High-performance combinatorial optimization based on classical mechanics

Hayato Goto1*, Kotaro Endo2, Masaru Suzuki3, Yoshisato Sakai1, Taro Kanao1, Yohei Hamakawa1, Ryo Hidaka1, Masaya Yamasaki1, Kosuke Tatsumura1

Quickly obtaining optimal solutions of combinatorial optimization problems has tremendous value but is extremely difficult. Thus, various kinds of machines specially designed for combinatorial optimization have recently been proposed and developed. Toward the realization of higher-performance machines, here, we propose an algorithm based on classical mechanics, which is obtained by modifying a previously proposed algorithm called simulated bifurcation. Our proposed algorithm allows us to achieve not only high speed by parallel computing but also high solution accuracy for problems with up to one million binary variables. Benchmarking shows that our machine based on the algorithm achieves high performance compared to recently developed machines, including a quantum annealer using a superconducting circuit, a coherent Ising machine using a laser, and digital processors based on various algorithms. Thus, high-performance combinatorial optimization is realized by massively parallel implementations of the proposed algorithm based on classical mechanics.

INTRODUCTION

Combinatorial optimization problems appear in various social and industrial situations, so quickly solving such problems makes the society and industry more efficient. However, these problems are notoriously hard due to combinatorial explosion, an exponential increase in the number of candidate solutions depending on the problem size (1). Thus, novel computational approaches to combinatorial optimization have been expected. A well-known example is a quantum annealer (QA), which is based on quantum annealing (2–4) and its superconducting circuit implementation (5, 6). The QA is an Ising machine designed to find ground states of Ising spin models (7). Such Ising machines are believed to be broadly useful, because the Ising problem belongs to the nondeterministic polynomial time (NP)–complete class (7), and consequently, many combinatorial optimization problems can be reduced to the Ising problem (8). Various Ising machines other than QA have been developed, including coherent Ising machines (CIMs) implemented with pulse lasers (9–14) and other kinds of optical Ising machine (15–18), as well as digital processors based on simulated annealing (SA) (19–23), which is a standard heuristic algorithm for combinatorial optimization (1, 24, 25), or other recently proposed algorithms (26–29).

A heuristic algorithm called simulated bifurcation (SB) has recently been proposed for accelerating combinatorial optimization (30). SB is a purely quantum-inspired algorithm, that is, it was derived from a classical-mechanical model corresponding to a quantum computer called a quantum bifurcation machine (QbM) (30–33), which is based on quantum adiabatic optimization using nonlinear oscillators exhibiting quantum-mechanical bifurcation phenomena. Consequently, SB is based on numerical simulation of adiabatic evolutions in classical nonlinear Hamiltonian systems exhibiting bifurcations (34). Different dynamical approaches have also recently been proposed (35–41). SB is not based on the gradient method, unlike other dynamical approaches such as the Hopfield-Tank model (42), simulated CIM (SimCIM) (35, 39), and their variants, but based on adiabatic evolutions of energy conservative systems like purely adiabatic QA and QbM. [Such interesting contrast between QbM and CIM has been summarized in a review paper on this topic (33).]

Unlike SA in general cases, SB allows simultaneous updating of variables and therefore can easily accelerate combinatorial optimization through massively parallel processing using modern many-core processors such as field-programmable gate arrays (FPGAs) (30, 43, 44) and graphic processing units (GPUs) (30). An SB-based machine (SBM) implemented with a single FPGA, where about 8000 operations are performed in parallel, was able to find good approximate solutions of a 2000-spin Ising problem in 0.5 ms, about 10 times faster than a CIM (30). This result suggests that parallelizability is a key property of optimization algorithms for their acceleration by fully exploiting modern high-performance computing systems. In this direction of research, other parallelizable algorithms have also recently been proposed by mapping a given problem to a bipartite one and applying parallel SA updating to each group of spins (26–28). These new algorithms essentially rely on the same mechanism, although they are given different names: momentum annealing (MA) (26), stochastic cellular automata annealing (SCA) (27), and restricted Boltzmann machine (RBM)’s parallel stochastic sampling (28).

The previous results on SB demonstrate that SB is useful for quickly finding good approximate solutions. However, it remains unclear whether SB can find optimal solutions of large-scale problems. For enhancing the power of SB in terms of solution accuracy, in this work, we introduce two SB variants, named ballistic SB (bSB) and discrete SB (dSB), in addition to the original adiabatic SB (aSB). We solve various problems to compare the performance of bSB and dSB with that of aSB and other recently developed machines, including a QA, a CIM, and digital processors based on various algorithms. This benchmarking shows that bSB and dSB provide faster and more accurate optimizations than does aSB and that our new SBMs achieve high performance compared to the other machines. dSB can find optimal or near-optimal solutions of problems with up to one million spins, which aSB and bSB cannot achieve. Thus, high-performance machines for combinatorial optimization are realized by massively parallel implementations of the proposed algorithm based on classical mechanics.
parallel implementations of the proposed algorithms based on classical mechanics.

