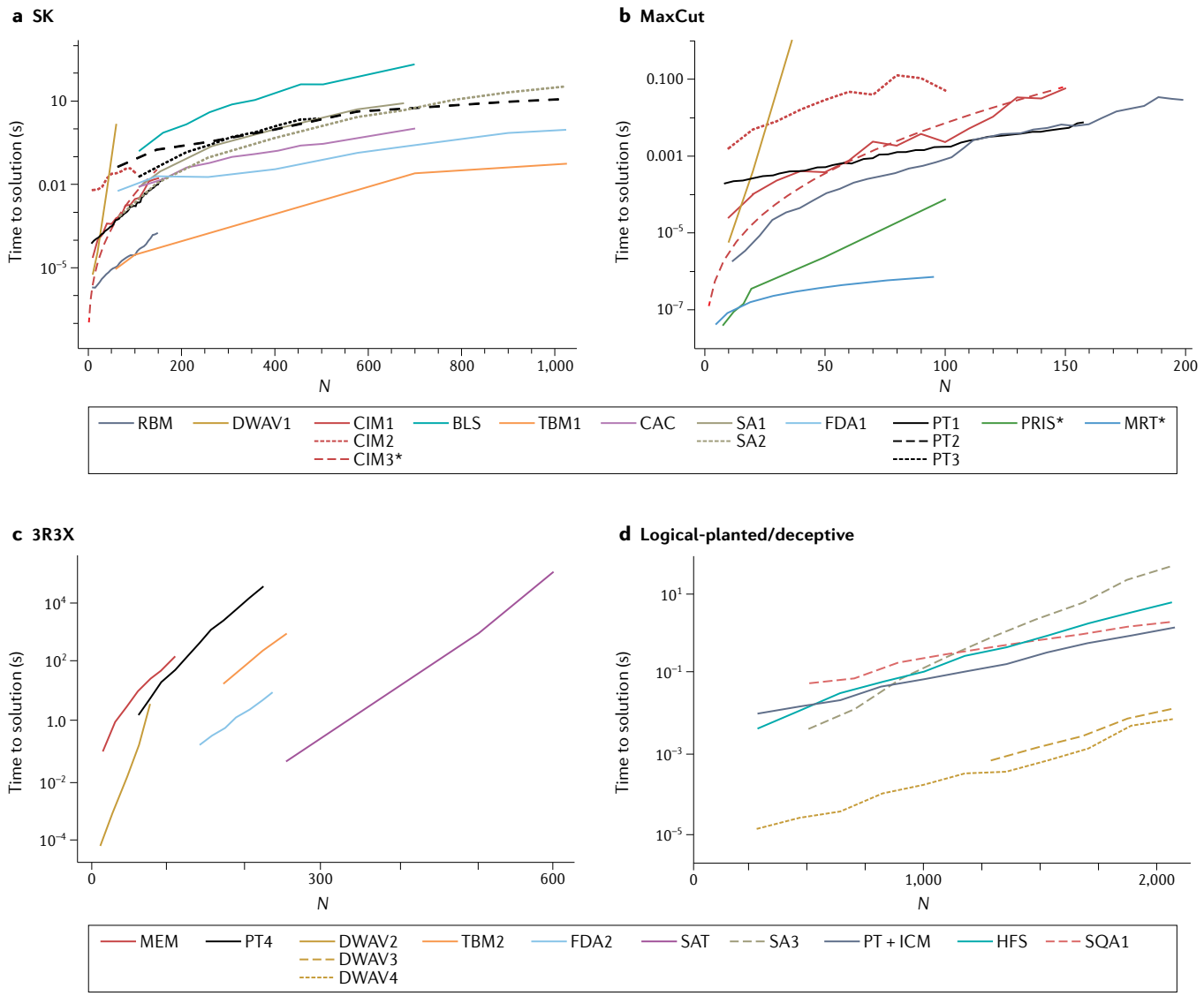

Fig. 4 | **Time-to-solution (TTS) comparison of Ising machines.** The time to obtain a 99% success probability of obtaining the ground state is shown for the Sherington–Kirkpatrick (SK) problem (**a**); dense MaxCut problem (**b**); 3R3X problems (**c**); and logical-planted (dashed lines) and deceptive cluster loops (solid lines) instance classes (**d**). For the SK and MaxCut cases, the Ising model definitions are as in FIG. 3. The 3R3X, logical-planted, and crafted problem definitions can be found in REF.[208], REF.[157] and REF.[211], respectively. The labels for each line and their references are given in TABLE 1. Error bars on original data where present have been omitted for clarity. For both SK and MaxCut, CIM1, CIM2, CIM3, DWAV and RBM are benchmarked on the same problem instances. For each reference, the best time to solution quoted is taken for each N. For results showing multiple annealing times, we have taken results optimized over annealing times. Data reported for theoretical predictions, rather than being directly measured from an hardware implementation, are labeled with *. 3R3X data are from REF.[208]. Logical-planted data are from REF.[157]. Deceptive cluster loops data are from REF.[211].

Table 1 | **Types of Ising machine examined in** FIGS 3 **and** 4

Ising machine/ algorithm	Acronym	Operating principle	Hardware	Hardware connectivity	Parallelization[a]	Benchmark problem	Reference[c]
Breakout local search	BLS	Local search and simulated annealing algorithm	CPU	All-to-all	No	SK	Fig. 3a[142]
Chaotic amplitude control	CAC	Dynamical chaotic algorithm	FPGA	All-to-all	Yes	SK	Fig. 3a[142]
Coherent Ising machine (NTT)	CIM1	Dynamical oscillator	Hybrid (optical/ FPGA)	All-to-all	Yes	MaxCut, SK	Fig. S6[119]
Coherent Ising machine (Stanford)	CIM2	Dynamical oscillator	Hybrid (optical/ FPGA)	All-to-all	Yes	MaxCut, SK	Fig. S6[119]
Coherent Ising machine	CIM3	Dynamical oscillator algorithm	Predicted[b]	All-to-all	Yes	MaxCut, SK	Fig. S10[119]
D-Wave quantum annealer 2Q	DWAV1	Quantum annealer	Superconducting qubits	Chimera	Yes	MaxCut, SK	Fig. 3b,4c[119]
D-Wave quantum annealer Advantage1.1	DWAV2	Quantum annealer	Superconducting qubits	Chimera	Yes	3R3X	Fig. 2[208]
D-Wave quantum annealer 2KQ	DWAV3	Quantum annealer	Superconducting qubits	Chimera	Yes	LP	Fig. 2[157]
D-Wave quantum annealer 2KQ	DWAV4	Quantum annealer	Superconducting qubits	Chimera	Yes	Deceptive	Fig. 1[211]
Fujitsu digital annealer	FDA1	Simulated annealing algorithm	ASIC	All-to-all	Yes	SK	Fig. 7a[58]
Fujitsu digital annealer	FDA2	Simulated annealing algorithm	ASIC	All-to-all	Yes	3R3X	Fig. 2[208]
Hamze–de-Freitas–Selby	HFS	Tree sampling	CPU	All-to-all	No	Deceptive	Fig. 1[211]
Memcomputing	MEM	Dynamical logic gate algorithm	CPU	All-to-all	Yes	3R3X	Fig. 2[208]
Memristor annealing	MRT	Simulated annealing algorithm	Predicted[b]	All-to-all	Yes	MaxCut	Fig. 6a, 6b[49]
Photonic recurrent Ising sampler	PRIS	Oscillator-based annealer	Predicted[b]	All-to-all	Yes	MaxCut	Fig. 2b[48]
Parallel tempering	PT1	Simulated annealing algorithm	CPU	All-to-all	No	MaxCut, SK	Fig. S12[119]
Parallel tempering	PT2	Simulated annealing algorithm	CPU	All-to-all	No	SK	Fig. 7a[58]
Parallel tempering	PT3	Simulated annealing algorithm	CPU	All-to-all	No	SK	Fig. 3a[142]
Parallel tempering	PT4	Simulated annealing algorithm	CPU	All-to-all	No	3R3X	Fig. 2[208]
Isoenergetic cluster moves plus parallel tempering	PT+ICM	Monte Carlo algorithm	Digital-CPU	All-to-all	Yes	Deceptive	Fig. 1[211]
Restricted Boltzmann machine	RBM	Simulated annealing algorithm	FPGA	All-to-all	Yes	MaxCut,SK	Fig. 3,4[64]
Simulated annealing	SA1	Simulated annealing algorithm	CPU	All-to-all	Yes	SK	Fig. 3a[142]
Simulated annealing	SA2	Simulated annealing algorithm	CPU	All-to-all	No	SK	Fig. 7a[58]
Simulated annealing	SA3	Simulated annealing algorithm	GPU	All-to-all	Yes	LP	Fig. 2[157]
SAT on GPU	SAT	SAT algorithm	GPU	All-to-all	Yes	3R3X	Fig. 2[208]
Simulated quantum annealing	SQA1	Quantum Monte Carlo algorithm	GPU	All-to-all	Yes	LP	Fig. 2[157]
Toshiba bifurcation machine	TBM1	Discrete simulated bifurcation algorithm	FPGA	All-to-all	Yes	SK	Fig. 3c[184]
Toshiba bifurcation machine	TBM2	Discrete simulated bifurcation algorithm	GPU	All-to-all	Yes	3R3X	Fig. 2[208]

