Received 26 July 2023; revised 24 October 2023; accepted 25 October 2023. Date of publication 2 November 2023; date of current version 15 November 2023. The review of this article was arranged by Associate Editor U. A Khan.

Digital Object Identifier 10.1109/OJSP.2023.3329756

Local Energy Distribution Based Hyperparameter Determination for Stochastic Simulated Annealing

NAOYA ONIZAWA [1] (Member, IEEE), KYO KUROKI[1], DUCKGYU SHIN [1,2], AND TAKAHIRO HANYU [1] (Senior Member, IEEE)

[1]Research Institute of Electrical Commuinaction, Tohoku University, Sendai 980-8577, Japan
[2]Graduate School of Engineering, Tohoku University, Sendai 980-8579, Japan

CORRESPONDING AUTHOR: NAOYA ONIZAWA (email: naoya.onizawa.a7@tohoku.ac.jp).

This work was supported by JST CREST under Grants JPMJCR19K3, JSPS KAKENHI and JP21H03404.

ABSTRACT This paper presents a local energy distribution based hyperparameter determination for stochastic simulated annealing (SSA). SSA is capable of solving combinatorial optimization problems faster than typical simulated annealing (SA), but requires a time-consuming hyperparameter search. The proposed method determines hyperparameters based on the local energy distributions of spins (probabilistic bits). The spin is a basic computing element of SSA and is graphically connected to other spins with its weights. The distribution of the local energy can be estimated based on the central limit theorem (CLT). The CLT-based normal distribution is used to determine the hyperparameters, which reduces the time complexity for hyperparameter search from $\mathcal{O}(n^3)$ of the conventional method to $\mathcal{O}(1)$. The performance of SSA with the determined hyperparameters is evaluated on the Gset and K2000 benchmarks for maximum-cut problems. The results show that the proposed method achieves mean cut values of approximately 98% of the best-known cut values.

INDEX TERMS Combinatorial optimization, Hamiltonian, Ising model, maximum-cut problem, simulated annealing, stochastic computing.

I. INTRODUCTION

Combinatorial optimization is used to solve many practical problems in various fields and involves finding the optimal solution for a given objective function subject to a set of constraints [1]. Combinatorial optimization problems are often NP-hard, meaning that finding the optimal solution requires an exponentially large amount of time with respect to the problem size [2]. One such approach is simulated annealing, which is a stochastic optimization method inspired by the physical annealing process in materials science [3], [4]. SA has been successfully applied to various combinatorial optimization problems, such as the traveling salesman problem, the graph coloring problem, and the maximum cut problem [5]. Quantum annealing (QA) is another optimization method that uses quantum mechanics to solve combinatorial optimization problems [6], [7]. It employs quantum devices that exploit quantum superposition and entanglement to explore the search space more efficiently than classical algorithms [8]; however, QA devices are still in the early stage of development and have limited qubit coherence times and connectivity, which restricts their ability to solve large-scale problems [9].

Recently, a novel method known as stochastic-computing-based simulated annealing (SSA) has been introduced, which significantly expedites the annealing process compared to traditional SA and QA methods [10]. SSA is a parallel variant of simulated annealing that utilizes an approximation of probabilistic bits (p-bits) [11], implemented through stochastic computing [12], [13]. Its flexibility allows for implementation in both software and hardware [14], potentially scaling up to handle large-scale problems. Combinatorial optimization problems are expressed using an Ising model [15], which comprises spin states denoted as σ, spin biases symbolized by h, and weights between spins represented by J. The Ising model embodies an energy. In SSA, a pseudo inverse temperature is incrementally raised to determine a solution that corresponds to the global minimum energy, using random signals. To enhance performance (i.e., to increase the likelihood of reaching the global minimum energy), hyperparameters related to the pseudo inverse temperature and the random signals must be fine-tuned. This process, however, can be time-consuming due to the significant computational cost involved.

In this paper, we introduce a statistical method for determining hyperparameters for SSA that can eliminate the time-consuming hyperparameter search. In the proposed method, the hyperparameters are calculated based on the local energy distributions of spins. The local energy at each spin is calculated by the spin bias and the summation of the multiplication of the spin weights and other spin states. As the number of connections from the other spins is sufficiently large, the distribution of the local energy can be estimated based on the central limit theorem (CLT). Therefore, the local-energy distribution can be approximated by the normal distribution, which determines the hyperparameters, such as the pseudo inverse temperature and the noise signals. Compared with a conventional method that uses a grid search (i.e. typical hyperparameter search) in [10], the time complexity for the hyperparameter search is reduced to $\mathcal{O}(1)$ from $\mathcal{O}(n^3)$. Using the determined hyperparameters, a unique noise control at each spin is also introduced for SSA to enhance the performance. The determined hyperparameters are evaluated in maximum-cut (MAX-CUT) problems [16] that are a typical combinatorial optimization problem. On the MAX-CUT problem benchmarks, such as Gset [17] and K2000 [18], the mean cut values using the proposed method reach approximately 98% of the best-known cut values for 1,000 cycles.