**RESULTS**

**bSB and dSB algorithms**

The Ising problem is to find a spin configuration minimizing the Ising energy, defined as

\[ E_{\text{Ising}} = - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} J_{ij} s_i s_j - \sum_{i=1}^{N} h_i s_i \]  

(1)

where \( s_i \) denotes the \( i \)th spin taking 1 or \(-1\), \( N \) is the number of spins, \( J_{ij} \) is the coupling coefficient between the \( i \)th and \( j \)th spins (\( J_{ij} = J_{ji} \) and \( J_{ii} = 0 \)), and \( h_i \) is the local field on the \( i \)th spin. Since introducing an ancillary spin reduces the Ising problem to the one without local fields (see section S1), here, we focus on the Ising problem with no local fields (\( h_i = 0 \)).

To solve the Ising problem, QbM uses quantum-mechanical nonlinear oscillators called Kerr-nonlinear parametric oscillators (KPOs), each of which can generate a Schrödinger cat state, i.e., a quantum superposition of two oscillating states, via a quantum-mechanical bifurcation (31). Such a KPO has recently been realized experimentally using superconducting circuits (45, 46). However, the realization of a large-scale QbM will require a long time. On the other hand, it was found that a classical-mechanical model corresponding to QbM, referred to as a classical bifurcation machine (CbM), also works as an Ising machine (31, 33). In this case, we can efficiently simulate such a classical machine using present digital computers, instead of building real machines. [This is not the case for QbM, because QbM can also be used as a universal quantum computer (47, 48), which is classically unsimulatable.] This simulation approach paves the way for large-scale Ising machines with all-to-all connectivity.

To modify the equations of motion for CbM such that computational costs are reduced and the symplectic Euler method (49) is applicable, we obtain the following Hamiltonian equations of motion for aSb (30)

\[ x_i = \frac{\partial H_{\text{aSb}}}{\partial y_i} = a_0 y_i \]  

(2)

\[ y_i = \frac{\partial H_{\text{aSb}}}{\partial x_i} = -\left( \frac{x_i^2}{4} + a_0 - a(t) \right)x_i + c_0 \sum_{j=1}^{N} J_{ij} x_j \]  

(3)

\[ H_{\text{aSb}} = \frac{a_0}{2} \sum_{i=1}^{N} y_i^2 + V_{\text{aSb}} \]  

(4)

\[ V_{\text{aSb}} = \sum_{i=1}^{N} \left( \frac{x_i^4}{4} + \frac{a_0 - a(t)}{2} x_i^2 \right) - \frac{c_0}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} J_{ij} x_i x_j \]  

(5)

where \( x_i \) and \( y_i \) are, respectively, the position and momentum of a particle corresponding to the \( i \)th spin, dots denote time derivatives, \( a(t) \) is a control parameter increased from zero, \( a_0 \) and \( c_0 \) are positive constants, and \( V_{\text{aSb}} \) is the potential energy in aSb.

To qualitatively explain the operation principle of aSb, we show an example of the dynamics in aSb in Fig. 1 (A and B), where the ferromagnetic two-spin Ising problem (\( f_{1,2} = f_{2,1} = 1 \)) with solutions \( s_1 = s_2 = \pm 1 \) is solved as the simplest problem. All positions and momenta are randomly set around zero at the initial time. The initial potential has a single local minimum at the origin (top and middle of Fig. 1B), so the particles circulate around the origin (Fig. 1A and middle of Fig. 1B). When \( a(t) \) becomes sufficiently large, bifurcations occur, that is, multiple local minima of the potential appear. Then, the particles adiabatically follow one of the minima. Consequently, each \( x_i \) bifurcates to a positive or negative value (Fig. 1A and bottom of Fig. 1B). Since two local minima corresponding to the two solutions have lower energies and appear earlier than the other two local minima, the particles successfully find one of the solutions. Last, the sign of \( x_i \), \( \text{sgn}(x_i) \), gives the \( i \)th spin, \( s_i \), for the solution of the Ising problem. It has been empirically found that aSb works well for much larger-scale and more complex problems (30).

This aSb relies on the fact that the second term in \( V_{\text{aSb}} \) is approximately proportional to the Ising energy at the final time (30). In this approximation, analog errors arise from the use of continuous variables (positions) instead of discrete variables (spins). These analog errors in aSb may degrade solution accuracy and result in approximate solutions. Such analog errors in different dynamical approaches have also been discussed (38, 40).

To suppress analog errors, we introduce perfectly inelastic walls at \( x_i = \pm 1 \). That is, at each time, we replace \( x_i \) with its sign, \( \text{sgn}(x_i) = \pm 1 \), and set \( y_i = 0 \) if \( |x_i| > 1 \). These walls force positions to be exactly equal to 1 or −1 when \( a(t) \) becomes sufficiently large. Moreover, we drop the fourth-order terms in \( V_{\text{aSb}} \) because the inelastic walls can play a role similar to the nonlinear potential walls. We thus obtain the following equations

\[ x_i = \frac{\partial H_{\text{bSb}}}{\partial y_i} = a_0 y_i \]  

(6)

\[ y_i = \frac{\partial H_{\text{bSb}}}{\partial x_i} = -[a_0 - a(t)]x_i + c_0 \sum_{j=1}^{N} J_{ij} x_j \]  

