PAPER Special Section on Forefront Computing

A Fully-Parallel Annealing Algorithm with Autonomous Pinning Effect Control for Various Combinatorial Optimization Problems*

Daiki OKONOGI[†a)], **Satoru JIMBO**[†], ***Nonmembers***, **Kota ANDO**[††], **Thiem Van CHU**[†], **Jaehoon YU**[†], **Masato MOTOMURA**[†b)], ***and*** **Kazushi KAWAMURA**[†c)], ***Members***

SUMMARY Annealing computation has recently attracted attention as it can efficiently solve combinatorial optimization problems using an Ising spin-glass model. Stochastic cellular automata annealing (SCA) is a promising algorithm that can realize fast spin-update by utilizing its parallel computing capability. However, in SCA, pinning effect control to suppress the spin-flip probability is essential, making escaping from local minima more difficult than serial spin-update algorithms, depending on the problem. This paper proposes a novel approach called APC-SCA (Autonomous Pinning effect Control SCA), where the pinning effect can be controlled autonomously by focusing on individual spin-flip. The evaluation results using max-cut, N-queen, and traveling salesman problems demonstrate that APC-SCA can obtain better solutions than the original SCA that uses pinning effect control pre-optimized by a grid search. Especially in solving traveling salesman problems, we confirm that the tour distance obtained by APC-SCA is up to 56.3% closer to the best-known compared to the conventional approach.

key words: *combinatorial optimization, Ising model, annealing processor, parallel annealing algorithm, stochastic cellular automata*

1. Introduction

Solving combinatorial optimization problems (COPs) is difficult, but it is an essential task in various industrial fields like financial service, physical distribution, and traffic management [1]–[4]. Annealing computation has attracted attention in recent years because it is expected to solve various COPs efficiently and uniformly. This computation technique utilizes an Ising model to represent different COPs in a unified manner and obtains the optimal solutions by searching for the ground state [5].

There are several annealing algorithms and their specialized processors. One of the representative algorithms is quantum annealing [6], and D-Wave [7] adopts it as the operating principle. However, D-Wave takes a huge cooling cost, which has been making its practical use difficult so far. Simulated annealing (SA) [8] is another algorithm, and some extended algorithms [9]–[11] have also been proposed to improve the spin-update efficiency. They include digital annealing [9] and stochastic cellular automata annealing (SCA) [11], the operating principles of digital annealer [9] and STATICA [12], respectively. Since these *solid-state* processors can operate at room temperature, they do not require a large-scale cooling system.

SCA is a promising algorithm that updates all spins in parallel and realizes highly efficient ground-state search. This parallel spin-update capability is ensured by self-interaction, named pinning parameter, to suppress the flip probability of each spin. However, this parameter makes the spin configuration easily trapped in local minima, depending on the COP to be solved.

Some prior works [13], [14] have indicated that dynamic parameter control is effective in escaping the spin configuration from local minima. The method in [13] dynamically gives an energy offset to forcibly cause some spin-flip, while the method in [14] introduces dynamic control of pseudo temperature. In this work, by examining dynamic control for the pinning parameter in fully-parallel annealing, we aim to realize that the solution search is rarely trapped in local minima.

This paper proposes APC-SCA (Autonomous Pinning effect Control SCA), a novel algorithm that helps the spin configuration escape from local minima, and evaluates its performance using max-cut, N-queen, and traveling salesman problems. APC-SCA autonomously controls pinning parameter in the annealing computation. We realize this autonomous control by focusing on the flip of each spin and adjusting its pinning parameter adaptively and individually. The evaluation results demonstrate that APC-SCA can obtain better solutions than the original SCA with an optimal pinning parameter scheduling found by a grid search.

The main contributions and findings of this paper are as follows:

- APC-SCA overcomes the weakness of SCA, i.e., the spin configuration is easily trapped in local minima, depending on the COP to be solved.
- The evaluation results show that APC-SCA can obtain superior solutions to the original SCA (where the pinning parameters are pre-optimized for individual problems) in all problems. Especially in solving traveling salesman problems, APC-SCA can improve the tour distance by up to 56.3%.

The rest of the paper is organized as follows. In Sect. 2,

Manuscript received December 16, 2022.
Manuscript revised April 10, 2023.
Manuscript publicized September 19, 2023.

[†]The authors are with Tokyo Institute of Technology, Yokohama-shi, 226–8502 Japan.
[††]The author is with Hokkaido University, Sapporo-shi, 060–0814 Japan.

*This paper is an extended version of the paper [15], which includes some additional discussions and evaluation results.

a) E-mail: okonogi.daiki@artic.iir.titech.ac.jp
b) E-mail: motomura@artic.iir.titech.ac.jp
c) E-mail: kawamura@artic.iir.titech.ac.jp

DOI: 10.1587/transinf.2023PAP0003

we clarify the problem about pinning parameter control in the annealing computation. In Sect. 3, we propose a novel algorithm, APC-SCA, to overcome the problem. Evaluation results are presented in Sect. 4. The paper is finally wrapped up in Sect. 5.

2. Preliminaries

2.1 Ising Model and Annealing Computation

The Ising model consists of N spins $\boldsymbol{\sigma} = \{\sigma_i\}_{1\le i\le N}$, spin-spin interactions $\boldsymbol{J} = \{J_{ij}\}_{1\le i,j\le N}$, and external magnetic fields $\boldsymbol{h} = \{h_i\}_{1\le i\le N}$. Each spin $\sigma_i \in \boldsymbol{\sigma}$ can be either of two states, $+1$ and -1. Based on the spin states, the system energy is determined by:

$$H(\boldsymbol{\sigma}) = -\frac{1}{2}\sum_{i\neq j} J_{ij}\sigma_i\sigma_j - \sum_i h_i\sigma_i. \tag{1}$$

A spin configuration that gives the minimum energy is called the ground state. When converting a COP into the Ising model, we construct the energy function Eq. (1) so that the ground state encodes the optimal solution. As demonstrated in [16]–[18], various COPs can be converted into the Ising model.

The annealing computation is a heuristic approach for the ground-state search of an Ising model. SA is one of the most well-known algorithms, which updates spins based on the Markov Chain Monte Carlo method [8]. There also exist some extended algorithms [9]–[11]. These SA-based algorithms control spin-flip probabilities using pseudo temperature T for preventing the spin configuration from being trapped in local minima. The temperature is initially set to a large value for global search, and it is gradually decreased to a small value for shifting to local search.