[a]Parallelization column indicates approaches in which simultaneous updates of Ising spins are performed. [b]If the results of FIG. 3 and 4 are for theoretical predictions, rather than being directly measured from a hardware implementation, the hardware type is quoted as being "Predicted". [c]Figure numbers are those in the cited references. ASIC, application-specific integrated circuit; CPU, central processing unit; FPGA, field-programmable gate array; GPU, graphics processing unit; LP, logical-planted; SK, Sherrington–Kirkpatrick model; 3R3X, 3-regular 3-XORSAT.

FIGURE 4a,b shows the time-to-solution metric for the MaxCut and SK models. The best-performing methods for the SK model use classical digital hardware, for which RBMs and Toshiba bifurcation machines (TBMs) show the lowest time to solution. For the MaxCut problem, RBMs achieve the lowest time-to-solution for a physically implemented machine. We note that the memristor annealing (MRT), photonic recurrent Ising sampler (PRIS) and coherent Ising machine (CIM3) curves involve theoretical prediction of the time-to-solution, rather than a direct measurement of the time. Most of the curves follow the phenomenological scaling relation

$$T_{\mathrm{sol}} \propto \mathrm{e}^{c\sqrt{N}}. \tag{11}$$

where c is a constant. However, if the range of available data is too small, the square-root behaviour may not yet be visible. The D-Wave results are better approximated by an exponential relation $T_{\mathrm{sol}} \propto \mathrm{e}^{cN}$ which requires $\propto N^2$ physical qubits, owing to the limited chimera connectivities of the qubits. One should note that the D-Wave results arise from a hardware implementation limitation that gives a different scaling and not the computational mechanism itself, and may be improved in the future[204–207]. For problem instances with sparse connectivity, the scaling of D-Wave was improved[119]. These results show that the connectivity is an important factor that determines the performance of an Ising machine — TABLE 1 gives the hardware connectivity of different Ising machines.

In FIG. 4c, we show results from REF.[208], which compare Ising machines for 3R3X problems. These problems have a golf-course energy landscape structure with known exact solutions. This class of problem can be solved in polynomial time using Gaussian elimination, but scales exponentially for general solvers such as quantum annealers[209]. The best-performing approach in this case is the SATonGPU approach, which is a highly parallelized version of a SAT algorithm implemented on a GPU. The Fujitsu Digital annealer and Toshiba bifurcation machine achieve similar scaling, but have a larger prefactor than the SATonGPU approach. The memcomputing results are based on classical simulation of a proposed system, hence dedicated hardware might result in some performance improvement[208]. Although there are fewer studies performed for this problem class, these results again suggest that the best-performing solvers today are based on digital computing hardware.

To show the potential of quantum approaches, we also discuss additional problem classes for which it is expected that QA has advantages over a class of classical methods[157–159] despite the above-mentioned limitations of D-Wave. FIGURE 4d compares the optimum time-to-solution for the class of logical-planted (LP) problems that are constructed such that they promote the presence of tunnelling barriers. For these problems, it is expected that barriers can be traversed more effectively by quantum, rather than thermal fluctuations. Here, D-Wave and SQA shows a scaling advantage over SA[157].

The superior performance of SQA implies that tunnelling through barriers may not be considered the exclusive advantage of quantum hardware. However, one should note that SQA cannot be applied for non-stoquastic Hamiltonians that have a sign problem, and as such the power of QA for non-stoquastic Hamiltonians requires further exploration. Non-stoquastic Hamiltonians are important from a computational complexity perspective because adiabatic quantum computation with non-stoquastic Hamiltonians is equivalent to the circuit model of quantum computing[210]. Therefore, they can simulate other universal models with a resource overhead that is, at worst, polynomial. The fact that D-Wave outperforms SA confirms the presence and advantage of quantumness, but the superior performance of SQA suggests that current QA hardware is still dominated by classical dynamics and needs to be improved. We note that for the LP problem class, there are classical algorithms that outperform or have comparable performance with D-Wave[157]. A separate study[211] compared D-Wave with classical heuristic algorithms for another class of specially crafted problem, called the deceptive cluster loop problem (FIG. 4d). For this problem class, D-Wave outperforms the best-known heuristic algorithms, such as parallel tempering Monte Carlo with isoenergetic cluster moves (PT+ICM) and Hamze–de Freitas–Selby (HFS), with approximately two orders of magnitude shorter time to solution. However, no scaling improvement is evident.

Numerous other benchmarking studies of Ising machines have been performed[49,58,64,119,142,169,180,182,184,212]. For example, in REF.[212], the performance of the D-Wave hybrid solver, TBM, FDA and SA was benchmarked for three different classes of problem instances including SK. The results highlight the fact that the performance of machines is problem-dependent. In particular, for the SK model, TBM showed the best performance. In REF.[169], the performance of QAOA was benchmarked on SK and MaxCut problems for problems up to 23 qubits.

Outlook

Comparing the performance of Ising machines, most approaches tend to have similar scalings in terms of the error probability and the time-to-solution metrics as a function of the number of spins, despite the different approaches and technologies used to realize them. The complexity of all approaches scales exponentially with the system size, with the difference being the power within the exponent and the prefactors. This scaling is expected given the NP-complete complexity of the Ising problem — the battle between competing approaches is with respect to the exponents that are achievable. A small difference in the exponent makes a large difference in time-to-solution for large system sizes.