(7)

\[ H_{\text{bSb}} = \frac{a_0}{2} \sum_{i=1}^{N} y_i^2 + V_{\text{bSb}} \]  

(8)

\[ V_{\text{bSb}} = \frac{a_0 - a(t)}{2} \sum_{i=1}^{N} x_i^2 - \frac{c_0}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} J_{ij} x_i x_j \]  

when \( |x_i| \leq 1 \) for all \( x_i \) otherwise \( V_{\text{bSb}} = \infty \)  

(9)

This artificial dynamical system is the basis of the bSb. In bSb, we solve Eqs. 6 and 7 by the symplectic Euler method, where time is discretized with a time step \( \Delta t \), together with the updating for the inelastic walls (see Methods for a detailed algorithm). (If we solve these equations by the standard Euler method, instead of the symplectic Euler method, then solution accuracy becomes lower. See section S2.)

Similar modification to the above walls has been proposed for SimCIM (39), but this algorithm is based on the gradient method, like the Hopfield-Tank model (42), and also uses stochastic processes. In contrast, bSb is based on a classical-mechanical system conserving energy except for the inelastic walls and adiabatic changes of energy and also uses deterministic processes except for initial value setting. As a result, the performance of bSb is quite different from that of SimCIM (see section S3 for the comparison between bSb and SimCIM).

In bSb, it is sufficient to increase \( a(t) \) to \( a_0 \). Then, the final potential has only the second term related to the Ising energy. Consequently, the following condition is satisfied for all \( i \) at the final time (see section S4)

\[ \Delta E_i = 2s_i \sum_{j=1}^{N} J_{ij} s_j \geq 0 \]  

(10)

where \( s_i = \text{sgn}(x_i) \) is the sign of \( x_i \) and \( \Delta E_i \) represents the change in the Ising energy for a flip of \( s_i \). Note that Eq. 10 is a sufficient condition to show that the spin configuration is a local minimum of the
Ising problem. Hence, solutions obtained by bSB are at least local minima in the Ising problem. In contrast, this is not necessarily guaranteed in aSB because of its nonlinear potential terms. (This means that solutions obtained by aSB can sometimes be improved by a naïve local search based on sequential spin flips.) This is another reason why bSB should achieve higher accuracy than aSB. Throughout this work, we linearly increase $a(t)$ from 0 to $a_0$ and set $a_0 = 1$.

Here, we show an example of the bSB dynamics using the same two-spin problem as above. The initial potential has a single local minimum at the origin (top of Fig. 1D) and particles circulate around the origin (Fig. 1C and middle of Fig. 1D), as in aSB. In bSB, however, stable points suddenly jump from the origin to the walls at $x_i = \pm 1$, which prevents adiabatic evolution. Instead, particles move toward walls in a ballistic manner (Fig. 1C and bottom of Fig. 1D). This ballistic (nonadiabatic) behavior in bSB leads to fast convergence to a local minimum of $V_{bSB}$ and, consequently, to fast optimization.

For further improvement, we introduce another variant of SB by discretizing $x_j$ to $\text{sgn}(x_j)$ in the second term in Eq. 7.

---

**Fig. 1. Dynamics in three SB algorithms.** Two-spin ferromagnetic Ising problem ($J_{1,2} = J_{2,1} = 1$) is solved by aSB, bSB, and dSB with $a(t) = 0.01t$, $a_0 = 1$, and $c_0 = 0.2$. (A) Time evolutions of $x_1$ and $x_2$ in aSB. (B) Potential energies $V_{aSB}$ in aSB at the initial (top), intermediate (middle), and final (bottom) times. The vertical dotted line in (A) indicates the intermediate time. Curves in white are trajectories between the initial and intermediate times (middle) and between the intermediate and final times (bottom). Local minima of $V_{aSB}$ are shown by $+$ in red. (C to F) Corresponding results for bSB (C and D) and dSB (E and F). (G to I) Time dependences of potential energy and total energy in aSB (G), bSB (H), and dSB (I). In each panel, the inset magnifies the part in the red dashed square. The dashed arrow in the inset in (I) indicates tunneling-like behavior in dSB. The red bold curve in the middle of (F) shows the trajectory corresponding to this inset.
The discrete version of bSB, namely, dSB (see Methods for a detailed algorithm), is expected to achieve higher accuracy than that of bSB, because analog errors are further suppressed by discretization. We found that this discretization is effective for bSB but not for other algorithms such as SimCIM and aSB (see sections S3 and S5).

Note that the singularity on the boundaries between positive and negative regions has been intentionally neglected. This leads to a violation of conservation of energy across boundaries and, hence, to escape from local minima over potential barriers, as shown below. In this sense, dSB goes beyond naive algorithms based on classical-mechanical systems conserving energy (except adiabatic change), such as aSB and bSB. We also increase \( a(t) \) to \( a_0 \) for the convergence to a local minimum of the Ising problem at the final time, as in bSB (see section S4).