2.2 SCA Algorithm and Its Practical Problem

SCA is a fully parallel annealing algorithm, enabling fast spin-update [11]. By utilizing this parallel spin-update capability, it is expected to improve the efficiency of the ground-state search. In SCA, the current spin states ($\boldsymbol{\sigma}$) are updated in parallel, and the next spin states ($\boldsymbol{\tau}$) are generated simultaneously. The algorithm of SCA is shown in **Algorithm 1**, which aims at minimizing the following energy function:

$$\begin{aligned} H(\boldsymbol{\sigma},\boldsymbol{\tau}) = &-\frac{1}{2}\sum_{i\neq j} J_{ij}\sigma_i\tau_j - \frac{1}{2}\sum_i h_i(\sigma_i+\tau_i) \\ &+ q\sum_i(1-\sigma_i\tau_i), \end{aligned} \tag{2}$$

where $q > 0$ is a pinning parameter. The third term indicates that the energy increases by $2q$ when a spin state σ_i changes after the update ($\tau_i \neq \sigma_i$). Therefore, q denotes a degree of the *pinning* effect that keeps the current spin states from being changed. In the previous study [11], it has been proved that SCA can correctly reach the ground state when using $q \ge \lambda/2$, where λ is the maximum eigenvalue of $-\boldsymbol{J}$.

Algorithm 1 SCA: Stochastic Cellular Automata annealing.

Input:
- # of spins: N
- initial spin states: $\boldsymbol{\sigma}(1) = \{\sigma_i(1)\}_{1\le i\le N}$
- spin-spin interactions: $\boldsymbol{J}$, external magnetic fields: $\boldsymbol{h}$
- # of Monte Carlo (MC) steps: S
- pseudo temperature scheduling: $T(s)$
- pinning parameter scheduling: $q(s)$

Output:
- optimized spin states: $\mathrm{argmin}_{1\le s\le S+1}\{H(\boldsymbol{\sigma}(s))\}$

```
for s = 1 to S do
    for i = 1 to N do
        Calculate local field
            ▷ h̃_i = Σ_j J_ij σ_j(s) + h_i
        Calculate flip probability
            ▷ p_i = sigmoid(−(h̃_i σ_i(s) + q(s))/T(s))
        Generate a random number rand = [0, 1)
        if p_i > rand then
            τ_i = −σ_i(s)
        else
            τ_i = σ_i(s)
    for i = 1 to N do
        σ_i(s + 1) = τ_i
```

In solving a practical COP by an annealing algorithm, we usually put a realistic limit on the number of MC steps (S). The exponential temperature scheduling $T_{\exp(s)}$ is suitable for this situation because it allocates high, medium, and low temperatures in a well-balanced manner:

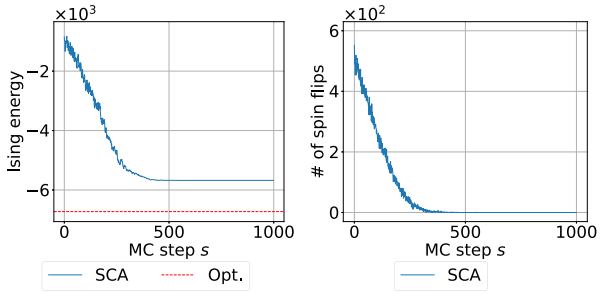
$$T_{\exp(s)} = T_{\mathrm{init}}\cdot {r_T}^{s-1},\ r_T = \left(\frac{T_{\mathrm{final}}}{T_{\mathrm{init}}}\right)^{\frac{1}{S-1}}, \tag{3}$$

where T_{init} and T_{final} are the initial and final temperatures, respectively. In this case, there is no mathematical guarantee to reach a ground state even when using $q = \lambda/2$ in SCA. We should note that the heuristic $T_{\exp(s)}$ scheduling has been commonly used in SA since it empirically gives good solutions even though there is no mathematical guarantee either: we follow the same practice.

Figure 1 visualizes the behavior of SCA when solving a well-known max-cut benchmark problem G22 [19] assuming $S = 1000$, $q(s) = \lambda/2$, and $T_{\exp(s)}$ with $(T_{\mathrm{init}}, T_{\mathrm{final}}) = (10, 0.1)$. This figure shows that the spin configuration is trapped in a local minimum, and the ground-state search does not progress in the latter half of the annealing computation.

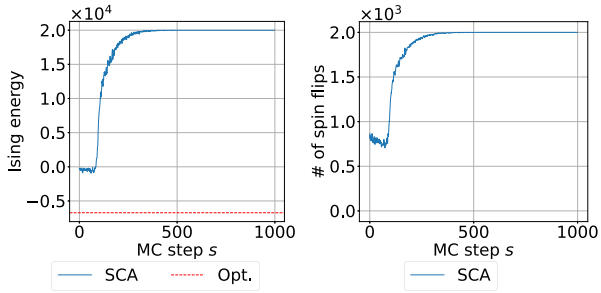
2.3 Related Work

To make escaping from local minima easy, some methods to dynamically control the spin-flip probability have been developed [13], [14]. Digital Annealer (DA) [13] is a solid-state annealing processor that updates a single spin per MC step. In DA, a dynamic offset E_{offset} (≥ 0) was introduced to prevent the spin configuration from being fixed for a long time. In the DA's spin-update procedure, E_{offset} continues to increase while there is no spin-flip candidate and is reset to zero if there is even one candidate. Since E_{offset} tem-



(a) Ising energy. **(b)** The number of spin flips.

Fig. 1 The behavior of SCA with $q(s) = \lambda/2$ when solving a max-cut problem G22 [19]. We here assumed $S = 1000$ and $T_{\exp(s)}$ with $(T_{\text{init}}, T_{\text{final}}) = (10, 0.1)$. Spin flips no longer appear in the latter half of the annealing computation ($s \geq 500$), indicating that the spin configuration cannot escape from a local minimum.



(a) Ising energy. **(b)** The number of spin flips.

Fig. 2 Illustration of the Ising energy transition and the spin-flip number transition when running G22 with $q = \lambda/6$. The Ising energy scale-out and the spin-flip number equal the spin number.

porarily increases the spin-flip probability, the spin configuration is forced to change and thus can be escaped from local minima. According to the paper [13], this technique approximately equivalent to multiplying the common factor $\exp(\beta \cdot E_{\text{offset}})$ by the flip probability of each spin, where β is the inverse temperature $1/T(s)$.

Tao et al. proposed the improved parallel annealing (IPA) [14] derived from momentum annealing (MA) [10]. IPA applies the dynamic offset to the pseudo temperature control for efficiently solving traveling salesman problems (TSPs). We aim to prevent the spin configuration from being trapped in local minima by utilizing dynamic parameter control. Our proposed method extends dynamic offset to pinning parameter with assigning it for each spin σ_i as q_i.

3. APC-SCA: SCA with Autonomous Pinning Parameter Control

3.1 Motivation

DA and SCA are single and parallel spin-update algorithms, respectively, so their flip probabilities are:

$$p_i^{\text{DA}} = \text{sigmoid}\left(-\frac{\tilde{h}_i\sigma_i}{T(s)}\right) \tag{4}$$

and

$$p_i^{\text{SCA}} = \text{sigmoid}\left(-\frac{\tilde{h}_i\sigma_i + q(s)}{T(s)}\right), \tag{5}$$

where $q(s)$ is a pinning parameter and should be set to $\lambda/2$ for keeping the reachability to the ground state. However, this parameter strongly prevents the spin state from escaping from local minima. Therefore, obtaining the ground state using SCA is more challenging than DA.