Although FIGS 3 and 4 suggest that classical digital methods are still the best-performing approaches at the time of writing, analogue and quantum computing technologies are rapidly developing, and the technology landscape may undergo a revolution. Some of the best-performing approaches are based on classical digital technology, which have had the benefit of decades

of development and in many cases can be highly parallelized. In comparison, QA approaches have only recently been developed to a scale where they can be tested, either theoretically or experimentally, and often have hardware limitations such as limited connectivity and the presence of decoherence. The specific form of the Ising instance being solved (for instance, the structure of the coupling matrix) can strongly affect the performance. It may be that in the future, much like various numerical algorithms are chosen based on the compatibility of a particular problem, different Ising machines will be used according to their suitability for the given problem. For example, whereas problem instances with small spectral gaps are known to be hard for QA, they may not limit the performance of QAOA[164]. A further step would be to explore hybrid quantum–classical and digital–analogue algorithms to gain the complementary advantage of each[213].

One point of active debate has been the role that quantum mechanics plays in CIMs and in D-Wave's quantum annealers. There exist models of the operation of CIMs that treat them quantum mechanically[113,117], including, for example, a description of the initial state as being in a coherent superposition of all logical states[117]. However, experimental realizations of CIMs thus far[114–116,121,125] have been in regimes of high photon loss, where purely classical models can accurately describe the pertinent dynamics of the systems. The clearest indication of this effective classicality is that similar performance to demonstrated CIMs may be achieved by simulating the mean-field dynamics[119,180,181,214]. However, with sufficiently high nonlinearity to loss in their constituent DOPOs, CIMs can be firmly in the quantum regime[215] and have a strong connection with quantum annealers[127]. Exploring how to construct experimental CIMs in which quantum effects play a crucial role, and designing them so that quantum effects improve the performance of the machine, are two topics of active investigation[120].

For QA, the computational advantage of incoherent tunnelling over certain classical methods (typically SA[35] and the Hamze–de Freitas–Selby algorithm[216,217]) for certain classes of problems has been shown[145,156–159,211]. However, for any real-world problem of interest, no evidence has been found of an unqualified quantum speed-up (as defined in REF.[218]). Perhaps the most compelling results with the D-Wave QA so far are for a specially crafted problem class, known as deceptive cluster loops, for which the QA was found to outperform in terms of time-to-solution for all classical heuristics that were tested, including parallel tempering[34]. The speed-up was of an approximately constant-factor nature, with no strong evidence of a scaling advantage[211].

Another disadvantage of QA algorithms is that they cannot sample uniformly all low-lying states, unlike other heuristic algorithms such as SA-based algorithms[219,220]. In SA, after many repetitions and starting from different initial states, one can record all the configurations that minimize the problem Hamiltonian. For an optimization machine, such an ability to sample fairly is beneficial because having different solutions for a problem is often useful. Furthermore, it is not yet well understood what the role of entanglement in QA is and whether it contributes to a quantum speed-up[221]. Although various aspects of quantumness, including entanglement, may or may not aid the performance of CIMs or QAs, this uncertainty has inspired the proposal of interesting quantum-inspired classical algorithms related to CIMs and QAs, which is a fruitful development in its own right.

Looking to the future, there is much room for development for Ising machines. For many classical algorithms, the time-to-solution and other metrics are known to several decimal places; the scaling of Ising machines should likewise be quantified with precision so that competing methods can be compared. FIGURES 3 and 4 are extremely preliminary in this regard. With improved quantification and investigation for different classes of problems, a better understanding of the suitability of various approaches for a particular problem can be known in advance. Another interesting direction is to compare the performance of the different Ising machines for finding an approximate solution with different levels of accuracy, since for many applications finding a high-quality solution, rather than the exact solution, is sufficient[222]. Several forms of Ising machine that either rely on or can be enhanced by quantum-mechanical mechanisms have been proposed and demonstrated. However, constructing large-scale quantum machines with high connectivity and low decoherence remains an outstanding challenge for the field of quantum information processing in general, and further progress in this direction is needed for experimental exploration of the benefits that quantum-mechanical methods may bring to solving Ising problems. This is in contrast with classical approaches, especially digital ones, which often have little difficulty in supporting full connectivity. Given the demand for faster methods of solving optimization problems in society, and the maturity of conventional algorithms and processors, it seems likely that the development of specialized Ising machines will continue well into the future, featuring an exciting interplay between hardware engineering, computer science, statistical physics and quantum mechanics.

Published online 4 May 2022

1. Lucas, A. Ising formulations of many NP problems. *Front. Phys.* **2**, 5 (2014).
2. Tanahashi, K., Takayanagi, S., Motohashi, T. & Tanaka, S. Application of Ising machines and a software development for Ising machines. *J. Phys. Soc. Japan* **88**, 061010 (2019).
3. Smelyanskiy, V. N. et al. A near-term quantum computing approach for hard computational problems in space exploration. Preprint at https://arxiv.org/abs/1204.2821 (2012).
4. Hauke, P., Katzgraber, H. G., Lechner, W., Nishimori, H. & Oliver, W. D. Perspectives of quantum annealing: methods and implementations. *Rep. Prog. Phys.* **83**, 054401 (2020).
5. Karp, R. M. in *Complexity of Computer Computations*, 85–103 (Springer, 1972).
6. Mézard, M., Parisi, G. & Virasoro, M. A. *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications* vol. 9 (World Scientific, 1987).
7. Barahona, F., Grötschel, M., Jünger, M. & Reinelt, G. An application of combinatorial optimization to statistical physics and circuit layout design. *Oper. Res.* **36**, 493–513 (1988).
8. Chang, K. & Du, D. C. Efficient algorithms for layer assignment problem. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **6**, 67–78 (1987).
9. Wang, J., Jebara, T. & Chang, S.-F. Semi-supervised learning using greedy Max-cut. *J. Mach. Learn. Res.* **14**, 771–800 (2013).
10. Collins, T. *Graph Cut Matching in Computer Vision* (Univ. Edinburgh, 2004).
11. Arora, C., Banerjee, S., Kalra, P. & Maheshwari, S. An efficient graph cut algorithm for computer vision problems. In *European Conf. Computer Vision*, 552–565 (Springer, 2010).