Figure 1 (E and F) shows an example of the dSB dynamics using the same two-spin problem as above. Unlike aSB and bSB, the particles go back and forth between two local minima through the potential barriers (Fig. 1E and middle of Fig. 1F). This is similar to quantum tunneling, as depicted by the inset in Fig. 1I. This tunneling-like behavior is possible due to the above-mentioned neglect of the singularity on the boundaries; otherwise, the potential walls on the boundaries prevent this tunneling. In contrast, conservation of energy prevents such tunneling in aSB and bSB, as suggested by the insets in Fig. 1 (G and H). Thus, it is expected that this tunneling-like behavior will help dSB to escape local minima of the potential, and hence, dSB will outperform aSB and bSB in terms of solution accuracy.

Note that both bSB and dSB maintain the advantage of aSB over SA, namely, high parallelizability. Therefore, they are expected to realize both high speed and high accuracy simultaneously.

**Performance for a 2000-spin Ising problem with all-to-all connectivity**

To compare the performance of bSB and dSB with that of aSB, we solved a 2000-spin Ising problem with all-to-all connectivity. This problem was named \( K_{2000} \) and previously solved by aSB \((30)\), a CIM \((11)\), and a recently developed digital chip called STATICA \((27)\), which is based on the above-mentioned SCA. This problem can be regarded as a 2000-node MAX-CUT problem \((11, 30)\); so, here, we evaluate performance using cut values (see section S6 for the definition of the cut value and the relation between MAX-CUT and the Ising problem). The best cut value for \( K_{2000} \) is estimated to be 33,337 (see section S7).

The lines and symbols in Fig. 2A show average and maximum cut values, respectively, for 1000 trials as functions of the number of time steps, \( N_{\text{step}} \). (Throughout the paper, \( N_{\text{step}} \) denotes the total number of time steps for each trial and the values of cost functions (cut values or Ising energies) as functions of \( N_{\text{step}} \) are final values, not intermediate values, in each trial.) The results clearly show that both bSB and dSB outperform aSB in terms of both speed and accuracy. In addition, only dSB obtained the best value. On the other hand, best values obtained by bSB and aSB become lower for larger \( N_{\text{step}} \). This result suggests that the best values may be obtained accidentally by nonadiabatic processes in bSB and aSB. For large \( N_{\text{step}} \), dynamics becomes more adiabatic and the chance to obtain better solutions by nonadiabatic processes may be lost.

We implemented 2048-spin-size bSB and dSB machines (bSBM and dSBM) using single FPGAs (see section S8 for details) and solved \( K_{2000} \) by them. Figure 2B shows the comparison between our machines and the above three other machines \((11, 27, 30)\), where the lines and the crosses show the average values of our machines and the others, respectively, for 100 trials, and the bars show the maximum and minimum values among the 100 trials. (The bars for our machines are shown at only typical computation times.) Only our dSBM obtained the best value in a short time (2 ms), thereby simultaneously realizing both high speed and high accuracy. Also, our bSBM is remarkably fast, about three times faster than STATICA \((27)\), the previously fastest machine for \( K_{2000} \). Note that the results by STATICA for \( K_{2000} \) are predicted values by a simulator, and a real STATICA
chip is still 512-spin size (27). On the other hand, in this work, we have implemented faster real machines.

**Benchmarking using time-to-solution and time-to-target**

To evaluate the computation speed more quantitatively, here, we introduce two metrics: time-to-solution (TTS) and time-to-target (TTT). TTS is a standard metric for evaluating Ising machine speeds (14, 23, 28, 29), defined as the computation time for finding an optimal or best known value with 99% probability. TTT uses a target value, instead of an optimal value, as a good approximate solution. In this work, we define the target as 99% of the optimal or best known value. TTS and TTT are formulated as $T_{\text{com}} \log (1 - 0.99) / \log (1 - P_S)$ (14, 23), where $T_{\text{com}}$ is the computation time per trial and $P_S$ is the success probability for finding the optimal (TTS) or target (TTT) value. $P_S$ is estimated from experimental results with many trials. When $P_S > 0.99$, TTS and TTT are defined as $T_{\text{com}}$.

In the following, we compare TTS and TTT of our 2048-spin-size bSBM and dSBM with those of other recently developed machines shown in Fig. 3A. Since the bSBM can quickly find good approximate solutions and the dSBM can find optimal solutions of large-scale problems, we use the bSBM and dSBM for evaluations of TTT and TTS, respectively. TTS and TTT of other machines are cited or estimated from the data in the literature (see section S9 for details), be-

The 2000-spin-size instance with sparse connectivity, we also solved G22, for which the TTT of our bSBM is two orders of magnitude shorter than that of the CIM. These results demonstrate that our bSBM can find good approximate solutions faster than other recently developed machines of the same spin size. (TTTs of our machines for other G-set instances are provided in table S2.)

Next, we show the results of TTS in Fig. 3C. We start with the same two instances, namely, K2000 and G22. The TTS of our dSBM for them are 1.3 and 2.7 s, respectively. While TTS for K2000 has not been reported so far, TTS for G22 was evaluated with a SimCIM implemented on a FPGA (29), which is based on a recently proposed algorithm called chaotic amplitude control (CAC) (29, 40). The TTS of the SimCIM is estimated at 12 s (see section S9). Thus, our dSBM has achieved a shorter TTS for G22 than that of the state-of-the-art machine. (TTTs of our machines for other G-set instances and the comparison between them and those of the SimCIM are provided in table S2 and fig. S6, respectively.)

