A Review of Simulation Algorithms of Classical Ising Machines for Combinatorial Optimization

Tingting Zhang
Department of ECE
University of Alberta
Edmonton, Canada
ttzhang@ualberta.ca

Qichao Tao
Department of ECE
University of Alberta
Edmonton, Canada
qichao@ualberta.ca

Bailiang Liu
Department of ECE
University of Alberta
Edmonton, Canada
bailiang@ualberta.ca

Jie Han
Department of ECE
University of Alberta
Edmonton, Canada
jhan8@ualberta.ca

***Abstract*—Combinatorial optimization problems are difficult to solve due to the space explosion in an exhaustive search. Using Ising model-based solvers can efficiently find near-optimal solutions by minimizing the energy of a nonlinear Hamiltonian system. In contrast to Ising machines based on quantum mechanics, classical Ising machines using conventional technologies, such as the complementary metal–oxide–semiconductor, offer efficient implementations with competitive performance. In this paper, we briefly review recently developed simulation algorithms of classical Ising machines. These algorithms are classified by considering various inherent mechanisms in the simulation of physical phenomena. Then, strategies to improve the simulation efficiency are discussed by generalizing their characteristics and behaviours. These simulation algorithms are key for improving the efficiency of classical Ising machines in solving combinatorial optimization problems.**

***Index Terms*—Combinatorial optimization, annealing, bifurcation, coherent Ising machine, Ising model**

I. Introduction

Combinatorial optimization (CO) is an important task in various social and industrial applications, such as machine learning, chip design, and data mining [1]. However, CO problems are non-deterministic polynomial time (or NP)-hard, characterized by the exponentially increasing number of candidate solutions as the problem size increases. It is very challenging to solve such problems by using enumeration. For example, an enumeration method needs to traverse all $(M-1)!$ possible routes to solve a traveling salesman problem (TSP) of M cities, which is prohibitive when M is large.

The Ising model has recently emerged as an efficient method to solve a CO problem by maximizing (or minimizing) an evaluation function under a given set of constraints. It describes the ferromagnetism of magnetic spins in statistical mechanics [2]. An Ising machine aims to find the ground state (i.e., the lowest-energy state) of an Ising model. Various Ising machines have been designed, including those implemented in superconducting circuits based on quantum annealing [3], [4] and the coherent Ising machines (CIMs) implemented using optical parametric oscillators [5], [6]. However, it is challenging to build those systems due to the requirements of cryogenic environments or a long optical fibre. Therefore, classical Ising machines [7]–[12] have been developed to offer inexpensive implementations and an easier integration with complementary metal–oxide–semiconductor (CMOS) circuits.

At the core of an Ising machine, the algorithm plays an important role in the solution search process. Simulation algorithms, which emulate certain physical phenomena on classical computers, have been developed for solving CO problems by decreasing the energy of the Ising model. In this paper, we review these algorithms for classical Ising machines to solve CO problems.

The remainder of this paper is organized as follows. Section II presents the background. In Section III, the simulation algorithms for Ising models are classified. Classical Ising machines based on the simulation algorithms are reviewed in Section IV. Section V concludes this paper and discusses the challenges and prospects.

II. Background

A. The Ising Model

The Hamiltonian of the Ising model (H) with the external magnetic field is defined as [2]

$$H = -\sum_{i}\sum_{j} J_{i,j}\sigma_i\sigma_j - \sum_{i} h_i\sigma_i, \tag{1}$$

where σ_i (or σ_j) denotes the state of the ith (or jth) spin that takes the value of -1 (as the down state) or $+1$ (as the up state), $J_{i,j}$ is the interaction coefficient between σ_i and σ_j, and h_i is the external magnetic field on σ_i.

B. Combinatorial Optimization (CO)

CO searches for the solution that indicates the maxima (or minima) of an objective function and that satisfies the given constraints. It can be formulated as a quadratic unconstrained binary optimization (QUBO) problem. Let $\mathbb{B} = \{0, +1\}$ be a binary set, $\mathbb{N}$ be a set of integers, and $\mathbb{R}$ be a set of real numbers. Given $\mathbb{B}$, $\mathbb{N}$ and $\mathbb{R}$, a QUBO problem can be described as [13]

$$\min_{\boldsymbol{x}\in\mathbb{B}^N} F(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{b} + c, \tag{2}$$

where $\boldsymbol{x}$ ($\boldsymbol{x} \in \mathbb{B}^N$, $N \in \mathbb{N}$) is a vector of binary variables, $\boldsymbol{A}$ ($\in \mathbb{R}^{N\times N}$), $\boldsymbol{b}$ ($\in \mathbb{R}^N$) and c ($\in \mathbb{R}$) denote the weight matrix, the weight vector and a constant scalar, respectively.