12. Turing, A. M. On computable numbers, with an application to the entscheidungsproblem. *Proc. Lond. Math. Soc.* **2**, 230–265 (1937).
13. Bournez, O. & Pouly, A. in *Handbook of Computability and Complexity in Analysis 2018* (eds Brattka, V. & Hertling, P.) 173–226 (Springer, 2018).
14. Feynman, R. P. Simulating physics with computers. *Int. J. Theor. Phys.* **21**, 467–488 (1982).
15. Buluta, I. & Nori, F. Quantum simulators. *Science* **326**, 108–111 (2009).
16. Georgescu, I. M., Ashhab, S. & Nori, F. Quantum simulation. *Rev. Mod. Phys.* **86**, 153 (2014).
17. Lloyd, S. Universal quantum simulators. *Science* **273**, 1073–1078 (1996).
18. Greiner, M., Mandel, O., Esslinger, T., Hänsch, T. W. & Bloch, I. Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms. *Nature* **415**, 39–44 (2002).
19. Labuhn, H. et al. Tunable two-dimensional arrays of single Rydberg atoms for realizing quantum Ising models. *Nature* **534**, 667–670 (2016).
20. Bernien, H. et al. Probing many-body dynamics on a 51-atom quantum simulator. *Nature* **551**, 579–584 (2017).
21. Keesling, A. et al. Quantum Kibble–Zurek mechanism and critical dynamics on a programmable Rydberg simulator. *Nature* **568**, 207–211 (2019).
22. Scholl, P. et al. Quantum simulation of 2D antiferromagnets with hundreds of Rydberg atoms. *Nature* **595**, 233–238 (2021).
23. Farhi, E. et al. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science* **292**, 472–475 (2001).
24. Byrnes, T., Yan, K. & Yamamoto, Y. Accelerated optimization problem search using Bose–Einstein condensation. *New J. Phys.* **13**, 113025 (2011).
25. King, A. D. et al. Observation of topological phenomena in a programmable lattice of 1,800 qubits. *Nature* **560**, 456–460 (2018).
26. Harris, R. et al. Phase transitions in a programmable quantum spin glass simulator. *Science* **361**, 162–165 (2018).
27. Vadlamani, S. K., Xiao, T. P. & Yablonovitch, E. Physics successfully implements Lagrange multiplier optimization. *Proc. Natl Acad. Sci. USA* **117**, 26639–26650 (2020).
28. Geman, S. & Geman, D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 721–741 (1984).
29. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087–1092 (1953).
30. Hastings, W. K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97–109 (1970).
31. Swendsen, R. H. & Wang, J.-S. Replica Monte Carlo simulation of spin-glasses. *Phys. Rev. Lett.* **57**, 2607–2609 (1986).
32. Earl, D. J. & Deem, M. W. Parallel tempering: theory, applications, and new perspectives. *Phys. Chem. Chem. Phys.* **7**, 3910–3916 (2005).
33. Wang, W., Machta, J. & Katzgraber, H. G. Population annealing: theory and application in spin glasses. *Phys. Rev. E* **92**, 063307 (2015).
34. Zhu, Z., Ochoa, A. J. & Katzgraber, H. G. Efficient cluster algorithm for spin glasses in any space dimension. *Phys. Rev. Lett.* **115**, 077201 (2015).
35. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science* **220**, 671–680 (1983).
36. Černý, V. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *J. Optim. Theory Appl.* **45**, 41–51 (1985).
37. Camsari, K. Y., Faria, R., Sutton, B. M. & Datta, S. Stochastic p-bits for invertible logic. *Phys. Rev. X* **7**, 031014 (2017).
38. Camsari, K. Y., Sutton, B. M. & Datta, S. p-bits for probabilistic spin logic. *Appl. Phys. Rev.* **6**, 011305 (2019).
39. Borders, W. A. et al. Integer factorization using stochastic magnetic tunnel junctions. *Nature* **573**, 390–393 (2019).
40. Sutton, B., Camsari, K. Y., Behin-Aein, B. & Datta, S. Intrinsic optimization using stochastic nanomagnets. *Sci. Rep.* **7**, 1–9 (2017).
41. Shim, Y., Jaiswal, A. & Roy, K. Ising computation based combinatorial optimization using spin-Hall effect induced stochastic magnetization reversal. *J. Appl. Phys.* **121**, 193902 (2017).
42. Arnalds, U. B. et al. A new look on the two-dimensional Ising model: thermal artificial spins. *New J. Phys.* **18**, 023008 (2016).
43. Bhanja, S., Karunaratne, D., Panchumarthy, R., Rajaram, S. & Sarkar, S. Non-Boolean computing with nanomagnets for computer vision applications. *Nature Nanotechnol.* **11**, 177–183 (2016).
44. Lee, A. et al. A thermodynamic core using voltage-controlled spin-orbit-torque magnetic tunnel junctions. *Nanotechnology* https://doi.org/10.1088/1361-6528/abeb9b (2021).
45. Mizushima, K., Goto, H. & Sato, R. Large-scale Ising-machines composed of magnetic neurons. *Appl. Phys. Lett.* **111**, 172406 (2017).
46. Pierangeli, D., Marcucci, G. & Conti, C. Large-scale photonic Ising machine by spatial light modulation. *Phys. Rev. Lett.* **122**, 213902 (2019).
47. Pierangeli, D., Marcucci, G. & Conti, C. Adiabatic evolution on a spatial-photonic Ising machine. *Optica* **7**, 1535–1543 (2020).
48. Roques-Carmes, C. et al. Heuristic recurrent algorithms for photonic Ising machines. *Nat. Commun.* **11**, 1–8 (2020).
49. Cai, F. et al. Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks. *Nat. Electron.* **3**, 409–418 (2020).
50. Bojnordi, M. N. & Ipek, E. Memristive Boltzmann machine: a hardware accelerator for combinatorial optimization and deep learning. In *2016 IEEE Int. Symp. High-Performance Computer Architecture (HPCA)*, https://doi.org/10.1109/HPCA.2016.7446049 (IEEE, 2016).
51. Behin-Aein, B., Diep, V. & Datta, S. A building block for hardware belief networks. *Sci. Rep.* **6**, 29893 (2016).
52. Sarkar, S. & Bhanja, S. Synthesizing energy minimizing quantum-dot cellular automata circuits for vision computing. In *5th IEEE Conf. Nanotechnol. 2005*, 541–544 (IEEE, 2005).
53. Kiraly, B., Knol, E. J., van Weerdenburg, W. M., Kappen, H. J. & Khajetoorians, A. A. An atomic Boltzmann machine capable of self-adaption. *Nat. Nanotechnol.* **16**, 414–420 (2021).
54. Guo, S. Y. et al. A molecular computing approach to solving optimization problems via programmable microdroplet arrays. *Matter* **4**, 1107–1124 (2021).
55. Byrnes, T., Koyama, S., Yan, K. & Yamamoto, Y. Neural networks using two-component Bose–Einstein condensates. *Sci. Rep.* **3**, 2531 (2013).
56. Yamaoka, M. et al. A 20k-spin Ising chip to solve combinatorial optimization problems with CMOS annealing. *IEEE J. Solid-State Circuits* **51**, 303–309 (2015).
57. Matsubara, S. et al. Digital annealer for high-speed solving of combinatorial optimization problems and its applications. In *2020 25th Asia and South Pacific Design Automation Conf. (ASP-DAC)*, 667–672 (IEEE, 2020).
58. Aramon, M. et al. Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. *Front. Phys.* **7**, 48 (2019).
59. Su, Y., Kim, H. & Kim, B. Cim-spin: A 0.5-to-1.2V scalable annealing processor using digital compute-in-memory spin operators and register-based spins for combinatorial optimization problems. In *2020 IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 480–482 (IEEE, 2020).
60. Yamamoto, K. et al. Statica: a 512-spin 0.25 m-weight full-digital annealing processor with a near-memory all-spin-updates-at-once architecture for combinatorial optimization with complete spin–spin interactions. In *2020 IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 138–140 (IEEE, 2020).
61. Tsukamoto, S., Takatsu, M., Matsubara, S. & Tamura, H. An accelerator architecture for combinatorial optimization problems. *Fujitsu Sci. Tech. J.* **53**, 8–13 (2017).
62. Matsubara, S. et al. Ising-model optimizer with parallel-trial bit-sieve engine. In *Conf. Complex, Intelligent, and Software Intensive Systems*, 432–438 (Springer, 2017).
63. Yamamoto, K. et al. A time-division multiplexing Ising machine on FPGAs. In *Proc. 8th Int. Symp. Highly Efficient Accelerators and Reconfigurable Technologies (HEART)*, https://doi.org/10.1145/3120895.3120905 (2017).
64. Patel, S., Chen, L., Canoza, P. & Salahuddin, S. Ising model optimization problems on a FPGA accelerated restricted Boltzmann machine. Preprint at https://arxiv.org/abs/2008.04436 (2020).
65. Aadit, N. A. et al. Massively parallel probabilistic computing with sparse Ising machines. Preprint at https://arxiv.org/abs/2110.02481 (2021).
66. Reuther, A. et al. Survey and benchmarking of machine learning accelerators. In *2019 IEEE High Perform. Extreme Comput. Conf. (HPEC)*, 1–9 (IEEE, 2019).
67. Arima, Y. et al. A 336-neuron, 28 K-synapse, self-learning neural network chip with branch-neuron-unit architecture. *IEEE J. Solid-state Circuits* **26**, 1637–1644 (1991).
68. Alspector, J., Allen, R. B., Jayakumar, A., Zeppenfeld, T. & Meir, R. Relaxation networks for large supervised learning problems. In *Adv. Neural Inf. Process. Syst.*, 1015–1021 (Citeseer, 1991).
69. Merolla, P. A. et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **345**, 668–673 (2014).
70. Skubiszewski, M. An exact hardware implementation of the Boltzmann machine. In *Proc. 4th IEEE Symp. Parallel and Distributed Processing*, 107–110 (IEEE, 1992).
71. Zhu, J. & Sutton, P. FPGA implementations of neural networks — a survey of a decade of progress. In *Int. Conf. Field Programmable Logic and Applications*, 1062–1066 (Springer, 2003).
72. Kim, S. K., McAfee, L. C., McMahon, P. L. & Olukotun, K. A highly scalable restricted Boltzmann machine FPGA implementation. In *2009 Int. Conf. Field Programmable Logic and Applications*, 367–372 (IEEE, 2009).
73. Kim, S. K., McMahon, P. L. & Olukotun, K. A large-scale architecture for restricted Boltzmann machines. In *18th IEEE Annu. Int. Symp. Field-Programmable Custom Computing Machines*, 201–208 (IEEE, 2010).
74. Le Ly, D. & Chow, P. High-performance reconfigurable hardware architecture for restricted Boltzmann machines. *IEEE Trans. Neural Networks* **21**, 1780–1792 (2010).
75. Kim, L.-W., Asaad, S. & Linsker, R. A fully pipelined FPGA architecture of a factored restricted boltzmann machine artificial neural network. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)* **7**, 1–23 (2014).
76. Ly, D. L., Paprotski, V. & Yen, D. *Neural Networks on GPUs: Restricted Boltzmann Machines* (2008); https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.431.8720&rep=rep1&type=pdf
77. Zhu, Y., Zhang, Y. & Pan, Y. Large-scale restricted Boltzmann machines on single GPU. In *2013 IEEE Int. Conf. Big Data*, 169–174 (IEEE, 2013).
78. Okuyama, T., Sonobe, T., Kawarabayashi, K.-i & Yamaoka, M. Binary optimization by momentum annealing. *Phys. Rev. E* **100**, 012111 (2019).
79. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl Acad. Sci. USA* **79**, 2554–2558 (1982).
80. Hopfield, J. J. & Tank, D. W. 'Neural' computation of decisions in optimization problems. *Biol. Cybern.* **52**, 141–152 (1985).
81. Von Neumann, J. Non-linear capacitance or inductance switching, amplifying, and memory organs. US Patent 2,815,488 (1957).
82. Wigington, R. A new concept in computing. *Proc. IRE* **47**, 516–523 (1959).
83. Goto, E. The parametron, a digital computing element which utilizes parametric oscillation. *Proc. IRE* **47**, 1304–1316 (1959).
84. Csaba, G. & Porod, W. Coupled oscillators for computing: a review and perspective. *Appl. Phys. Rev.* **7**, 011302 (2020).
85. Raychowdhury, A. et al. Computing with networks of oscillatory dynamical systems. *Proc. IEEE* **107**, 73–89 (2018).
86. Kuramoto, Y. in *International Symposium on Mathematical Problems in Theoretical Physics*, 420 (Lecture Notes in Physics vol. 30, Springer, 1975).
87. Acebrón, J. A., Bonilla, L. L., Vicente, C. J. P., Ritort, F. & Spigler, R. The Kuramoto model: a simple paradigm for synchronization phenomena. *Rev. Mod. Phys.* **77**, 137 (2005).
88. Breakspear, M., Heitmann, S. & Daffertshofer, A. Generative models of cortical oscillations: neurobiological implications of the Kuramoto model. *Front. Human Neurosci.* **4**, 190 (2010).
89. Wu, C. W. & Chua, L. O. Application of graph theory to the synchronization in an array of coupled nonlinear oscillators. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **42**, 494–497 (1995).
90. Wu, C. W. Graph coloring via synchronization of coupled oscillators. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **45**, 974–978 (1998).
91. Wu, J., Jiao, L., Li, R. & Chen, W. Clustering dynamics of nonlinear oscillator network: application to graph