The contributions of the paper are:

1) Introducing a new statistical method for determining hyperparameters in SSA that can eliminate the time-consuming hyperparameter search required by the conventional method.
2) Demonstrating the effectiveness of the proposed method in solving the MAX-CUT problems and achieving mean cut values that are close to the best-known cut values.
3) Reducing the time complexity for hyperparameter search from $\mathcal{O}(n^3)$ to $\mathcal{O}(1)$ compared to the conventional method.

The rest of the paper is structured as follows. Section II reviews SSA and its hyperparameters. Section III presents the proposed statistical method for determining the hyperparameters in SSA. Section IV introduces the unique noise magnitude for each spin. Section V evaluates the determined hyperparameters and compares SSA with conventional SA. Section VI discusses the hyperparameter determination with the conventional hyperparameter search. Finally, Section VII concludes the paper.

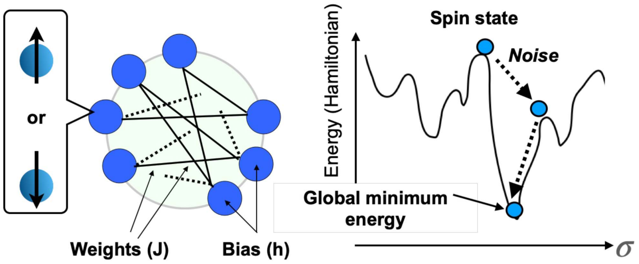


FIGURE 1. Stochastic simulated annealing (SSA) [10] based on a spin network that consists of spins, spin biases h, and spin weights J. Spin states are flipped between "+1" and "−1" by noise signals to reach the global minimum energy of the Hamiltonian.

II. PRELIMINARY

A. STOCHASTIC SIMULATED ANNEALING (SSA)

There are several SA methods, such as serial updating [5] and parallel tempering [19]. Recently, SSA has been presented as one of the SA methods [10]. SSA is designed based on p-bit-based SA (pSA), which was introduced in [20]. A p-bit is a probabilistic bit that can be one of two spin states, '+1' and '−1'. It has been proposed for use in invertible logic, an unconventional computing technique [11], [21], [22]. pSA, implemented on an underlying Boltzmann machine [23], realizes parallel updating of the spins for fast simulated annealing. SSA approximates the behavior of p-bit using stochastic computing, which overcomes the slow convergence to the global minimum energy of pSA. Compared to other SA methods, SSA achieved the highest maximum-cut value in K2000 [24].

Fig. 1 illustrates SSA, which is on a spin network designed using spins, spin biases h, and spin weights J between spins. Each spin state σ can be either '−1' and '+1'. The spin network is an Ising model [15] that represents a Hamiltonian (energy function) as follows:

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

To solve a combinatorial optimization problem, the problem is mapped to the Hamiltonian coefficients of h and J. The coefficients differ depending on the problem, such as graph isomorphism and MAX-CUT problems [16], [25]. During the annealing process, spin states are flipped between '+1' and '−1' in an attempt to reach the global minimum of the Hamiltonian.

In SSA, the spin behavior is modeled by approximating the p-bit using integral stochastic computing (ISC) [26]. Note that ISC is an extended version of typical stochastic computing [12], [13], which can be used for area-efficient hardware implementation [27], [28], [29]. An i-th spin has a bias h_i and edge weights J_{ij} from/to other spins. At each cycle, the spin state is updated as follows:

$$I_i(t+1) = h_i + \sum_j J_{ij}\cdot\sigma_j(t) + n_{\text{rnd}}\cdot r_i(t), \tag{2a}$$