By using $\boldsymbol{x} = \frac{\boldsymbol{1}+\boldsymbol{\sigma}}{\boldsymbol{2}}$, where $\boldsymbol{\sigma} \in \{-1, +1\}^N$, (2) can be converted to the expression in (1). Then, a QUBO problem can easily be mapped to an Ising model. The configuration of spins at the ground state provides the optimal solution.

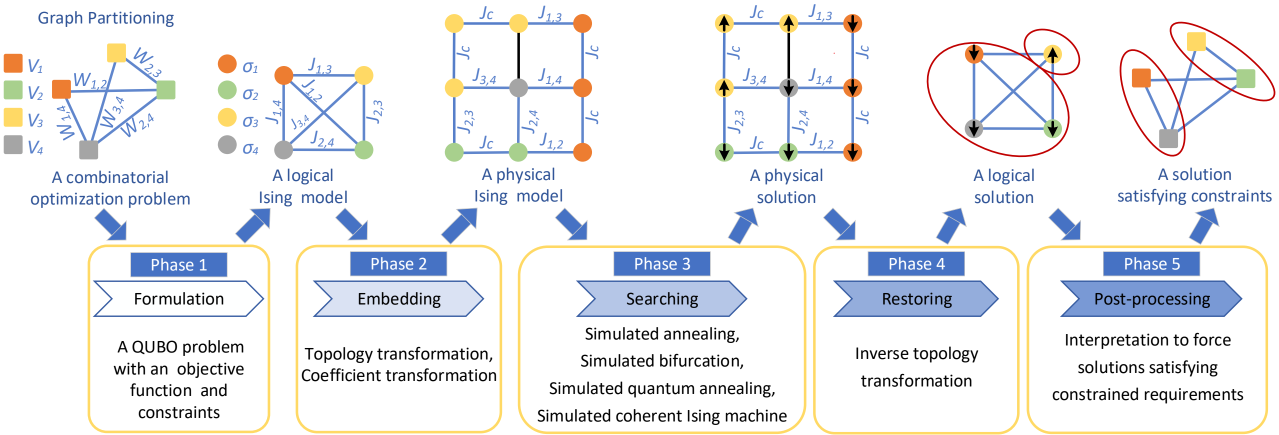

Fig. 1. Solving a combinatorial optimization problem using an Ising machine. An example of graph partitioning is given at the top. Given an undirected graph with an even number of vertices (V) and edges with weights (W), graph partitioning divides the vertices into two subsets of an equal size (as the constraint) with the sum of the weights of edges between the vertices belonging to different subsets minimized (as the objective). To this end, the problem is first mapped to a logical Ising model, where the weights of edges (W) are converted to the interactions between spins (J) by considering the objective and the constraint. ..Then, the logical Ising model is embedded into the topology of an Ising machine, where J_C is the coupling strength between duplicated spins to ensure that they are in the same states [15]. The solution found in Phase 3 is interpolated back to the logical Ising model, which may provide a solution that violates the constraint. Finally, the solution is modified to satisfy the constraint in Phase 5.

C. Problem Solving via Ising Machines

As shown in Fig. 1, solving a CO problem using an Ising machine consists of the following five phases [14]:

Phase 1: A CO problem formulated as a QUBO problem is mapped to a logical Ising model that describes the problem without any restriction in the topology and the precision of coefficients [1].

Phase 2: This logical Ising model is converted to a physical Ising model that can readily be embedded into the topology of an Ising machine and that meets the requirement for the bit width of coefficients [14]–[17].

Phase 3: A specific algorithm searches for the ground state of the Hamiltonian of the physical Ising model [18]–[23].

Phase 4: The result from Phase 3 is restored back to the logical Ising model using the inverse steps that are specified in the embedding method in Phase 2 [16].

Phase 5: The solution found may not meet the constraints due to the stochastic behavior of the Ising model. Thus, interpretation methods are applied to force the solution to satisfy the constraints by identifying and modifying the states of spins that cause the violation of constraints [24].