As an approach to help the spin state escape from local minima, DA utilizes a dynamic offset E_{offset} that continues to increase as long as the spin state remains unchanged and is reset to zero once the spin state changes. In the spin-update procedure, E_{offset} is updated as follows:

$$E_{\text{offset}}(s+1) = \begin{cases} E_{\text{offset}}(s) + E_{\text{inc}} & \text{if } \boldsymbol{\sigma}(s+1) = \boldsymbol{\sigma}(s) \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

where E_{inc} is a constant value that shows the increment of E_{offset}. Since the offset value varies across MC steps, Eq. (4) is re-written as follows:

$$p_i^{\text{DA}} = \text{sigmoid}\left(-\frac{\tilde{h}_i\sigma_i}{T(s)} + E_{\text{offset}}(s)\right). \tag{7}$$

Similarly, we can use this approach in SCA and obtain the following flip probability from Eq. (5):

$$p_i^{\text{SCA}} = \text{sigmoid}\left(-\frac{\tilde{h}_i\sigma_i + q(s)}{T(s)} + E_{\text{offset}}(s)\right) \tag{8}$$

$$= \text{sigmoid}\left(-\frac{\tilde{h}_i\sigma_i + q(s) - E_{\text{offset}}(s)T(s)}{T(s)}\right) \tag{9}$$

In Eq. (9), $q(s)$, $E_{\text{offset}}(s)$, and $T(s)$ are all functions of s so that SCA can incorporate the dynamic offset into the pinning parameter. As a result, Eq. (9) is equivalent to Eq. (5). SCA can naturally escape the spin state from local minima by appropriately controlling the pinning parameter $q(s)$. The next question is how we should control $q(s)$ to maximize the capability.

To devise a strategy for dynamic control of the pinning parameter, we first observe the behavior of SCA by using a constant q: large or small. In the case of a large q ($= \lambda/2$), the spin configuration is easily trapped in local minima because the pinning effect enforces spin configuration to keep the current states in the latter half of annealing (see Fig. 1). In the case of a small q ($= \lambda/6$), the spin configuration goes toward high Ising energy because the pinning effect becomes weak and the number of flips excessively increases (see Fig. 2). These results indicate that we should use a smaller q if the number of flips is too few and a larger q if it is too many. Now the question is how to judge, in the annealing computation, that the number of flips is few (many). However, it is not realistic to estimate, before or during annealing, how many spins should flip. For this reason, we focus only on individual spins and aim to control their pinning parameters independently. We let the i-th spin

have an exclusive pinning parameter q_i. The idea of this *independent* control leads us to a strategy for autonomous control of each pinning parameter: decrease the value of q_i if the i-th spin does not flip in the previous MC step, and vice versa.

3.2 Proposed Algorithm

In this section, we propose a novel fully-parallel annealing algorithm, APC-SCA, based on the strategy in Sect. 3.1. This algorithm is constructed by integrating autonomous and individual pinning parameter control into SCA. **Algorithm 2** shows the algorithm of APC-SCA. Note that the difference between **Algorithm 1** and this algorithm is highlighted in blue. Unlike the original SCA, we need to prepare N pinning parameters $\{q_i\}_{1\le i\le N}$ and initialize each of them to $\lambda/2$ as shown in Lines 1–2. In each MC step, APC-SCA updates the i-th spin in the same manner as SCA except that the individual pinning parameter q_i is used for calculating the flip probability p_i in Line 6. Every MC step updates q_i according to whether the i-th spin flips or not. This update is controlled by two additional parameters, $r_q < 1$ and $q_{\text{limit}} \ge 0$, which are constant values showing the decreasing rate and the lower limit, respectively. The value of q_i becomes $\lambda/2$ after the i-th spin flips (Line 10), while it decreases to $q_i \cdot r_q$ (or remains at q_{limit}) if it does not flip (Line 13).

Algorithm 2 APC-SCA: Autonomous Pinning effect Control SCA.

Input:
- # of spins: N
- initial spin states: $\boldsymbol{\sigma}(1) = \{\sigma_i(1)\}_{1\le i\le N}$
- spin-spin interactions: $\boldsymbol{J}$, external magnetic fields: $\boldsymbol{h}$
- # of Monte Carlo (MC) steps: S
- pseudo temperature scheduling: $T(s)$
- decreasing rate of pinning parameter: r_q
- lower limit of pinning parameter: q_{limit}

Output:
- optimized spin states: $\text{argmin}_{1\le s\le S+1}\{H(\boldsymbol{\sigma}(s))\}$

1: **for** $i = 1$ to N **do**
2: $\quad q_i(1) = \lambda/2$
3: **for** $s = 1$ to S **do**
4: $\quad$ **for** $i = 1$ to N **do**
5: $\quad\quad$ Calculate local field
$\quad\quad\quad \triangleright\ \tilde{h}_i = \sum_j J_{ij}\sigma_j(s) + h_i$
6: $\quad\quad$ Calculate flip probability
$\quad\quad\quad \triangleright\ p_i = \text{sigmoid}(-(\tilde{h}_i\sigma_i(s) + q_i(s))/T(s))$
7: $\quad\quad$ Generate a random number rand $= [0, 1)$
8: $\quad\quad$ **if** $p_i >$ rand **then**
9: $\quad\quad\quad \tau_i = -\sigma_i(s)$
10: $\quad\quad\quad q_i(s+1) = \lambda/2$
11: $\quad\quad$ **else**
12: $\quad\quad\quad \tau_i = \sigma_i(s)$
13: $\quad\quad\quad q_i(s+1) = \max\{q_i(s)\cdot r_q,\ q_{\text{limit}}\}$
14: $\quad$ **for** $i = 1$ to N **do**
15: $\quad\quad \sigma_i(s+1) = \tau_i$

3.3 Applicability of Autonomous Pinning Effect Control to Other Annealing Algorithms

Our autonomous pinning effect control, proposed and introduced into SCA in Sect. 3.2, can be applied to other parallel annealing algorithms. To realize parallel annealing extended from SA, introducing self-interaction like q in SCA is essential for converging the spin configuration to the ground state. Momentum annealing (MA) [10] and restricted Boltzmann machine (RBM) [20] are other parallel annealing algorithms on an Ising model and have self-interactions denoted as w_i and w_{ii}, respectively. We expect these self-interactions to be controlled autonomously using the same strategy applied to SCA.

We here try to apply autonomous pinning effect control to w_i in MA and propose APC-MA. Note that we will evaluate APC-MA in Sect. 4.4. The main differences between MA and SCA are threefold: 1) temperature scheduling, 2) pinning effect calculation, and 3) dropout.

1) Temperature scheduling: MA uses the logarithmic temperature scheduling given by:

$$T(s) = \frac{1}{\beta_0 \ln(1+s)}, \tag{10}$$

where β_0 is the inverse of the initial temperature.