We also solved other various instances of the Ising problem (MAX-CUT) by dSBM to compare it with other machines shown in Fig. 3A. For small-scale problems, we can simultaneously perform multiple trials using the 2048-spin-size machine by a block-diagonal structure of the $J$ matrix, as done using a CIM (14). This so-called batch processing improves the success probability $P_S$ by selecting the best result among multiple trials, while $T_{\text{com}}$ is defined as the computation time per batch. In the limit as the number of trials per batch $N_{\text{batch}}$ goes to infinity, $P_S$ may exceed 0.99, and then TTS and TTT become $T_{\text{com}}$ from the above definitions. In this sense, the TTS and TTT are well defined even for batch processing.

As shown in Fig. 3C, for two small-scale problems with 60 spins (all-to-all connectivity) and 200 spins (sparse connectivity), our dSBM achieved much shorter TTSs than those of a QA and a CIM (14). (This QA is not the latest version.) For 100-spin and 700-spin problems with all-to-all connectivity, the TTSs of the dSBM are much shorter than those of the SimCIM with CAC (29). These short TTSs of dSBM compared to the SimCIM come not from computation speed or implementations but the algorithmic advantage of dSB over CAC. That is, dSB needs fewer matrix-vector multiplications (MVMs), which is the most computation-intensive part in both algorithms, to obtain solutions than does the SimCIM with CAC. (The numbers of MVMs to solutions of the 100-spin and 700-spin problems are $9.4 \times 10^4$ and $8.1 \times 10^5$, respectively, for dSB but $5.6 \times 10^5$ and $7.8 \times 10^5$, respectively, for CAC (29).) For two 1024-spin problems
Fig. 4. Results for ultralarge-scale Ising problems. (A) Computation times of aSB, bSB, dSB, and our best SA implemented on a 16-GPU cluster, aSB and SA (25) running on a CPU core, and MA implemented on a four-GPU cluster (26) for a 100,000-spin Ising problem with all-to-all connectivity. Each coupling coefficient $J_{ij}$ was a randomly chosen 10-bit integer in $\{-2^9 + 1, \ldots, 2^9 - 1\}$ (26). Lines and symbols represent average and best values, respectively, for 100 trials (one trial for aSB and SA running on a CPU). (B) Magnification of (A) around the estimated optimal value. (C) and (D) Corresponding results for a 1,000,000-spin Ising problem with a sparse connectivity of 1%. Each 1% nonzero $J_{ij}$ was a randomly chosen 10-bit integer in $\{-2^9 + 1, \ldots, 2^9 - 1\}$.
(all-to-all connectivity) with different ranges of $f_{ij}$, the dSBM achieved shorter TTSs than those of a Digital Annealer (DA) (23), which is based on an FPGA implementation of “Digital Annealer’s algorithm” developed from SA and outperformed CPU implementations of SA (25) and parallel tempering (50). (This DA is not the latest version.) Last, for 60-spin and 100-spin problems with all-to-all connectivity, the TTSs of the dSBM are comparable to those of another state-of-the-art machine based on an FPGA implementation of the above-mentioned RBM’s stochastic sampling (28), the size of which, however, is limited to 200 spins. Overall, we conclude that our bSBM and dSBM have achieved remarkably high performance for the present benchmark instances compared to the recently developed machines.

**Performance for ultralarge-scale Ising problems**

Last, we present the results for two ultralarge-scale Ising problems: a 100,000-spin problem with all-to-all connectivity and a 1,000,000-spin problem with a sparse connectivity of 1% (see section S7 for their detailed definitions for reproduction). Using a GPU cluster with 16 GPUs, we solved these by aSB, bSB, dSB, and our best implementation of SA (see section S10 for details). For comparison, we also solved them by aSB and SA (25) running on a CPU core. Figure 4 shows the results, where the result obtained by the four-GPU implementation of the above-mentioned MA (26) is also shown. The horizontal lines show optimal (dashed) and target (dotted) values estimated using a formula based on statistical mechanics (see section S7 for details) (51, 52).

Figure 4A shows that all three GPU-cluster SBMs outperformed the MA machine (26) in terms of both speed and accuracy. Figure 4 (A and B) also show that the GPU-cluster implementation achieved about 1000 times speedup over a CPU core for aSB but only about 100 times for SA. This difference comes from the higher parallelizability of aSB than that of SA. The GPU-cluster bSBM and dSBM are faster than the GPU-cluster aSBM, because of their algorithmic advantage. Figure 4B shows that the dSBM achieved the closest value to the estimated optimal value, giving it the highest accuracy. As shown in Fig. 4 (C and D), similar results also hold for the 1,000,000-spin Ising problem with sparse connectivity.