III. A Classified Review

A. Simulated Annealing (SA)

As a generic algorithm for the classical Ising machine, simulated annealing (SA) emulates the thermal annealing process in metallurgy [18]. Initially, the spin states are randomly configured at the beginning of the annealing process with a high temperature. Then, the Hamiltonian decreases through flipping the states of spins with a decreasing temperature. The flip probability for the ith spin (P_i) depends on the energy variation (ΔE_i) and the temperature (T), given by $P_i = min\{1, exp(-\Delta E_i/T)\}$, where ΔE_i indicates the variation of Hamiltonian when σ_i is flipped. With a lower temperature, the flip probability decreases. This randomness is introduced to help the Ising model to escape from the local minimum energy states. Finally, the Ising model approaches the ground state when the temperature decreases to a sufficiently low value.

The computational flow of SA is divided into three phases: (1) the calculation of ΔE_i, (2) the update of spin states, and (3) the random flip of spins. In what follows, we discuss the typical methods for accelerating the annealing process and improving the hardware efficiency during these three phases.

1) The Calculation of ΔE_i: An N-spin Ising model calculates ΔE_i through accumulation by N times. The calculation of ΔE_i is the most time-consuming step in SA.

The parallel computation of ΔE_i is considered in the stochastic cellular automata annealing (SCA) [10], [11] and the digital annealing (DA) [25], [26] by allocating N accumulators to N spins. Compared with the computation in series, a parallel ΔE_i calculation can achieve a speedup of N times, however at the cost of a large circuit area.

2) The Update of Spins: The states of disconnected spins are updated according to the spin flip probability in each iteration of annealing, which is considered as one Monte Carlo (MC) step. The connected spins cannot be simultaneously updated in SA, so it takes up to N MC steps to update all spins in a fully connected N-spin Ising model.

To realize a parallel spin update, the SCA [10] and the momentum annealing (MA) [27] utilize a two-layer structure that duplicates N spins to $2N$ spins. Each spin in one layer is connected to N spins in the other layer without any connection between the N spins in the same layer. Specifically, the interaction between a spin and its replica must be sufficiently large at the end of the annealing to ensure that these two spins are in the same state.

3) The Random Flip of Spins: Random flips of spins help an Ising model to escape from the local minimum energy. There are three main approaches for optimizing this process: increasing the flip probability, simplifying the function to generate the flip probability and generating random numbers for a higher hardware efficiency.

A dynamic offset is applied in the calculation of the energy variation to increase the probability of the spin flip in the DA [25], [26]. The energy variation increases with a growing offset over MC steps when there is no spin-state transition. This method can increase the probability of the spin flip and then improve the chance of jumping out of the local minima.

The function to generate the spin flip probability, such as the sigmoid function, is difficult to implement on circuits. Hence, it is approximated by a piece-wise linear function in the SCA [10] to reduce the hardware cost.

Moreover, due to the high hardware cost of random number generators, randomness is introduced by using the variability in the minimum operating voltage of a static random access memory (SRAM) cell in CMOS annealing [9].

B. Simulated Bifurcation (SB)

A quantum mechanical bifurcation machine is based on the adiabatic optimization of the Kerr-nonlinear parametric oscillators (KPOs), where two branches of the bifurcation indicate two states of a spin [22]. Simulated bifurcation (SB) has been proposed to numerically model the adiabatic evolution of a classical nonlinear Hamiltonian system [23].

The SB, also referred to as adiabatic SB (aSB), essentially searches for a solution by solving a pair of differential equations related to the positions and momenta of KPOs with respect to time. The semi-implicit Euler method is used as an efficient integrator to solve the differential equations [23]. The sign of the position indicates the state of the spin at the end of the search. The SB can efficiently solve CO problems by the simultaneous update of positions and momenta.

The ballistic SB (bSB) and the discrete SB (dSB) have been developed to restrain errors introduced by the use of continuous variables (for positions) to represent the discrete spin states [28]. The bSB introduces inelastic walls as hard limits to force positions into $[-1, 1]$. The dSB further discretizes positions to -1 or 1 in the sum of product operations. Compared with aSB, the bSB can quickly find a better approximate solution and the dSB can obtain a solution with a higher quality when solving large-scale problems.

A fully connected Ising machine incurs unnecessary overhead to traverse the zero entries in coefficients when solving sparsely connected Ising problems [23]. An edge-centric SB only traverses non-zero connections by treating the SB as an iterative algorithm over edges instead of vertices when solving a graph-based CO problem [29]. Moreover, various evolution strategies for the position and different dynamic configurations of the time step used in the integrator have been discussed in [30] and [31] to efficiently solve the TSP by using bSB.