2) Pinning effect calculation: SCA originally applies a uniform pinning parameter q to all spins. In contrast, MA calculates the self-interaction w_i for each spin using the following equation:

$$w_i = \begin{cases} \displaystyle\sum_{\sigma_j\in\boldsymbol{\sigma}} |J_{ij}| - \frac{1}{2}\sum_{\sigma_j\in\boldsymbol{\sigma}_{\text{A}}} |J_{ij}| & (\sigma_i \in \boldsymbol{\sigma}_{\text{A}}) \\ \dfrac{\lambda}{2} \end{cases}, \tag{11}$$

where λ is the maximum eigenvalue of $-\boldsymbol{J}$, and $\boldsymbol{\sigma}_{\text{A}}$ is an arbitrary subset of $\boldsymbol{\sigma}$. We used $\boldsymbol{\sigma}_{\text{A}} = \left\{\sigma_i \,\middle|\, \lambda \ge \sum_{\sigma_j\in\boldsymbol{\sigma}} \left|J_{ij}\right|\right\}$ in this paper. MA increases the pinning effect by multiplying the scaling factor $c(s)$ to w_i. The scaling factor $c(s)$ varies from a small value to 1 as follows:

$$c(s) = \min\left\{1, \sqrt{\frac{s}{S}}\right\}. \tag{12}$$

3) Dropout: MA utilizes a dropout technique that temporarily removes some pinning effects with a certain probability. The dropout probability p_{drop} gradually decreases to zero as annealing progresses and is calculated by:

$$p_{\text{drop}}(s) = \max\left\{0, 0.5 - \frac{s}{2S}\right\}. \tag{13}$$

In APC-MA, w_i is autonomously controlled in the same manner as APC-SCA using the self-interaction given by Eq. (11) as its initial value. Moreover, the scaling factor and the dropout are applied to the post-calculated w_i.

Table 1 List of Ising models used in this paper.

Name	# of spins (N)	# of interactions	Graph topology
G22	2,000	19,990	Random graph with $J_{ij} = -1$ for all interactions
G30	2,000	19,990	Random graph with $J_{ij} \in \{+1, -1\}$ for all interactions
G32	2,000	4,000	Toroidal graph with $J_{ij} \in \{+1, -1\}$ for all interactions
G35	2,000	11,778	Planar graph with $J_{ij} = -1$ for all interactions
32-queen	1,024	52,576	Two-dimensional graph where negative interactions are given to all pairs of spins that align in the horizontal, vertical, and diagonal directions
burma14	196	5,096	Two-dimensional lattice graph with $J_{ij} \in [-630.5, -4.75]$
ulysses16	256	7,680	Two-dimensional lattice graph with $J_{ij} \in [-1394.5, -13]$
ulysses22	484	20,328	Two-dimensional lattice graph with $J_{ij} \in [-1394.5, -3.5]$

Table 2 Common parameter settings.

Description (Notation)	Max-cut	32-queen	TSP
# of MC steps (S)	1,000	1,000	10,000
# of trials (M)	128	128	128
Initial temperature (T_{init})	10	10	$0.2N \min_{J_{ij} \neq 0} \lvert J_{ij} \rvert$
Final temperature (T_{final})	0.1	0.1	$0.1 \max_{i,j} \lvert J_{ij} \rvert$

4. Experimental Results

4.1 Environments and Settings

We implemented APC-SCA in CUDA C++ and ran it on a GPU, NVIDIA GeForce RTX 2080 Ti. We utilized the method [21] as a baseline implementation of SCA. This implementation uses the single precision floating point number (i.e., float) for storing spin-spin interactions, external magnetic fields, and local fields. In the experiments, we assumed four max-cut problems (G22, G30, G32, and G35) [19], the 32-queen problem (32-queen), and three traveling salesman problems (burma14, ulysses16, and ulysses22) [22]. These problems are briefly described in Table 1. Unless otherwise noted, we assumed the parameter settings in Table 2 to solve each problem.

For comparison, we also solved the eight problems using the original SCA approach, i.e., **Algorithm 1**, that determines the pinning parameter scheduling $q(s)$ in advance. Here we assumed that $q(s)$ is an exponential function, which is represented by:

$$q(s) = q_{\text{init}} \cdot {r_q}^{s-1}, \; r_q = \left(\frac{q_{\text{final}}}{q_{\text{init}}} \right)^{\frac{1}{S-1}}, \tag{14}$$

where q_{init} and q_{final} are the initial and final values of q, which denotes $q(1)$ and $q(S)$, respectively. To optimize $q(s)$ for individual problems, we solved the eight problems while varying q_{init} and q_{final} independently in the range $(0, \lambda/2]$ as follows:

$$q_{\text{init}} \in \left\{ \frac{\lambda}{64} n \right\}_{1 \leq n \leq 32}, \; q_{\text{final}} \in \left\{ \frac{\lambda}{64} n \right\}_{1 \leq n \leq 32}. \tag{15}$$

We here conducted 128 rounds of annealing on each parameter set, $(q_{\text{init}}, q_{\text{final}})$, and then recorded the average Ising energy of 128 optimized solutions. For each problem, we judge that a parameter set that gives the lowest average Ising energy is optimized. We fine-tuned the parameters for each problem through a 32×32 grid search on Eqs. (14) and (15), which resulted in the optimized parameters shown in Table 3. We call this approach "fine-tuned SCA" in the following.

Table 3 Optimized pairs of $(q_{\text{init}}, q_{\text{final}})$ for SCA. This pair is fine-tuned for each problem through a grid search on Eqs. (14) and (15). The obtained energy is the average energy of 128 rounds by using SCA with the optimized $(q_{\text{init}}, q_{\text{final}})$ pair, while the optimal energy is the best-known energy of each benchmark, i.e., the energy of the ground state.

benchmark	q_{init}	q_{final}	Obtained energy	Optimal energy
G22	$\frac{31}{64}\lambda$	$\frac{1}{64}\lambda$	−6,545.5	−6,728.0
G30	$\frac{2}{64}\lambda$	$\frac{2}{64}\lambda$	−6,638.9	−6,826.0
G32	$\frac{26}{64}\lambda$	$\frac{3}{64}\lambda$	−2,726.3	−2,798.0
G35	$\frac{27}{64}\lambda$	$\frac{1}{64}\lambda$	−3,430.1	−3,596.0
32-queen	$\frac{4}{64}\lambda$	$\frac{1}{64}\lambda$	28.7	0.0
burma14	$\frac{22}{64}\lambda$	$\frac{1}{64}\lambda$	5,652.1	3,323.0
ulysses16	$\frac{5}{64}\lambda$	$\frac{1}{64}\lambda$	11,333.2	6,859.0
ulysses22	$\frac{32}{64}\lambda$	$\frac{1}{64}\lambda$	13,773.7	7,013.0

4.2 Max-Cut Problems

We evaluated APC-SCA with four max-cut problems while varying the decreasing rate of the pinning parameter (r_q) from 0.0 to 1.0. Figures 3 (a) to 3 (d) show the average Ising energy used as solution quality. We assumed the lower limit of pinning parameter $q_{\text{limit}} = 0$. We use a percentage ($x\%$) notation for evaluation purposes. It indicates that the annealing process results in $x\%$ closer to the optimal solution from the baseline. We put 0% at the expected Ising energy of random spin configuration. These figures show that APC-SCA can obtain a better solution than the fine-tuned SCA. Although the trend varies from problem to problem, the range from 0.4 to 0.9 looks suitable for obtaining a good solution.