coloring problem. *Physica D* **240**, 1972–1978 (2011).
92. Kalinin, K. P. & Berloff, N. G. Global optimization of spin Hamiltonians with gain-dissipative systems. *Sci. Rep.* **8**, 1–9 (2018).
93. Wang, T. & Roychowdhury, J. OIM: oscillator-based Ising machines for solving combinatorial optimisation problems. In *Int. Conf. Unconventional Computation and Natural Computation*, 232–256 (Springer, 2019).
94. Afoakwa, R., Zhang, Y., Vengalam, U. K. R., Ignjatovic, Z. & Huang, M. BRIM: bistable resistively-coupled Ising machine. In *2021 IEEE Int. Symp. High-Performance Computer Architecture (HPCA)*, 749–760 (IEEE, 2021).
95. McGoldrick, B. C., Sun, J. Z. & Liu, L. Ising machine based on electrically coupled spin Hall nano-oscillators. *Phys. Rev. Appl.* **17**, 014006 (2021).
96. Albertsson, D. I. et al. Ultrafast Ising machines using spin torque nano-oscillators. *Appl. Phys. Lett.* **118**, 112404 (2021).
97. Chou, J., Bramhavar, S., Ghosh, S. & Herzog, W. Analog coupled oscillator based weighted Ising machine. *Sci. Rep.* **9**, 14786 (2019).
98. Xiao, T. P. *Optoelectronics for Refrigeration and Analog Circuits for Combinatorial Optimization*. PhD thesis, Univ. California Berkeley (2019).
99. Saito, K., Aono, M. & Kasai, S. Amoeba-inspired analog electronic computing system integrating resistance crossbar for solving the travelling salesman problem. *Sci. Rep.* **10**, 20772 (2020).
100. Shukla, N. et al. Synchronized charge oscillations in correlated electron systems. *Sci. Rep.* **4**, 4964 (2014).
101. Parihar, A., Shukla, N., Jerry, M., Datta, S. & Raychowdhury, A. Vertex coloring of graphs via phase dynamics of coupled oscillatory networks. *Sci. Rep.* **7**, 911 (2017).
102. Dutta, S. et al. An Ising Hamiltonian solver using stochastic phase-transition nano-oscillators. Preprint at https://arxiv.org/abs/2007.12331 (2020).
103. Zahedinejad, M. et al. Two-dimensional mutually synchronized spin Hall nano-oscillator arrays for neuromorphic computing. *Nat. Nanotechnol.* **15**, 47–52 (2020).
104. Houshang, A. et al. A spin Hall Ising machine. Preprint at https://arxiv.org/abs/2006.02236 (2020).
105. Mallick, A. et al. Using synchronized oscillators to compute the maximum independent set. *Nat. Commun.* **11**, 4689 (2020).
106. Bashar, M. K. et al. Experimental demonstration of a reconfigurable coupled oscillator platform to solve the Max-Cut problem. *IEEE J. Exploratory Solid-State Computational Devices and Circuits* **6**, 116–121 (2020).
107. Ahmed, I., Chiu, P.-W., Moy, W. & Kim, C. H. A probabilistic compute fabric based on coupled ring oscillators for solving combinatorial optimization problems. *IEEE J. Solid-State Circuits* **56**, 2870–2880 (2021).
108. Traversa, F. L. & Di Ventra, M. Universal memcomputing machines. *IEEE Trans. Neural Netw. Learn. Syst.* **26**, 2702–2715 (2015).
109. Di Ventra, M. & Traversa, F. L. Perspective: Memcomputing: leveraging memory and physics to compute efficiently. *J. Appl. Phys.* **123**, 180901 (2018).
110. Sheldon, F., Traversa, F. L. & Di Ventra, M. Taming a nonconvex landscape with dynamical long-range order: memcomputing Ising benchmarks. *Phys. Rev. E* **100**, 053311 (2019).
111. Aiken, J. & Traversa, F. L. Memcomputing for accelerated optimization. Preprint at https://arxiv.org/abs/2003.10644 (2020).
112. Utsunomiya, S., Takata, K. & Yamamoto, Y. Mapping of Ising models onto injection-locked laser systems. *Opt. Express* **19**, 18091–18108 (2011).
113. Wang, Z., Marandi, A., Wen, K., Byer, R. L. & Yamamoto, Y. Coherent Ising machine based on degenerate optical parametric oscillators. *Phys. Rev. A* **88**, 063853 (2013).
114. Marandi, A., Wang, Z., Takata, K., Byer, R. L. & Yamamoto, Y. Network of time-multiplexed optical parametric oscillators as a coherent Ising machine. *Nat. Photonics* **8**, 937–942 (2014).
115. McMahon, P. L. et al. A fully programmable 100-spin coherent Ising machine with all-to-all connections. *Science* **354**, 614–617 (2016).
116. Inagaki, T. et al. A coherent Ising machine for 2000-node optimization problems. *Science* **354**, 603–606 (2016).
117. Yamamoto, Y. et al. Coherent Ising machines — optical neural networks operating at the quantum limit. *npj Quantum Inf.* **3**, 49 (2017).
118. Leleu, T., Yamamoto, Y., McMahon, P. L. & Aihara, K. Destabilization of local minima in analog spin systems by correction of amplitude heterogeneity. *Phys. Rev. Letters* **122**, 040607 (2019).
119. Hamerly, R. et al. Experimental investigation of performance differences between coherent Ising machines and a quantum annealer. *Sci. Adv.* **5**, eaau0823 (2019).
120. Yamamoto, Y., Leleu, T., Ganguli, S. & Mabuchi, H. Coherent Ising machines — quantum optics and neural network perspectives. *Appl. Phys. Lett.* **117**, 160501 (2020).
121. Honjo, T. et al. 100,000-spin coherent Ising machine. *Science Advances* **7**, eabh0952 (2021).
122. Marandi, A., Leindecker, N. C., Vodopyanov, K. L. & Byer, R. L. All-optical quantum random bit generation from intrinsically binary phase of parametric oscillators. *Opt. Express* **20**, 19322–19330 (2012).
123. Kalinin, K. P. & Berloff, N. G. Networks of non-equilibrium condensates for global optimization. *New J. Phys.* **20**, 113023 (2018).
124. Takata, K. et al. A 16-bit coherent Ising machine for one-dimensional ring and cubic graph problems. *Sci. Rep.* **6**, 34089 (2016).
125. Inagaki, T. et al. Large-scale Ising spin network based on degenerate optical parametric oscillators. *Nat. Photonics* **10**, 415–419 (2016).
126. Okawachi, Y. et al. Demonstration of chip-based coupled degenerate optical parametric oscillators for realizing a nanophotonic spin-glass. *Nat. Commun.* **11**, 4119 (2020).
127. Goto, H. Bifurcation-based adiabatic quantum computation with a nonlinear oscillator network. *Sci. Rep.* **6**, 21686 (2016).
128. Tamate, S., Yamamoto, Y., Marandi, A., McMahon, P. & Utsunomiya, S. Simulating the classical XY model with a laser network. Preprint at https://arxiv.org/abs/1608.00358 (2016).
129. Babaeian, M. et al. A single shot coherent Ising machine based on a network of injection-locked multicore fiber lasers. *Nat. Commun.* **10**, 3516 (2019).