C. Simulated Quantum Annealing (SQA)

An Ising machine using quantum annealing can solve CO problems significantly faster than a classical Ising machine. However, it is impractical to solve large-scale problems due to the limited number of quantum bits available in a quantum Ising machine [3]. Thus, simulated quantum annealing (SQA) imitates the quantum annealing dynamics on classical computers based on a quantum MC method [19].

The transverse field in SQA plays a similar role as the temperature in SA to control the probability of transition between the states of spins. By decreasing the transverse field from a very large value to zero, the Ising model tends to converge to the ground state with the lowest energy. Multiple replicas of spins called Trotters are used to map a quantum Ising model into a classical Ising model. In each Trotter, the spins flip in three steps. First, the energy variation is computed in series for each spin. Then, the transverse field decreases with MC steps. Finally, the spin states are updated in a probabilistic manner determined by the energy variations.

The spins can be updated in parallel by restricting the number of interactions between spins for sparsely connected Ising models [19], [32]. However, it is challenging to map real-world problems with dense connectivity into a sparsely connected Ising model. Thus, parallel processing is essential for a fully connected Ising model using SQA. The spin flips in a Trotter depend on the spin states in the two neighboring Trotters. Hence, temporal parallelism is carefully scheduled without violating the data dependency [33]–[36]. In this way, multiple independent spins flip at the same MC step. Moreover, spatial parallelism is applied to compute the energy variations using multiple threads in a thread block for a spin [33], [34].

D. Coherent Ising Machine Simulation (CIMS)

A CIM has shown a significant promise over existing quantum annealers due to its ability to run at room temperature, especially for solving densely connected Ising problems [37]. However, the physical implementation of a CIM requires hundreds of meters of optical fibres. Thus, innovative simulation algorithms are inspired to improve space utilization and computational speed.

The noisy mean-field annealing (NMFA) algorithm emulates the operation of the CIM [20] by utilizing traditional mean-field annealing (MFA) [38]. Gaussian noise is introduced to simulate the quantum noise associated with the optical parameters. The spin states in the NMFA are represented by a set of continuous variables within a range of $(-1, +1)$, which is determined from the tanh activation function. All variables are simultaneously updated based on their previous values at the end of each MC step. This allows the NMFA to achieve massive parallelism and speed up the annealing process. After completing all MC steps, the final spin states are determined by the signs of variables.

As another efficient simulated CIM algorithm, the SimCIM [21] models the interference of the optical parametric oscillators (OPOs) within the CIM. SimCIM treats each spin as a continuous variable within $[-1, 1]$ and truncates other values. It updates the spin states by simulating the physical behavior of a CIM. Similar to the NMFA, SimCIM can achieve massive parallelism. At the end of the search, the spin states are given by the signs of the corresponding variables.

IV. Classical Ising Machines

Classical Ising machines using the simulation algorithms have been developed with various topologies, as shown in Table I. In the King topology, each spin has connections with its neighbors in all eight directions. The 3D topology is based on connected 2D lattices, where the neighbor spins are connected to each other in a lattice. In the complete topology, all spins are fully connected. A DA-based Ising machine implemented on an FPGA can solve problems with up to 8192 spins [25]. A two-layer structure has been realized in SCA [10] and MA [27]. A CMOS annealing-based Ising machine constructed by 9 ASIC chips can scale up to $144k$ spins [39]. The SB-based Ising machines implemented using 16 clustered GPUs employ continuous variables in 32-bit floating-point numbers [28]. An FPGA implementation with a 16-bit representation of continuous variables has been realized in [40], while a multi FPGA SB machine achieves a higher scalability [41]. For the SQA algorithm, a GPU implementation can solve problems with up to 32768 spins [33]. Especially, the connectivity coefficients can be implemented by using 32-bit floating-point numbers. Both NMFA and SimCIM have been implemented on GPUs, but only an Ising problem with $2k$ spins and 2-bit connectivity coefficients was tested [20], [21].

V. Summary, Challenges and Prospects

In this paper, recently developed simulation algorithms are reviewed in a classified manner for classical Ising machines. Various strategies to reduce the search time and decrease the computational complexity are investigated based on the characteristics of the algorithms.