**DISCUSSION**

In this work, we have proposed two new variants of the SB algorithm, named bSB and dSB, both of which outperform the original aSB in terms of both speed and solution accuracy. dSB allows us to find optimal solutions of large-scale problems, which aSB and bSB cannot achieve. We have implemented 2048-spin-size bSBM and dSBM using single FPGAs. Our benchmarking with TTS and TTT has shown that the bSBM and dSBM can achieve remarkably high performance compared to other recently developed machines. GPU-cluster implementations of bSB and dSB also allow us to find nearly optimal solutions of ultralarge-scale problems with up to one million spins.

Last, we discuss possible future works on SB. First, it is important to check the performance of SB for a broader range of instances than those evaluated in this work, which were chosen to compare our machines with previously developed machines reported in the literature. It is known that a single solver cannot achieve the highest performance for all kinds of instances (53). Thus, we should examine what kinds of instances can be solved well by SB. Second, it is desirable to develop a technique for auto-tuning of parameters in SB, such as a constant $c_0$ and a time step $\Delta t$. In this work, we used the definition of $c_0$ based on random matrix theory (30) and also chose the best value of $\Delta t$ among five values (see Methods for details). Such preliminary search for the best parameter values is often used for benchmarking to evaluate potential performance of solvers. In practical applications, however, it is desirable to determine appropriate parameter values automatically without preliminary search. Last, development of other variants of SB is an interesting possibility. In this work, we have focused on the Ising problem (MAX-CUT). The generalization of SB to broader classes of problem, e.g., combinatorial optimization with higher-order polynomial cost functions, is an interesting next target.

**METHODS**

**Ballistic simulated bifurcation**

In bSB, we numerically solve the Hamiltonian equations of motion given by Eqs. 6 and 7 using the symplectic Euler method (49), as in aSB (30). The updating rule for bSB is as follows

$$y_i(t_{k+1}) = y_i(t_k) + \left\{-y_i(t_k) + \Delta t \sum_{j=1}^{N} J_{ij} y_j(t_k) + c_0 \right\} \Delta t$$

$$x_i(t_{k+1}) = \begin{cases} x_i(t_k) + a_0 y_i(t_{k+1}) & \text{for } x_i < 0 \\ x_i(t_k) - a_0 y_i(t_{k+1}) & \text{for } x_i > 0 \end{cases} \Delta t$$

Here, $\Delta t$ is the time step and $t_k$ is the discretized time satisfying $t_0 = 0$ and $t_{k+1} = t_k + \Delta t$. After updating $x_i$, we check whether $|x_i| > 1$. If so, we replace $x_i$ with $\text{sgn}(x_i)$ and set $y_i = 0$, which implements perfectly inelastic walls at $x_i = \pm 1$.

**Discrete simulated bifurcation**

In dSB, we numerically solve Eqs. 11 and 12. The updating rule for dSB is as follows

$$y_i(t_{k+1}) = y_i(t_k) + \left\{-y_i(t_k) + \Delta t \sum_{j=1}^{N} J_{ij} \text{sgn}\{y_j(t_k)\} \right\} \Delta t$$

$$x_i(t_{k+1}) = \begin{cases} x_i(t_k) + a_0 y_i(t_{k+1}) & \text{for } x_i < 0 \\ x_i(t_k) - a_0 y_i(t_{k+1}) & \text{for } x_i > 0 \end{cases} \Delta t$$

As in bSB, we replace $x_i$ with $\text{sgn}(x_i)$ and set $y_i = 0$ if $|x_i| > 1$.

**Efficient implementations of bSB and dSB**

The bSB and dSB can be implemented more efficiently as follows. [A similar speedup technique has been used for SA (25).]

Storing $z_i(k) = \sum_{j=1}^{N} J_{ij} x_j(t_k)$, we can rewrite Eq. 15 as

$$y_i(t_{k+1}) = y_i(t_k) + \{-y_i(t_k) + c_0 z_i(k)\} \Delta t$$

$$z_i(k) = z_i(k-1) + \sum_{j=1}^{N} J_{ij} \Delta x_j(k)$$

where $\Delta x_j(k) = x_j(t_k) - x_j(t_{k-1})$. Note that $\Delta x_j(k)$ often becomes zero around the final time. (This is not the case for aSB, making this implementation ineffectual.) Hence, the product-sum operation in Eq. 20 can be accelerated by neglecting the terms corresponding to $\Delta x_j(k) = 0$.

In the case of dSB, instead of $\Delta x_j(k)$, we use $\Delta s_j(k) = \text{sgn}\{s_j(t_k)\} - \text{sgn}\{s_j(t_{k-1})\}$ as

$$z_i(k) = z_i(k-1) + \sum_{j=1}^{N} J_{ij} \Delta s_j(k)$$

**Performance for ultralarge-scale Ising problems**

Last, we discuss possible future works on SB. First, it is important to check the performance of SB for a broader range of instances than those evaluated in this work, which were chosen to compare our machines with previously developed machines reported in the literature. It is known that a single solver cannot achieve the highest performance for all kinds of instances (53). Thus, we should examine what kinds of instances can be solved well by SB. Second, it is desirable to develop a technique for auto-tuning of parameters in SB, such as a constant $c_0$ and a time step $\Delta t$. In this work, we used the definition of $c_0$ based on random matrix theory (30) and also chose the best value of $\Delta t$ among five values (see Methods for details). Such preliminary search for the best parameter values is often used for benchmarking to evaluate potential performance of solvers. In practical applications, however, it is desirable to determine appropriate parameter values automatically without preliminary search. Last, development of other variants of SB is an interesting possibility. In this work, we have focused on the Ising problem (MAX-CUT). The generalization of SB to broader classes of problem, e.g., combinatorial optimization with higher-order polynomial cost functions, is an interesting next target.
Since $\Delta s_j(k)$ becomes zero more often than $\Delta x_j(k)$, this implementation is more effective for speedup in dSB than in bSB.