130. Parto, M., Hayenga, W., Marandi, A., Christodoulides, D. N. & Khajavikhan, M. Realizing spin Hamiltonians in nanoscale active photonic lattices. *Nat. Mater.* **19**, 725–731 (2020).
131. Nixon, M., Ronen, E., Friesem, A. A. & Davidson, N. Observing geometric frustration with thousands of coupled lasers. *Phys. Rev. Lett.* **110**, 184102 (2013).
132. Böhm, F., Verschaffelt, G. & Van der Sande, G. A poor man's coherent Ising machine based on opto-electronic feedback systems for solving optimization problems. *Nat. Commun.* **10**, 3538 (2019).
133. Lagoudakis, P. G. & Berloff, N. G. A polariton graph simulator. *New J. Phys.* **19**, 125008 (2017).
134. Berloff, N. G. et al. Realizing the classical *XY* Hamiltonian in polariton simulators. *Nat. Mater.* **16**, 1120–1126 (2017).
135. Kalinin, K. P. & Berloff, N. G. Simulating Ising and *n*-state planar Potts models and external fields with nonequilibrium condensates. *Phys. Rev. Lett.* **121**, 235302 (2018).
136. Kyriienko, O., Sigurdsson, H. & Liew, T. C. H. Probabilistic solving of NP-hard problems with bistable nonlinear optical networks. *Phys. Rev. B* **99**, 195301 (2019).
137. Mahboob, I., Okamoto, H. & Yamaguchi, H. An electromechanical Ising Hamiltonian. *Sci. Adv.* **2**, e1600236 (2016).
138. Tezak, N. et al. Integrated coherent Ising machines based on self-phase modulation in microring resonators. *IEEE J. Sel. Top. Quant. Electron.* **26**, 1–15 (2019).
139. Bernaschi, M., Billoire, A., Maiorano, A., Parisi, G. & Ricci-Tersenghi, F. Strong ergodicity breaking in aging of mean-field spin glasses. *Proc. Natl Acad. Sci. USA* **117**, 17522–17527 (2020).
140. Ercsey-Ravasz, M. & Toroczkai, Z. Optimization hardness as transient chaos in an analog approach to constraint satisfaction. *Nat. Phys.* **7**, 966–970 (2011).
141. Molnár, B., Molnár, F., Varga, M., Toroczkai, Z. & Ercsey-Ravasz, M. A continuous-time MaxSAT solver with high analog performance. *Nat. Commun.* **9**, 4864 (2018).
142. Leleu, T. et al. Chaotic amplitude control for neuromorphic Ising machine in silico. Preprint at https://arxiv.org/abs/2009.04084 (2020).
143. Yin, X. et al. Efficient analog circuits for Boolean satisfiability. *IEEE Trans. Very Large Scale Integration (VLSI) Systems* **26**, 155–167 (2017).
144. Elser, V., Rankenburg, I. & Thibault, P. Searching with iterated maps. *Proc. Natl Acad. Sci. USA* **104**, 418–423 (2007).
145. Albash, T. & Lidar, D. A. Adiabatic quantum computation. *Rev. Mod. Phys.* **90**, 015002 (2018).
146. Das, A. & Chakrabarti, B. K. Colloquium: Quantum annealing and analog quantum computation. *Rev. Mod. Phys.* **80**, 1061–1081 (2008).
147. Crosson, E. J. & Lidar, D. A. Prospects for quantum enhancement with diabatic quantum annealing. *Nat. Rev. Phys.* **3**, 466 (2021).
148. Apolloni, B., Carvalho, C. & De Falco, D. Quantum stochastic optimization. *Stoch. Process. Their Appl.* **33**, 233–244 (1989).
149. Kadowaki, T. & Nishimori, H. Quantum annealing in the transverse Ising model. *Phys. Rev. E* **58**, 5355–5363 (1998).
150. Farhi, E., Goldstone, J., Gutmann, S. & Sipser, M. Quantum computation by adiabatic evolution. Preprint at https://arxiv.org/abs/quant-ph/0001106 (2000).
151. Roland, J. & Cerf, N. J. Quantum search by local adiabatic evolution. *Phys. Rev. A* **65**, 042308 (2002).
152. Amin, M. Effect of local minima on adiabatic quantum optimization. *Phys. Rev. Lett.* **100**, 130503 (2008).
153. Schaller, G., Mostame, S. & Schützhold, R. General error estimate for adiabatic quantum computing. *Phys. Rev. A* **73**, 062307 (2006).
154. Lidar, D. A., Rezakhani, A. T. & Hamma, A. Adiabatic approximation with exponential accuracy for many-body systems and quantum computation. *J. Math Phys.* **50**, 102106 (2009).
155. Katzgraber, H. G., Hamze, F., Zhu, Z., Ochoa, A. J. & Munoz-Bauza, H. Seeking quantum speedup through spin glasses: the good, the bad, and the ugly. *Phys. Rev. X* **5**, 031026 (2015).
156. Muthukrishnan, S., Albash, T. & Lidar, D. A. Tunneling and speedup in quantum optimization for permutation-symmetric problems. *Phys. Rev. X* **6**, 031010 (2016).
157. Albash, T. & Lidar, D. A. Demonstration of a scaling advantage for a quantum annealer over simulated annealing. *Phys. Rev. X* **8**, 031016 (2018).
158. Denchev, V. S. et al. What is the computational value of finite-range tunneling? *Phys. Rev. X* **6**, 031015 (2016).
159. Boixo, S. et al. Computational multiqubit tunnelling in programmable quantum annealers. *Nat. Commun.* **7**, 10327 (2016).
160. Cerezo, M. et al. Variational quantum algorithms. *Nat. Rev. Phys.* **3**, 625–644 (2021).
161. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
162. Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. Preprint at https://arxiv.org/abs/1411.4028 (2014).
163. Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem. Preprint at https://arxiv.org/abs/1412.6062 (2014).
164. Zhou, L., Wang, S.-T., Choi, S., Pichler, H. & Lukin, M. D. Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices. *Phys. Rev. X* **10**, 021067 (2020).
165. Guerreschi, G. G. & Smelyanskiy, M. Practical optimization for hybrid quantum-classical algorithms. Preprint at https://arxiv.org/abs/1701.01450 (2017).
166. Khairy, S., Shaydulin, R., Cincio, L., Alexeev, Y. & Balaprakash, P. Learning to optimize variational quantum circuits to solve combinatorial problems. In *Proc. AAAI Conf. Artificial Intelligence* vol. 34, 2367–2375 (2020).
167. Pagano, G. et al. Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator. *Proc. Natl Acad. Sci. USA* **117**, 25396–25401 (2020).
168. Farhi, E. & Harrow, A. W. Quantum supremacy through the quantum approximate optimization algorithm. Preprint at https://arxiv.org/abs/1602.07674 (2016).
169. Harrigan, M. P. et al. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nat. Phys.* **17**, 332–336 (2021).
170. Qiang, X. et al. Large-scale silicon quantum photonics implementing arbitrary two-qubit processing. *Nat. Photonics* **12**, 534–539 (2018).
171. Ozaeta, A., van Dam, W. & McMahon, P. L. Expectation values from the single-layer quantum approximate optimization algorithm on Ising problems. Preprint at https://arxiv.org/abs/2012.03421 (2020).
172. Sanders, Y. R. et al. Compilation of fault-tolerant quantum heuristics for combinatorial optimization. *PRX Quantum* **1**, 020312 (2020).