SA emulates the thermal annealing in metallurgy to realize the convergence of Hamiltonian. However, it suffers from the sequential update of spin states and the high implementation cost of introducing randomness. Therefore, various methods have been developed for improving efficiency, including calculating the variation of energy in parallel, applying a dynamic offset on the variation, simplifying the spin flip probability function, and introducing randomness by using the SRAM. In particular, a two-layer structure using duplicated spins achieves a massive parallel processing.

As quantum mechanics-inspired algorithms, SB emulates the adiabatic evolution in a classical nonlinear Hamiltonian system exhibiting bifurcation, whereas SQA describes the quantum tunneling phenomena with a transverse field. SB allows the simultaneous update of positions and momenta, but errors arise due to the use of continuous variables (for the oscillator positions) to represent the spin states. Thus, two variant SB algorithms introduce inelastic walls or discretization to improve the solution quality and speed up the convergence of Hamiltonian. To alleviate the time explosion problem in SQA, temporal and spatial parallelism are, respectively, applied to the spin update and the computation of energy variations.

TABLE I: Classical Ising Machines

Algorithm		Topology	Platform	Precision	Size
SA	DA [25]	Complete	ASIC	-	8192
	DA [26]	Complete	FPGA	16 bits	1024
	MA [27]	Complete	4 GPUs	10 bits	$10k$
	SCA [10]	Complete	ASIC	5 bits	512
	CMOS annealing [9]	3D	ASIC	2 bits	$20k$
	CMOS annealing [12]	King	3 ASICs	3 bits	$60k$
	CMOS annealing [39]	King	9 ASICs	5 bits	$144k$
SB	aSB, bSB, dSB [28]	Complete	16 GPUs	10 bits	$100k$
	aSB [40]	Complete	FPGA	1 bit	4096
	aSB [41]	Complete	8 FPGAs	-	16384
SQA	SQA [19]	King	FPGA	8 bits	9216
	SQA [34]	Complete	2 FPGAs	32* bits	32768
	SQA [33]	Complete	GPU	32* bits	32768
CIMS	NMFA [20]	Complete	GPU	-	$\geq 2k$
	SimCIM [21]	Complete	GPU	-	$\geq 2k$

Precision: the bit-width of connectivity coefficients (default: fixed point; *: floating point); Size: the maximum number of spins.

The high performance of CIMs motivates the development of simulation algorithms such as the NMFA and SimCIM to improve computational speed and hardware efficiency. Even though both algorithms are similar to some extent, they originated from a different perspective. The NMFA algorithm emulates the convergence of a CIM, while the SimCIM algorithm models the interference of the OPO within the CIM. The Ising machines using the NMFA and SimCIM algorithms perform similarly to or even better than a CIM.

A major challenge for SA and SQA is to efficiently realize the parallel spin update for speeding up annealing in fully connected Ising models. The duplicated structure developed from SA mitigates the sequential spin-update problems but it suffers from a high hardware cost due to the use of redundant spins. For SQA, temporal parallelism and spatial parallelism can only realize partial parallelization because each spin update is highly dependent on multiple other spins. SB and CIM simulation algorithms utilize continuous variables to represent the discrete states of spins. However, the computation with continuous variables is more hardware-consuming compared to the computation with discrete variables. Therefore, the relatively high computational complexity and the resulting hardware implementation are key obstacles for both methods. Finally, due to the inherent complex mechanism of SB, it is difficult to find the optimal settings of parameters to accelerate the convergence of Hamiltonian.

Hence, efficient parallel processing for accelerating the update of spin states and reducing computational complexity for improving the scalability of classical Ising machines, as well as how to improve the probability of escaping from the local energy minima, are worthy of further investigation.

Acknowledgment

This work was partly supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada (Project Number: RES0048688). T. Zhang is supported by a Ph.D. scholarship from the China Scholarship Council (CSC).

References

[1] A. Lucas, “Ising formulations of many NP problems,” FUJITSU Sci Tech J., vol. 5, no. 5, 2014.

[2] B. A. Cipra, “An introduction to the Ising model,” The American Mathematical Monthly, vol. 94, no. 10, pp. 937-959, 1987.

[3] M. W. Johnson, M. H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson et al., “Quantum annealing with manufactured spins,” Nature, vol. 473, no. 7346, pp. 194-198, 2011.

[4] A. A. Houck, H. E. Türeci, and J. Koch, “On-chip quantum simulation with superconducting circuits,” Nature Physics, vol. 8, no. 4, pp. 292-299, 2012.