To measure the extent of implementation overhead, we solved G22 by APC-SCA with $r_q = 0.45$ and $q_{\text{limit}} = 0$ while

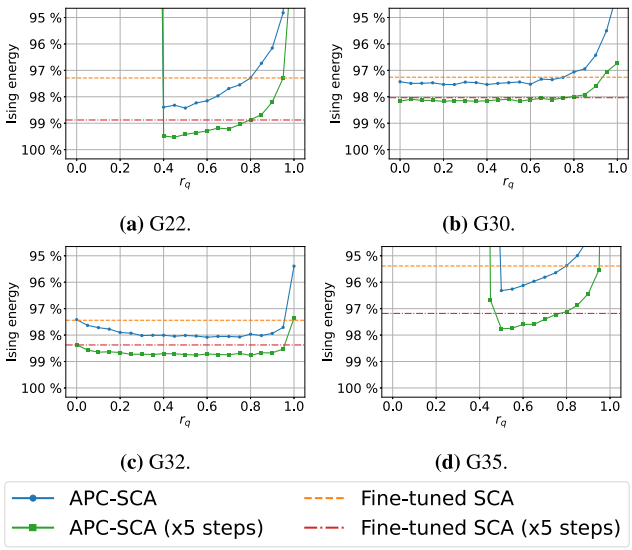


Fig. 3 The average Ising energy when solving max-cut problems with APC-SCA while varying r_q from 0.0 to 1.0. The blue and green lines show the APC-SCA with 1,000 and 5,000 MC steps, respectively. The orange and red horizontal lines show the fine-tuned SCA with 1,000 and 5,000 MC steps, respectively. These results indicate that APC-SCA can obtain a better solution than the fine-tuned SCA.

Table 4 Parameter settings for G22 to evaluate the relationship between execution time and Ising energy.

	SA	Fine-tuned SCA	APC-SCA
# of MC steps (S)	10,240	400	400
	20,480	800	800
	40,960	1,600	1,600
	81,920	3,200	3,200
	163,840	6,400	6,400
	327,680	12,800	12,800
	655,360	25,600	25,600
	1,310,720	51,200	51,200
	2,621,440	102,400	102,400
	5,242,880	204,800	204,800
	10,485,760	409,600	409,600

varying the MC steps listed in Table 4. Figure 4 shows the result of the relationship between execution time and solution quality. The execution time in this figure is calculated by dividing the overall annealing time by the number of trials $M = 128$. For comparison, we solved the same problem by SA and fine-tuned SCA assuming the settings in Table 4. Note that we implemented the algorithm of SA using the work [21] as a reference and ran it on the same GPU as APC-SCA. Figure 4 demonstrates that APC-SCA is superior to SA and fine-tuned SCA. By comparing the execution time between fine-tuned SCA and APC-SCA, we find that GPU can realize our autonomous pinning parameter control with little overhead. That is because each q_i is updated independently, and hence the update procedure (i.e., Lines 10 and 13 in **Algorithm 2**) is easily parallelized.

4.3 32-Queen Problem

The 32-queen problem is used in the experiments as a typical constraint satisfaction problem. Figure 5 shows the result when solving 32-queen by APC-SCA with $r_q = 0.9$ while varing q_{limit} from 0 to $\lambda/128$. The upper and lower figures show the average Ising energies and the constraint satisfaction (CS) rates, respectively. As metrics of the solution quality, the CS rate indicates the number of optimal (i.e., $H(\sigma) = 0$) solutions obtained by solving the problem $M = 128$ times. Figure 5 indicates that the solution quality is improved by using small $q_{\text{limit}} > 0$. Practically, $q_{\text{limit}} = \lambda/512$ achieves a high CS rate.

Figure 6 shows the relationship between the execution time and the CS rate when solving 32-queen by APC-SCA with $r_q = 0.9$ and $q_{\text{limit}} = 0$ or $\lambda/512$. For comparison, we also solved the same problem by SA and fine-tuned SCA. Here we varied the number of MC steps as listed in Table 4. Comparing the two results by APC-SCA, $q_{\text{limit}} = \lambda/512$ can improve the solution quality.

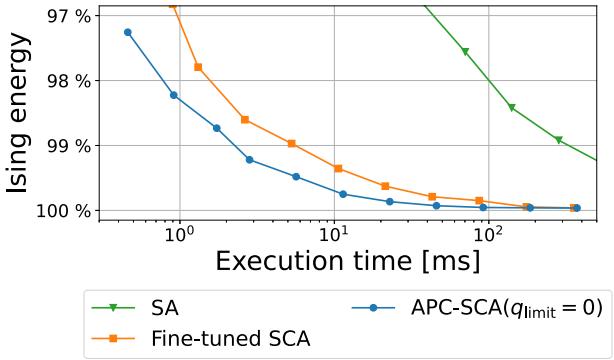


Fig. 4 Comparison of average Ising energies when solving G22 while varying the number of MC steps. APC-SCA can obtain better solutions in a short time compared to SA and fine-tuned SCA.

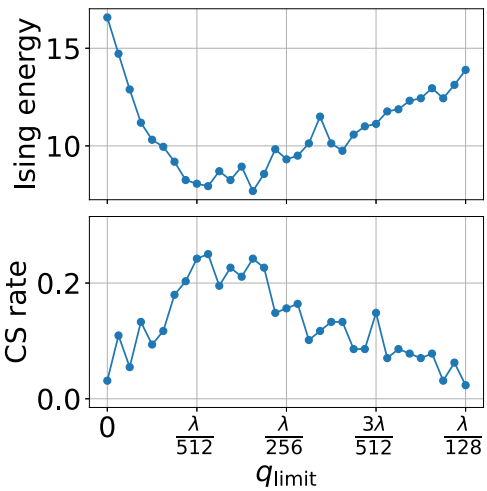


Fig. 5 Average Ising energies (upper) and CS rates (lower) when solving 32-queen by APC-SCA while varying q_{limit} from 0 to $\lambda/128$. By using a small positive value as q_{limit}, the solution quality can be improved significantly.