173. Somma, R. D., Boixo, S., Barnum, H. & Knill, E. Quantum simulations of classical annealing processes. *Phys. Rev. Lett.* **101**, 130504 (2008).
174. Boixo, S., Ortiz, G. & Somma, R. Fast quantum methods for optimization. *Eur. Phys. J. Spec. Top.* **224**, 35–49 (2015).
175. Lemieux, J., Heim, B., Poulin, D., Svore, K. & Troyer, M. Efficient quantum walk circuits for Metropolis–Hastings algorithm. *Quantum* **4**, 287 (2020).
176. Bapst, V. & Semerjian, G. Thermal, quantum and simulated quantum annealing: analytical comparisons for simple models. *J. Phys. Conf. Ser.* **473**, 012011 (2013).
177. Das, A. & Chakrabarti, B. K. *Quantum Annealing and Related Optimization Methods* vol. 679 (Springer Science & Business Media, 2005).
178. Crosson, E. & Harrow, A. W. Simulated quantum annealing can be exponentially faster than classical simulated annealing. In *2016 IEEE 57th Annual Symp. Foundations of Computer Science (FOCS)*, 714–723 (IEEE, 2016).
179. Andriyash, E. & Amin, M. H. Can quantum Monte Carlo simulate quantum annealing? Preprint at https://arxiv.org/abs/1703.09277 (2017).
180. King, A. D., Bernoudy, W., King, J., Berkley, A. J. & Lanting, T. Emulating the coherent Ising machine with a mean-field algorithm. Preprint at https://arxiv.org/abs/1806.08422 (2018).
181. Tiunov, E. S., Ulanov, A. E. & Lvovsky, A. Annealing by simulating the coherent Ising machine. *Opt. Express* **27**, 10288–10295 (2019).
182. Goto, H., Tatsumura, K. & Dixon, A. R. Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems. *Sci. Adv.* **5**, eaav2372 (2019).
183. Tatsumura, K., Dixon, A. R. & Goto, H. FPGA-based simulated bifurcation machine. In *2019 29th Int. Conf. Field Programmable Logic and Applications (FPL)*, 59–66 (IEEE, 2019).
184. Goto, H. et al. High-performance combinatorial optimization based on classical mechanics. *Sci. Adv.* **7**, eabe7953 (2021).
185. Tatsumura, K., Yamasaki, M. & Goto, H. Scaling out Ising machines using a multi-chip architecture for simulated bifurcation. *Nat. Electronics* **4**, 208–217 (2021).
186. Orús, R. Tensor networks for complex quantum systems. *Nat. Rev. Phys.* **1**, 538–550 (2019).
187. Alcazar, J. & Perdomo-Ortiz, A. Enhancing combinatorial optimization with quantum generative models. Preprint at https://arxiv.org/abs/2101.06250 (2021).
188. Mugel, S. et al. Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks. Preprint at https://arxiv.org/abs/2007.00017 (2020).
189. Mohseni, N., Navarrete-Benlloch, C., Byrnes, T. & Marquardt, F. Deep recurrent networks predicting the gap evolution in adiabatic quantum computing. Preprint at https://arxiv.org/abs/2109.08492 (2021).
190. Bojesen, T. A. Policy-guided Monte Carlo: reinforcement-learning Markov chain dynamics. *Phys. Rev. E* **98**, 063303 (2018).
191. Huang, L. & Wang, L. Accelerated Monte Carlo simulations with restricted Boltzmann machines. *Phys. Rev. B* **95**, 035105 (2017).
192. Bello, I., Pham, H., Le, Q. V., Norouzi, M. & Bengio, S. Neural combinatorial optimization with reinforcement learning. Preprint at https://arxiv.org/abs/1611.09940 (2016).
193. Dai, H., Khalil, E. B., Zhang, Y., Dilkina, B. & Song, L. Learning combinatorial optimization algorithms over graphs. Preprint at https://arxiv.org/abs/1704.01665 (2017).
194. Zhou, J. et al. Graph neural networks: a review of methods and applications. *AI Open* **1**, 57–81 (2020).
195. Dwivedi, V. P., Joshi, C. K., Laurent, T., Bengio, Y. & Bresson, X. Benchmarking graph neural networks. Preprint at https://arxiv.org/abs/2003.00982 (2020).
196. Schuetz, M. J., Brubaker, J. K. & Katzgraber, H. G. Combinatorial optimization with physics-inspired graph neural networks. Preprint at https://arxiv.org/abs/2107.01188 (2021).
197. Vinyals, O., Fortunato, M. & Jaitly, N. Pointer networks. Preprint at https://arxiv.org/abs/1506.03134 (2015).
198. Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **41**, 303–332 (1999).
199. Arora, S. & Barak, B. *Computational Complexity: A Modern Approach* (Cambridge Univ. Press, 2009).
200. Aaronson, S. BQP and the polynomial hierarchy. In *Proc. 42nd ACM Symp. Theory of Computing*, 141–150 (ACM, 2010).
201. Papadimitriou, C. H. & Yannakakis, M. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.* **43**, 425–440 (1991).
202. Goemans, M. X. & Williamson, D. P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM* **42**, 1115–1145 (1995).
203. Håstad, J. Some optimal inapproximability results. *J. ACM* **48**, 798–859 (2001).
204. Mukai, H., Tomonaga, A. & Tsai, J.-S. Superconducting quantum annealing architecture with LC resonators. *J. Phys. Soc. Japan* **88**, 061011 (2019).
205. Onodera, T., Ng, E. & McMahon, P. L. A quantum annealer with fully programmable all-to-all coupling via Floquet engineering. *npj Quantum Inf.* **6**, 1–10 (2020).
206. Lechner, W., Hauke, P. & Zoller, P. A quantum annealing architecture with all-to-all connectivity from local interactions. *Sci. Adv.* **1**, e1500838 (2015).
207. Puri, S., Andersen, C. K., Grimsmo, A. L. & Blais, A. Quantum annealing with all-to-all connected nonlinear oscillators. *Nat. Commun.* **8**, 1–9 (2017).
208. Kowalsky, M., Albash, T., Hen, I. & Lidar, D. A. 3-regular 3-XORSAT planted solutions benchmark of classical and quantum heuristic optimizers. Preprint at https://arxiv.org/abs/2103.08464 (2021).
209. Jörg, T., Krzakala, F., Semerjian, G. & Zamponi, F. First-order transitions and the performance of quantum algorithms in random optimization problems. *Phys. Rev. Lett.* **104**, 207206 (2010).
210. Aharonov, D. et al. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM Rev.* **50**, 755–787 (2008).
211. Mandra, S. & Katzgraber, H. G. A deceptive step towards quantum speedup detection. *Quantum Sci. Technol.* **3**, 04LT01 (2018).
212. Oshiyama, H. & Ohzeki, M. Benchmark of quantum-inspired heuristic solvers for quadratic unconstrained binary optimization. Preprint at https://arxiv.org/abs/2104.14096 (2021).
213. Chancellor, N. Modernizing quantum annealing using local searches. *New J. Phys.* **19**, 023024 (2017).
214. Bilbro, G. et al. Optimization by mean field annealing. In *Advances in Neural Information Processing Systems 1 (NIPS 1988)* (ed. Touretzky, D) 91–98 (Morgan Kaufmann, 1989).
215. Onodera, T. et al. Nonlinear quantum behavior of ultrashort-pulse optical parametric oscillators. Preprint at https://arxiv.org/abs/1811.10583 (2018).
216. Hamze, F. & de Freitas, N. From fields to trees. Preprint at https://arxiv.org/abs/1207.4149 (2012).
217. Selby, A. Efficient subgraph-based sampling of Ising-type models with frustration. Preprint at https://arxiv.org/abs/1409.3934 (2014).
218. Job, J. & Lidar, D. Test-driving 1000 qubits. *Quantum Sci. Technol.* **3**, 030501 (2018).
219. Mandra, S., Zhu, Z. & Katzgraber, H. G. Exponentially biased ground-state sampling of quantum annealing machines with transverse-field driving Hamiltonians. *Phys. Rev. Lett.* **118**, 070502 (2017).
220. Zhu, Z., Ochoa, A. J. & Katzgraber, H. G. Fair sampling of ground-state configurations of binary optimization problems. *Phys. Rev. E* **99**, 063314 (2019).
221. Albash, T. & Lidar, D. A. Adiabatic quantum computation. *Rev. Mod. Phys.* **90**, 015002 (2018).
222. King, J., Yarkoni, S., Nevisi, M. M., Hilton, J. P. & McGeoch, C. C. Benchmarking a quantum annealing processor with the time-to-target metric. Preprint at https://arxiv.org/abs/1508.05087 (2015).
223. Takesue, H., Inagaki, T., Inaba, K. & Honjo, T. Performance comparison between coherent Ising machines and quantum annealer. NTT R&D Technical Report (NTT Basic Research Laboratories, 2021); https://www.rd.ntt/e/research/JN202103_10945.html