[5] Z. Wang, A. Marandi, K. Wen, R. L. Byer, and Y. Yamamoto, “Coherent Ising machine based on degenerate optical parametric oscillators,” Physical Review A, vol. 88, no. 6, p. 063853, 2013.

[6] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo et al., “A coherent Ising machine for 2000-node optimization problems,” Science, vol. 354, no. 6312, pp. 603-606, 2016.

[7] M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura, and H. G. Katzgraber, “Physics-inspired optimization for quadratic unconstrained problems using a digital annealer,” Frontiers in Physics, vol. 7, p. 48, 2019.

[8] T. Wang, L. Wu, P. Nobel, and J. Roychowdhury, “Solving combinatorial optimisation problems using oscillator based Ising machines,” Natural Computing, pp. 1-20, 2021.

[9] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, “A 20k-spin Ising chip to solve combinatorial optimization problems with CMOS annealing,” IEEE Journal of Solid-State Circuits, vol. 51, no. 1, pp. 303-309, 2015.

[10] K. Yamamoto, K. Ando, N. Mertig, T. Takemoto, M. Yamaoka, H. Teramoto 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,” IEEE International Solid-State Circuits Conference-(ISSCC), pp. 138-140, 2020.

[11] K. Yamamoto, K. Kawamura, K. Ando, N. Mertig, T. Takemoto, M. Yamaoka et al., “STATICA: A 512-spin 0.25 M-weight annealing processor with an all-spin-updates-at-once architecture for combinatorial optimization with complete spin–spin interactions,” IEEE Journal of Solid-State Circuits, vol. 56, no. 1, pp. 165-178, 2020.

[12] T. Takemoto, M. Hayashi, C. Yoshimura, M. Yamaoka, “A 2× 30k-spin multi-chip scalable CMOS annealing processor based on a processing-in-memory approach for solving large-scale combinatorial optimization problems,” IEEE Journal of Solid-State Circuits, vol. 55, no. 1, pp. 145-156, 2019.

[13] F. Glover, G. Kochenberger, and Y. Du, “A tutorial on formulating and using QUBO models,” arXiv preprint arXiv:1811.11538, 2018.

[14] D. Oku, M. Tawada, S. Tanaka, and N. Togawa, “How to reduce the bit-width of an Ising model by adding auxiliary spins,” IEEE Transactions on Computers, vol. 71, no. 1, pp. 223-234, 2020.

[15] T. Shirai, S. Tanaka, and N. Togawa, “Guiding principle for minor-embedding in simulated-annealing-based Ising machines,” IEEE Access, vol. 8, pp. 210490-210502, 2020.

[16] D. Oku, K. Terada, M. Hayashi, M. Yamaoka, S. Tanaka, and N. Togawa, “A fully-connected Ising model embedding method and its evaluation for CMOS annealing machines,” IEICE Transactions on Information and Systems, vol. 102, no. 9, pp. 1696-1706, 2019.

[17] K. Tatsumura, A. R. Dixon, and H. Goto, “Performance comparison of typical binary-integer encodings in an Ising machine,” IEEE Access, vol. 9, pp. 81032-81039, 2021.

[18] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” Science, vol. 220, no. 4598, pp. 671-680, 1983.

[19] T. Okuyama, M. Hayashi, and M. Yamaoka, “An Ising computer based on simulated quantum annealing by path integral Monte Carlo method,” in IEEE International Conference on Rebooting Computing (ICRC), pp. 1-6, 2017.

[20] A. D. King, W. Bernoudy, J. King, A. J. Berkley, and T. Lanting, “Emulating the coherent Ising machine with a mean-field algorithm,” arXiv preprint arXiv:1806.08422, 2018.

[21] E. S. Tiunov, A. E. Ulanov, and A. I. Lvovsky, “Annealing by simulating the coherent Ising machine,” Optics Express, vol. 27, no. 7, pp. 10288-10295, 2019.

[22] H. Goto, “Bifurcation-based adiabatic quantum computation with a nonlinear oscillator network,” Scientific Reports, vol. 6, no. 1, pp. 1-8, 2016.

[23] H. Goto, T. Kosuke, and R. D. Alexander, “Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems,” Science Advances, vol. 5, no. 4, 2019.