**Parameter setting**

In the SB algorithms, the time step $\Delta t$ and the constants $c_0$ and $c_1$ are appropriately set in advance. In this work, we set the constants as $a_0 = 1$ and $c_0 = \frac{0.5}{\langle J \rangle}$, where $\langle J \rangle$ is defined as $\langle J \rangle = \frac{1}{N(N - 1)} \sum_j J_{ij}^2$. This definition of $c_0$ is based on random matrix theory (30). Although this setting of $c_0$ may not be optimal for some instances, this is enough to achieve high performance presented in this work. On the other hand, the setting of $\Delta t$ is more sensitive to performance. We therefore selected the best value among five values: 0.25, 0.5, 0.75, 1, and 1.25. In Figs. 2A and 4, $\Delta t$ is set to 0.5 (asSB) or 1 (bSB and dSB). In Fig. 2B, $\Delta t$ is set to 1.25 for the bSBM and dSBM. The $\Delta t$ values used in Fig. 3 are provided in table S1. As mentioned in Discussion, automatic setting of $c_0$ and $\Delta t$ is an important issue but beyond the scope of the present work and so left for future work.

**SUPPLEMENTARY MATERIALS**

Supplementary material for this article is available at http://advances.sciencemag.org/cgi/content/full/7/6/eabe7953/DC1

**REFERENCES AND NOTES**

1. P. Storrs, *Metaheuristics* (Springer International Publishing, 2016).
2. T. Kadawski, H. Nishimori, Quantum annealing in the transverse ising model. *Phys. Rev. E* **58**, 5355–5363 (1998).
3. G. E. Santoro, R. Martoňák, E. Tosatti, R. Car, Theory of quantum annealing of an Ising spin glass. *Science* **295**, 2427–2430 (2002).
4. W. M. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Buryk, E. M. Chapple, C. Cerverud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. S. Truncik, S. Uchachin, J. Wang, B. Wilson, G. Rose, Quantum annealing with manufactured spins. *Nature* **473**, 194–198 (2011).
5. S. Boixo, T. F. Ranyon, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, M. Troyer, Evidence for quantum annealing with more than one hundred qubits. *Nat. Phys.* **10**, 218–224 (2014).
6. F. Barahona, On the computational complexity of ising spin glass models. *J. Phys. A* **15**, 3241–3253 (1982).
7. A. Lucas, Ising formulations of many NP problems. *Front. Phys.* **2**, 5 (2014).
8. Z. Wang, A. Marandi, K. Wen, R. L. Byer, Y. Yamamoto, Coherent ising machine based on degenerate optical parametric oscillators. *Phys. Rev. A* **88**, 063853 (2013).
9. A. Marandi, Z. Wang, K. Takata, R. L. Byer, Y. Yamamoto, Network of time-multiplexed optical parametric oscillators as a coherent ising machine. *Nat. Photonics* **8**, 937–942 (2014).
10. T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeda, K. Enbutsu, K. Aihara, R. Kasahara, S. Utsunomiya, H. Takesue, A. H. Safavi-Naeini, Quantum dynamics of a few-photon parametric oscillator. *Phys. Rev. Lett.* **122**, 213902 (2019).
11. T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeda, K. Enbutsu, K. Aihara, R. Kasahara, S. Utsunomiya, H. Takesue, A. H. Safavi-Naeini, Quantum dynamics of a few-photon parametric oscillator. *Phys. Rev. Lett.* **122**, 213902 (2019).
12. T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeda, K. Enbutsu, K. Aihara, R. Kasahara, S. Utsunomiya, H. Takesue, A. H. Safavi-Naeini, Quantum dynamics of a few-photon parametric oscillator. *Phys. Rev. Lett.* **122**, 213902 (2019).
13. T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeda, K. Enbutsu, K. Aihara, R. Kasahara, S. Utsunomiya, H. Takesue, A. H. Safavi-Naeini, Quantum dynamics of a few-photon parametric oscillator. *Phys. Rev. Lett.* **122**, 213902 (2019).
14. T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeda, K. Enbutsu, K. Aihara, R. Kasahara, S. Utsunomiya, H. Takesue, A. H. Safavi-Naeini, Quantum dynamics of a few-photon parametric oscillator. *Phys. Rev. Lett.* **122**, 213902 (2019).
15. T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeda, K. Enbutsu, K. Aihara, R. Kasahara, S. Utsunomiya, H. Takesue, A. H. Safavi-Naeini, Quantum dynamics of a few-photon parametric oscillator. *Phys. Rev. Lett.* **122**, 213902 (2019).
16. T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeda, K. Enbutsu, K. Aihara, R. Kasahara, S. Utsunomiya, H. Takesue, A. H. Safavi-Naeini, Quantum dynamics of a few-photon parametric oscillator. *Phys. Rev. Lett.* **122**, 213902 (2019).
46. A. Grimm, N. E. Frattini, S. Puri, S. O. Mundhada, S. Touzard, M. Mirrahimi, S. M. Girvin, S. Shankar, M. H. Devoret, Stabilization and operation of a Kerr-cat qubit. Nature 584, 205–209 (2020).