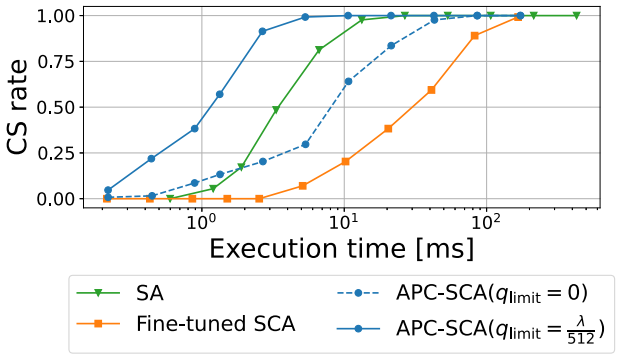
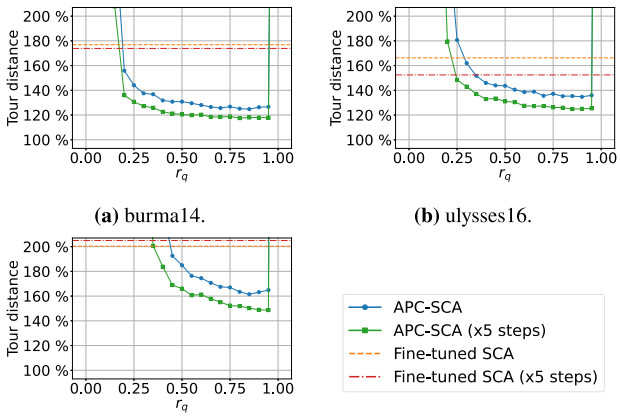


Fig. 6 Comparison of CS rates when solving 32-queen while varying the number of MC steps. APC-SCA with $q_{\text{limit}} = \lambda/512$ can obtain high-quality solutions in a shorter time than SA and fine-tuned SCA.



(a) burma14. **(b)** ulysses16.

(c) ulysses22.

Fig. 7 The average tour distance when solving three TSPs with APC-SCA while varying r_q from 0.0 to 1.0. The blue and green lines show the APC-SCA with 10,000 and 50,000 MC steps, respectively. The orange and red lines show the fine-tuned SCA with 10,000 and 50,000 MC steps, respectively.

4.4 Traveling Salesman Problems (TSPs)

TSP is the most difficult problem in the three problem classes used in this study because large energy walls are generated between two local minima by the constraints of this problem. Although there is a viable approach that decreases the importance of the constraints when converting the problem into an Ising model, we use the same Ising models as IPA [14] for comparison. As listed in Table 1, we assumed three TSPs from TSPLIB [22]. Figure 7 shows the average tour distance when solving them by APC-SCA and fine-tuned SCA with 10,000 and 50,000 MC steps. In APC-SCA, we varied r_q from 0.0 to 1.0. We use a percentage (x%) notation for evaluation purposes. The 100% and 200% grid lines show the minimum and double minimum tour distances, respectively. Figure 7 indicates that $r_q \approx 0.8$ is appropriate for TSPs. We confirmed that tour distances can be 31.5% to 52.1% closer at 10,000 MC steps and 27.6% to 56.3% closer at 50,000 MC steps to the optimal distance compared with fine-tuned SCA.

Table 5 compares APC-SCA, APC-MA, Fine-tuned SCA, IPA [14], and MA. We cited the IPA's result from [14] and tried to solve 100 times for each problem with these algorithms. As for APC-SCA and APC-MA, we show the results of $r_q = 0.8$. We also use Eq. (10) as the temperature scheduling in MA and APC-MA, where $\beta_0 = 9.0 \times 10^{-4}$ for burma14, $\beta_0 = 8.0 \times 10^{-4}$ for ulysses16, and $\beta_0 = 5.0 \times 10^{-4}$ for ulysses22. Comparing APC-SCA with fine-tuned SCA, APC-SCA can obtain superior solutions to fine-tuned SCA in all problems in terms of average, minimum, and maximum metrics. APC-SCA is superior to IPA in terms of minimum metrics, and is not significantly inferior to IPA in terms of average and maximum metrics. Since annealing computation is generally performed in multiple trials, obtaining minimum energy solutions can be an advantage.

We also compare APC-MA to that without the dropout. In APC-MA, after the dropout occurs, the spin will flip with a high probability, and the pinning effect will be reset to the initial value. This means that the dropout disturbs the effect of autonomous control. For this reason, APC-MA without the dropout outperforms APC-MA in all metrics. As a result, APC-MA (without the dropout) can be up to 26.9% closer at 10,000 MC steps and up to 30.1% closer at 50,000 MC steps to the optimal tour distance than MA.

Comparing APC-SCA with APC-MA, the latter obtained better solutions on average. This may be attributed to temperature scheduling, the main difference between APC-SCA and APC-MA. Unlike exponential scheduling, logarithmic scheduling maintains a constant temperature for a long time. Therefore, logarithmic scheduling achieves more efficient searches when an appropriate temperature is set than exponential annealing scheduling, which rapidly lowers the temperature. However, in ulysses22 at 10,000 MC steps, APC-MA decreased by 18.9% in accuracy compared to MA. This may be attributed to incomplete temperature cooling, meaning that more MC steps (e.g., 50,000 MC steps) are required for the convergence. On the other hand, incorporating IPA-like temperature control into APC-SCA or APC-MA is also an interesting direction in the future. However, it is not easy to realize because both parameter control methods depend on the same branch condition, i.e., whether spin-flips occur.

Finally, we observe the escape from local minima in ulysses16 and illustrate the spin transition during the annealing process. Figures 8 (a) and 8 (b) illustrate the spin transition in the final stage of annealing with fine-tuned SCA and APC-SCA, respectively. While the fine-tuned SCA is stuck in a local minimum without being able to get over the energy walls, APC-SCA can do and transition to a lower energy state. From these results, we confirm that APC-SCA achieves escape from local minima.

5. Conclusion

We have proposed APC-SCA, an annealing algorithm that enables escape from local minima by utilizing autonomous pinning parameter control in SCA. APC-SCA realizes the

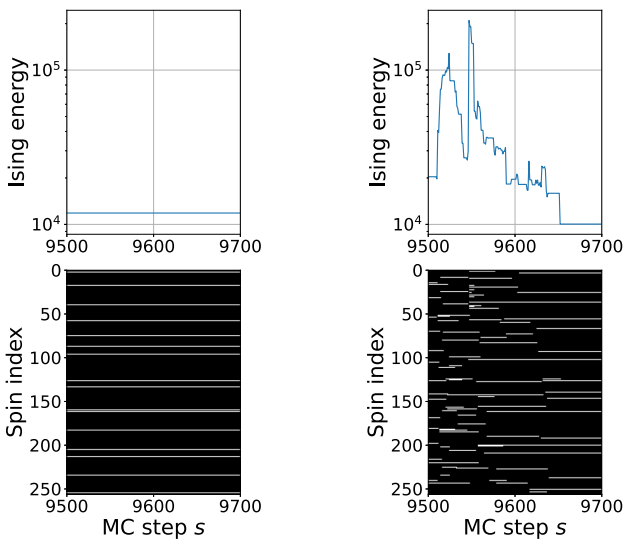
Table 5 The performance of APC-SCA, APC-MA, fine-tuned SCA, IPA, and MA for solving burma14, ulysses16, and ulysses22. We cite the IPA's result from [14] and summarize the average, maximum, and minimum tour distance. All methods tried to solve 100 times for each problem. Tables 5 (a) and 5 (b) show the results of 10,000 and 50,000 MC steps, respectively.