[24] S. Kanamaru, K. Kawamura, S. Tanaka, Y. Tomita, H. Matsuoka, K. Kawamura, and N. Togawa, “Mapping constrained slot-placement problems to Ising models and its evaluations by an Ising machine,” IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin), pp. 221-226, 2019.

[25] S. Matsubara, M. Takatsu, T. Miyazawa, T. Shibasaki, Y. Watanabe, K. Takemoto et al., “Digital annealer for high-speed solving of combinatorial optimization problems and its applications,” in the 25th Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 667-672, 2020.

[26] S. Tsukamoto, M. Takatsu, S. Matsubara, and H. Tamura, “An accelerator architecture for combinatorial optimization problems,” Fujitsu Sci. Tech. J, vol. 53, no. 5, pp. 8-13, 2017.

[27] T. Okuyama, T. Sonobe, K. Kawarabayashi, and M. Yamaoka, “Binary optimization by momentum annealing,” Physical Review E, vol. 100, no. 1, p. 012111, 2019.

[28] H. Goto, E. Kotaro, S. Masaru, S. Yoshisato, T. Kanao, Y. Hamakawa et al.,“High-performance combinatorial optimization based on classical mechanics,” Science Advances, vol. 7, no. 6, 2021.

[29] Y. Zou and M. Lin, “Massively simulating adiabatic bifurcations with FPGA to solve combinatorial optimization,” in Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, pp. 65-75, 2020.

[30] T. Zhang, Q. Tao, and J. Han, “Solving traveling salesman problems using Ising models with simulated bifurcation,” in the 18th International SoC Design Conference (ISOCC), pp. 288-289, 2021.

[31] T. Zhang and J. Han, “Efficient traveling salesman problem solvers using the Ising model with simulated bifurcation,” in Design, Automation and Test in Europe Conference and Exhibition (DATE), 2022.

[32] H. M. Waidyasooriya, Y. Araki, and M. Hariyama, “Accelerator architecture for simulated quantum annealing based on resource-utilization-aware scheduling and its implementation using opencl,” in International Symposium on Intelligent Signal Processing and Communication Systems (ISPAC), pp. 336–340, 2018.

[33] H. M. Waidyasooriya, and M. Hariyama, “A GPU-based quantum annealing simulator for fully-connected ising models utilizing spatial and temporal parallelism,” IEEE Access, vol. 8, pp. 67929-67939, 2020.

[34] H. M. Waidyasooriya, and M. Hariyama, “Highly-parallel FPGA accelerator for simulated quantum annealing,” IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 4, pp. 2019-2029, 2019.

[35] H. M. Waidyasooriya, M. Hariyama, M. J. Miyama, and M. Ohzeki, “OpenCL-based design of an FPGA accelerator for quantum annealing simulation,” J. Supercomput., vol. 75, no. 8, pp. 5019–5039, 2019.

[36] C. Y. Liu, H. M. Waidyasooriya, and M. Hariyama, “Data-transfer bottleneck-less architecture for FPGA-based quantum annealing simulation,” in Proceeding of the 7th International Symposium on Computing and Networking, pp. 164–170, 2019.

[37] R. Hamerly, T. Inagaki, P. L. McMahon, D. Venturelli, A. Marandi, T. Onodera et al., ”Experimental investigation of performance differences between coherent Ising machines and a quantum annealer,” Science Advances, vol. 5, no. 5, p. eaau0823, 2019.

[38] G. Bilbro, R. Mann, T. K. Miller, W. E. Snyder, D. E. Van den Bout, and M. White, “Optimization by mean field annealing,” in Advances in Neural Information Processing Systems, pp. 91-98. 1989.

[39] T. Takemoto, K. Yamamoto, C. Yoshimura, M. Hayashi, M. Tada, H. Saito et al., “4.6 A 144Kb annealing system composed of 9× 16Kb annealing processor chips with scalable chip-to-chip connections for large-scale combinatorial optimization problems,” IEEE International Solid-State Circuits Conference (ISSCC), vol. 64, pp. 64-66, 2021.

[40] K. Tatsumura, A. R. Dixon, and H. Goto, “FPGA-based simulated bifurcation machine,” in the 29th International Conference on Field Programmable Logic and Applications (FPL), pp. 59-66, 2019.

[41] K. Tatsumura, M. Yamasaki, and H. Goto, “Scaling out Ising machines using a multi-chip architecture for simulated bifurcation,” Nature Electronics, vol. 4, no. 3, pp. 59-66, 2019.