47. H. Goto, Universal quantum computation with a nonlinear oscillator network. Phys. Rev. A 93, 050301 (2016).

48. S. Puri, S. Boutin, A. Blais, Engineering the quantum states of light in a Kerr-nonlinear resonator by two-photon driving. npj Quantum Inf. 3, 18 (2017).

49. B. Leimkuhler, S. Reich, Simulating Hamiltonian Dynamics (Cambridge Univ. Press, Cambridge, 2004).

50. K. Hukushima, K. Nemoto, Exchange Monte Carlo method and application to spin glass simulations. J. Physical Soc. Japan 65, 1604–1608 (1996).

51. S. Boettcher, Simulations of ground state fluctuations in mean-field Ising spin glasses. J. Stat. Mech., P07002 (2010).

52. S. Boettcher, Ground state properties of the diluted sherrington-kirkpatrick spin glass. Phys. Rev. Lett. 124, 177202 (2020).

53. I. Dunning, S. Gupta, J. Silberholz, What works best when? A systematic evaluation of heuristics for Max-Cut and QUBO. INFORMS J. Comput. 30, 608–624 (2018).

54. W.-K. Ma, T. N. Davidson, K. M. Wong, Z.-Q. Luo, P.-C. Ching, Quasi-maximum-likelihood multiuser detection using semi-definite relaxation with application to synchronous CDMA. IEEE Trans. Signal Proces. 50, 912–922 (2002).

55. Z. Luo, W. Ma, A. M. So, Y. Ye, S. Zhang, Semidefinite relaxation of quadratic optimization problems. IEEE Signal Proces. Mag. 27, 20–34 (2010).

56. P. E. Kloeden, E. Platen, Numerical Solution of Stochastic Differential Equations (Springer-Verlag, 2010).

57. F. Ma, J.-K. Hao, A multiple search operator heuristic for the max-k-cut problem. Annals Operations Res. 248, 365–403 (2017).

58. F. Liers, M. Jünger, Spin Glass Server; http://informatik.uni-koeln.de/spinglass/.

59. K. Hukushima, K. Nemoto, Exchange Monte Carlo method and application to spin glass simulations. J. Physical Soc. Japan 65, 1604–1608 (1996).

60. M. Matsumoto, T. Nishimura, Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Trans. Model. Comput. Simul. 8, 3–30 (1998).

61. D. Lewis, G. Chiu, J. Chromczak, D. Galloway, B. Gamas, V. Manoharajah, I. Milton, T. Vanderhoek, J. Van Dyken, The Stratix 10 Highly Pipelined FPGA architecture. Proceedings of ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA) (2016), pp. 159–168.

Acknowledgments: We thank Y. Kaneko, O. Hori, M. Watabiki, M. Iwasaki, M. Ootomo, M. Tomoda, and Y. Izumi for comments and support. Funding: The authors acknowledge no funding in support of this research. Author contributions: H.G. described the dynamical properties of three SBs by producing Fig. 1; estimated optimal solutions for ultralarge instances with T.K.; wrote the manuscript and the Supplementary Materials with data from K.E., Y.H., and R.H.; and supervised the project. K.E. conceived bSB and dSB through discussion with M.S., implemented SBs and SA on a GPU and a 16-GPU cluster, and collected performance data; Y.S. pointed out the mapping of the Ising problem with local fields to one without local fields and convergence to a local minimum of the Ising problem in bSB and dSB; Y.H. and R.H. implemented dSB and bSB, respectively, on an FPGA and collected performance data; M.Y. optimized the FPGA implementations; and K.T. developed the basis of the FPGA implementations, wrote the part of FPGA implementations in the Supplementary Materials, and supervised the FPGA team. Competing interests: H.G., K.E., M.S., Y.S., and K.T. are inventors on a Japanese patent application related to this work filed by the Toshiba Corporation (no. P2019-164742, filed 10 September 2019). The authors declare that they have no other competing interests. Data and materials availability: All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials. Additional data related to this paper may be requested to the authors.

Submitted 15 September 2020
Accepted 16 December 2020
Published 3 February 2021
10.1126/sciadv.abe7953

Citation: H. Goto, K. Endo, M. Suzuki, Y. Sakai, T. Kanao, Y. Hamakawa, R. Hidaka, M. Yamasaki, K. Tatsumura, High-performance combinatorial optimization based on classical mechanics. Sci. Adv. 7, eabe7953 (2021).