Acknowledgements
The authors thank S. Aaronson, T. Albash, H. Katzgraber, T. Leleu, S. King, M. Narozniak, S. Vadlamani, T. van Vaerenbergh and D-Wave Systems for discussions and comments on the manuscript. T.B. is supported by the National Natural Science Foundation of China (62071301); NYU-ECNU Institute of Physics at NYU Shanghai; the Joint Physics Research Institute Challenge Grant; the Science and Technology Commission of Shanghai Municipality (19XD1423000,22ZR1444600); the NYU Shanghai Boost Fund; the China Foreign Experts Program (G2021013002L); and the NYU Shanghai Major-Grants Seed Fund. P.L.M. thanks all his collaborators on the topic of Ising machines — especially S. Ganguli, R. Hamerly, T. Leleu, H. Mabuchi, A. Marandi, E. Ng, T. Onodera and Y. Yamamoto — for discussions that have shaped his understanding over the years. P.L.M. acknowledges funding from NSF award CCF-1918549, and NTT Research for their financial and technical support. P.L.M. also acknowledges membership in the CIFAR Quantum Information Science Program as an Azrieli Global Scholar.

Author contributions
All authors contributed in compiling the results and preparing the manuscript.

Competing interests
P.L.M. declares an interest in QC Ware Corp., a company producing software for quantum computers, to which he is an advisor. T.B. and N.M. declare no competing interests.

Peer review information
Nature Reviews Physics thanks Natalia Berloff, Masano Yamaoka and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.