(a) 10,000 MC steps

Benchmark	Metrics	APC-SCA	APC-MA	APC-MA w/o dropout	Fine-tuned SCA	IPA [14]	MA
burma14	Ave.	4138.3	4407.1	**4075.4**	5878.1	4241.6	4422.0
	Max.	4702.0	4943.0	**4409.0**	8091.0	4703.0	4916.0
	Min.	3514.0	3673.0	3516.0	4331.0	3839.0	**3444.0**
	Std.	205.6	251.0	182.7	715.5	185.1	263.1
ulysses16	Ave.	9419.5	10294.4	9209.0	11403.9	**8804.2**	11055.7
	Max.	10988.0	12474.0	10922.0	13611.0	**9869.0**	13554.0
	Min.	7768.0	8024.0	**7659.0**	8676.0	7816.0	8485.0
	Std.	527.7	899.0	676.3	1009.2	407.9	1126.8
ulysses22	Ave.	11488.6	17337.4	13626.0	14053.1	**11170.0**	12297.2
	Max.	14309.0	23035.0	17807.0	17415.0	**12301.0**	14894.0
	Min.	**9144.0**	12055.0	10127.0	10061.0	9527.0	10545.0
	Std.	942.2	2580.9	1453.9	1466.0	527.3	861.0

(b) 50,000 MC steps

Benchmark	Metrics	APC-SCA	APC-MA	APC-MA w/o dropout	Fine-tuned SCA	IPA [14]	MA
burma14	Ave.	3902.4	4042.9	**3821.9**	5777.6	4018.5	4235.7
	Max.	4271.0	4462.0	**4133.0**	7613.0	4423.0	4846.0
	Min.	3459.0	3667.0	**3439.0**	3897.0	3580.0	3745.0
	Std.	155.5	183.6	138.6	667.7	159.9	237.1
ulysses16	Ave.	8589.0	10197.5	9086.4	10456.7	**8387.6**	11148.0
	Max.	9529.0	12668.0	11806.0	12528.0	**9218.0**	13457.0
	Min.	7356.0	8359.0	**7068.0**	8779.0	7554.0	8810.0
	Std.	424.4	743.8	667.4	786.7	303.3	1045.1
ulysses22	Ave.	10630.8	12296.9	10756.0	14374.9	**10389.0**	11664.6
	Max.	11808.0	14645.0	12465.0	19199.0	**11167.0**	13672.0
	Min.	8817.0	9405.0	**8557.0**	10368.0	9163.0	9729.0
	Std.	636.3	978.8	668.1	1421.0	433.7	806.6



(a) Fine-tuned SCA **(b)** APC-SCA.

Fig. 8 Illustrations of the Ising energy and spin state transitions during the annealing process with ulysses16. The spin state ('+1' in white, '−1' in black) at one step is arranged vertically, and the horizontal axis represents the number of MC steps.

autonomous control by updating an individual pinning parameter q_i based only on the flip information of the i-th spin. Thanks to this independent q_i update, we can implement APC-SCA on a GPU in almost the same manner as SCA did. Through the experiments on max-cut, 32-queen, and TSP, we confirmed that APC-SCA could escape from local minima. For a max-cut problem and a 32-queen problem, APC-SCA could obtain equivalent quality solutions in a shorter time than SA and fine-tuned SCA. For TSP, we confirm that tour distances are 31.5% to 52.1% closer at 10,000 MC steps and 27.6% to 56.3% closer at 50,000 MC steps to the optimal distance than fine-tuned SCA.

As a future direction, we envision improved APC-SCA in combination with dynamic temperature control like IPA. Moreover, it may also be an exciting research direction to design a parallel annealing hardware architecture that adopts APC-SCA as the operating principle.

Parallel spin-update algorithms like SCA are powerful annealing methods for solving COPs efficiently. However, some problem classes make the annealing machine stuck on local minima. Although we believe that the advantage of annealing computation is that it can solve COPs on a single platform, it is not good at solving problems, like TSP, that are stuck in local minima. The experimental results in this paper indicate that parallel annealing can escape from local minima by controlling the parameters, and uniformly solving various COPs becomes realistic. By expanding further our knowledge of the graph topology and the problem scale, we hope to establish the annealing machine as a unified solver for COPs.

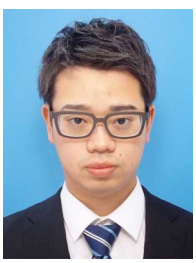
Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR18K3, Japan.

References

[1] L. Wei, Z. Zhang, D. Zhang, and S.C.H. Leung, "A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints," European Journal of Operational Research, vol.265, no.3, pp.843–859, 2018.
[2] A.M. Fathollahi-Fard, K. Govindan, M. Hajiaghaei-Keshteli, and A. Ahmadi, "A green home health care supply chain: New modified simulated annealing algorithms," Journal of Cleaner Production, vol.240, 118200, 2019.
[3] W. Zhang, A. Maleki, M.A. Rosen, and J. Liu, "Optimization with a simulated annealing algorithm of a hybrid system for renewable energy including battery and hydrogen storage," Energy, vol.163, pp.191–207, 2018.
[4] M.C. Aguitoni, L.V. Pavão, and M. Antonio da Silva Sá Ravagnani, "Heat exchanger network synthesis combining simulated annealing and differential evolution," Energy, vol.181, pp.654–664, 2019.
[5] A. Lucas, "Ising formulations of many NP problems," Frontiers in Physics, vol.2, 5, 2014.
[6] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," Phys. Rev. E, vol.58, no.5, pp.5355–5363, Nov. 1998.
[7] M.W. Johnson, M.H.S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A.J. Berkley, J. Johansson, P. Bunyk, E.M. Chapple, C. Enderud, J.P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M.C. Thom, E. Tolkacheva, C.J.S. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose, "Quantum annealing with manufactured spins," Nature, vol.473, no.7346, pp.194–198, 2011.
[8] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, "Optimization by simulated annealing," Science, vol.220, no.4598, pp.671–680, 1983.
[9] 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, 48, 2019.
[10] T. Okuyama, T. Sonobe, K. Kawarabayashi, and M. Yamaoka, "Binary optimization by momentum annealing," Phys. Rev. E, vol.100, no.1, 012111, July 2019.
[11] B.H. Fukushima-Kimura, S. Handa, K. Kamakura, Y. Kamijima, and A. Sakai, "Mixing time and simulated annealing for the stochastic cellular automata," arXiv preprint arXiv:2007.11287, 2021.
[12] K. Yamamoto, K. Kawamura, K. Ando, N. Mertig, T. Takemoto, M. Yamaoka, H. Teramoto, A. Sakai, S. Takamaeda-Yamazaki, and M. Motomura, "STATICA: A 512-spin 0.25M-weight annealing processor with an all-spin-updates-at-once architecture for combinatorial optimization with complete spin-spin interactions," IEEE J. Solid-State Circuits, vol.56, no.1, pp.165–178, 2021.
[13] S. Matsubara, M. Takatsu, T. Miyazawa, T. Shibasaki, Y. Watanabe, K. Takemoto, and H. Tamura, "Digital annealer for high-speed solving of combinatorial optimization problems and its applications," 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), pp.667–672, 2020.
[14] Q. Tao and J. Han, "Solving traveling salesman problems via a parallel fully connected ising machine," Proc. 59th ACM/IEEE Design Automation Conference, DAC '22, New York, NY, USA, pp.1123–1128, Association for Computing Machinery, 2022.
[15] D. Okonogi, S. Jimbo, K. Ando, T. Van Chu, J. Yu, M. Motomura, and K. Kawamura, "APC-SCA: A fully-parallel annealing algorithm with autonomous pinning effect control," 2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp.414–420, 2022.
[16] D. Venturelli, D. Marchand, and G. Rojo, "Job shop scheduling solver based on quantum annealing," Proc. ICAPS-16 Workshop on Constraint Satisfaction Techniques for Planning and Scheduling (COPLAS), pp.25–34, 2016.
[17] K. Tamura, T. Shirai, H. Katsura, S. Tanaka, and N. Togawa, "Performance comparison of typical binary-integer encodings in an Ising machine," IEEE Access, vol.9, pp.81032–81039, 2021.
[18] K. Ikeda, Y. Nakamura, and T.S. Humble, "Application of quantum annealing to nurse scheduling problem," Scientific Reports, vol.9, no.1, 12837, 2019.
[19] Y. Ye, "Index of /~yyye/yyye/Gset," https://web.stanford.edu/~yyye/yyye/Gset/, accessed Dec. 16. 2022.
[20] S. Patel, L. Chen, P. Canoza, and S. Salahuddin, "Ising model optimization problems on a FPGA accelerated restricted Boltzmann machine," 2020.
[21] K. Kawamura, K. Okawa, G. Gutmann, T.V. Chu, J. Yu, and M. Motomura, "GPU-based acceleration of fully parallel annealing algorithm for combinatorial optimization," Proc. 34th IEEE International System-on-Chip Conference (SOCC), pp.1–6, 2022.
[22] G. Reinelt, "TSPLIB - Discrete and combinatorial optimization," http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95, accessed Dec. 16. 2022.

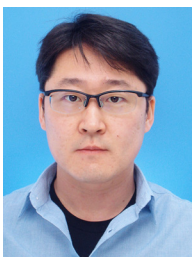
Daiki Okonogi received his B.S. degree in engineering from NIAD-QE with the National Institute of Technology (KOSEN), Gunma College, in 2021 and received his M.E. degree in information and communication engineering from Tokyo Institute of Technology in 2023. He is currently pursuing the Ph.D. degree with the Tokyo Institute of Technology. His current research interests include Ising computing and computer architecture.



Satoru Jimbo received his B.S. degree in mechanics from Tohoku University in 2021 and received his M.E. degree in information and communication engineering from the Tokyo Institute of Technology in 2023. He is currently pursuing the Ph.D. degree with the Tokyo Institute of Technology. His research interests include Ising computing.

Kota Ando received his B.E. degree in electronics and M.S. degree in information technology from Hokkaido University, Sapporo, Japan, in 2016 and 2018, respectively. He received his Ph.D. degree in engineering from Tokyo Institute of Technology, Yokohama, Japan, in 2021. He is currently an assistant professor at Hokkaido University, Sapporo, Japan. He worked as an assistant professor at Tokyo Institute of Technology, Yokohama, Japan, from 2021 to 2022. He was a JSPS Research Fellow from 2018 to 2021 during his Ph.D. study. His research interests include reconfigurable architectures, memory-centric processing, and hardware-aware algorithms for efficient deep learning processing. He has been a member of IEICE since 2016. He received the Best Student Presentation Award from the Technical Committee on Reconfigurable Systems of IEICE, Japan, in 2016 and 2017, respectively, the Best Student Poster Award from the Technical Committee on Integrated Circuits and Devices of IEICE in 2018, and the Best Paper Award at the 2018 International Conference on Field-Programmable Technology.

Thiem Van Chu completed his Ph.D. at Tokyo Institute of Technology in 2018. Upon graduation, he joined the School of Information Science, Japan Advanced Institute of Science and Technology as an Assistant Professor. In 2020, he moved to Tokyo Institute of Technology. His research interests lie at the intersection of computer architecture, reconfigurable computing, and machine learning.



Jaehoon Yu received his B.E. degree in Electrical and Electronic Engineering and his M.S. degree in Communications and Computer Engineering from Kyoto University, Kyoto, Japan, in 2005 and 2007, respectively, and received his Ph.D. degree in Information Systems Engineering from Osaka University, Osaka, Japan, in 2013. From 2013 to 2019, he was an assistant professor at Osaka University. He is currently an associate professor at Tokyo Institute of Technology, Japan. His research interests include computer vision, machine learning, and system-level design. He is a member of IEEE, IEICE, and IPSJ.

Masato Motomura received B.S., M.S., and Ph.D. (electrical engineering) from Kyoto University, Kyoto, Japan, in 1985, 1987, and 1996, respectively. In 1987, he joined NEC Central Research Laboratories, Kawasaki, Japan, where he worked on various hardware architectures including string search engines, multi-threaded on-chip parallel processors, embedded DRAM-FPGA hybrid systems, memory-based processors, and reconfigurable systems. From 2001 to 2008, he was with NEC Electronics, Kawasaki, Japan, where he led research and business development of dynamically reconfigurable processor (DRP) that he invented. He was also a visiting researcher at the MIT Laboratory for Computer Science, Cambridge, MA, USA, from 1991 to 1992. He had been a professor at Hokkaido University, Sapporo, Japan, from 2011 to 2019. After that, he has been a professor at Tokyo Institute of Technology, Yokohama, Japan, where he is leading artificially intelligent computing (ArtIC) research unit. He is actively working on reconfigurable and parallel architectures for deep neural networks, machine learning, annealing machines, and intelligent computing in general. He is an IEEE Fellow, and a member of IEICE, IPSJ, JSAI, and EAJ. He was a recipient of the IEEE JSSC Annual Best Paper Award in 1992, the IPSJ Annual Best Paper Award in 1999, the IEICE Achievement Award in 2011, and the ISSCC Silkroad Award as the corresponding author in 2018, respectively.

Kazushi Kawamura received the B. Eng., M. Eng. and Dr. Eng. degrees from Waseda University in 2012, 2013 and 2016, respectively, all in computer science. From 2018 to 2019, he was an Assistant Professor in the Department of Communications and Computer Engineering, Waseda University. He is presently a Specially Appointed Assistant Professor in the Institute of Innovative Research, Tokyo Institute of Technology. His research interests include Ising computing and LSI design methodologies. He is a member of IEEE, IEICE, and IPSJ.