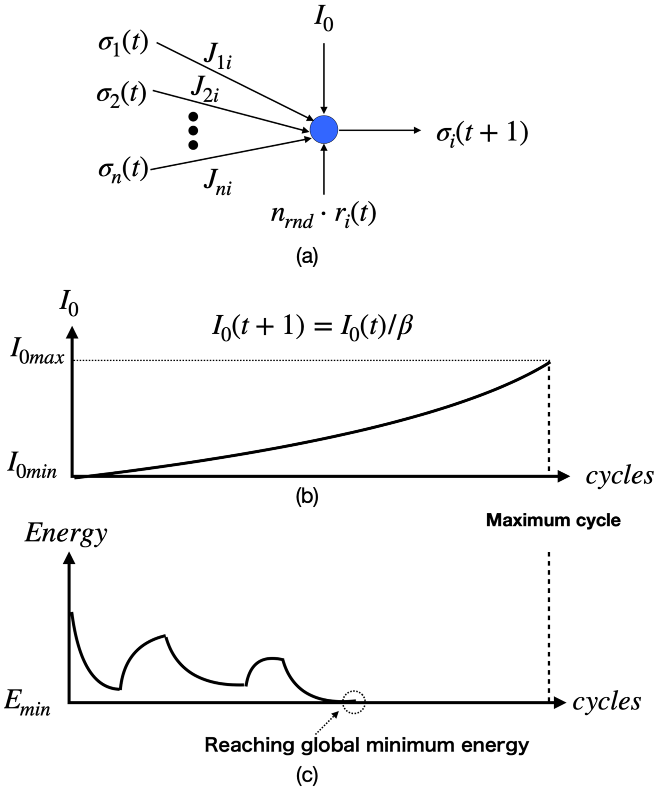


FIGURE 2. Annealing process of SSA: (a) spin, (b) I_0 control, and (c) energy transition. In SSA, I_0 is gradually increased from $I_{0\min}$ to $I_{0\max}$ with weighted noise signals $n_{\text{rnd}} \cdot r_i(t)$. These hyperparameters need to be selected for the best performance of SSA.

$$Is_i(t+1) = \begin{cases} I_0(t) - \alpha, \text{ if } Is_i(t) + I_i(t+1) \geq I_0(t) \\ -I_0(t), \text{ else if } Is_i(t) + I_i(t+1) < -I_0(t) \\ Is_i(t) + I_i(t+1), \text{ otherwise} \end{cases} \tag{2b}$$

$$\sigma_i(t+1) = \begin{cases} 1, & \text{if } Is_i(t+1) \geq 0 \\ -1, & \text{otherwise,} \end{cases} \tag{2c}$$

where $\sigma_i(t) \in \{-1, 1\}$ and $\sigma_i(t+1) \in \{-1, 1\}$ is binary input and output spin states, respectively. $I_i(t+1)$ and $Is_i(t+1)$ are real-valued internal signals and n_{rnd} is the magnitude of a random signal, $r_i(t) \in \{-1, 1\}$. I_0 is the pseudo inverse temperature and α is the minimum resolution of data representation. If only integer values are used in (2), α is 1 [10]. If floating-point values are used, α can be approximated by 0.

B. HYPERPARAMETER SEARCH FOR SSA

Let us explain the annealing process of SSA using Fig. 2. At each cycle, I_0 is gradually increased from $I_{0\min}$ to $I_{0\max}$ as $I_0(t+1) = I_0(t)/\beta$. The parameter I_0 controls the strength of the external field that acts on each spin. When I_0 is small, the spin states can be easily flipped to search for many spin states. This helps the spin states escape from local energy minima and explore a wider region of the solution space. When I_0 is large, the spin states can be stabilized in an attempt to reach the global minimum energy. Additionally, n_{rnd} is an important parameter used to control the stability of the spin states.

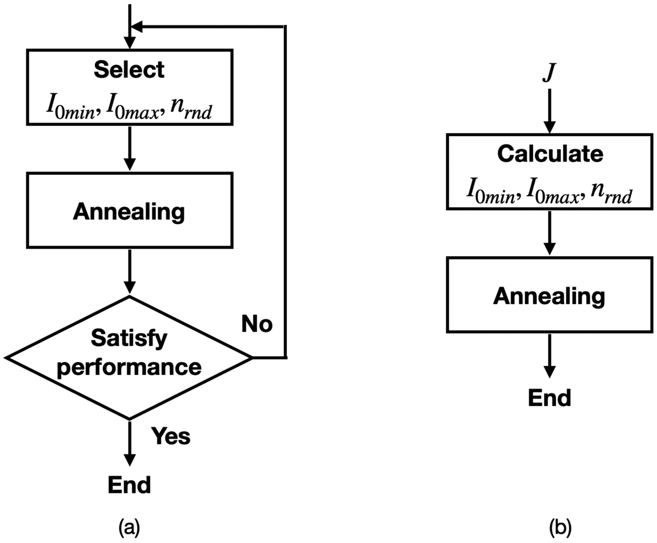


FIGURE 3. Selecting hyperparameters in SSA: (a) conventional method [10], [24] and (b) proposed method. The proposed method statistically determines hyperparameters without searching while the conventional method uses the grid/random search to find good hyperparameters.

To achieve a high probability of reaching the global minimum energy, the three hyperparameters of $I_{0\min}$, $I_{0\max}$ and n_{rnd} need to be carefully selected. In previous studies [10], [24], a grid search or a random search has been used to select these hyperparameters, as shown in Fig. 3(a). In a grid search, a search space is defined as a grid of hyperparameter values, and every position in the grid is evaluated. In a random search, a search space is defined as a bounded domain of hyperparameter values, and points are randomly sampled within that domain. If good hyperparameters are not found, the search process is iteratively carried out until the required performance (e.g., probability of reaching the global minimum energy) is satisfied. While these methods are simple, they often require a long time to find good hyperparameters. In particular, when solving large combinatorial optimization problems, the time-consuming hyperparameter search could be a critical issue. In this paper, we determine the hyperparameters statistically by calculation, without performing a search, while achieving high performance of SSA, as shown in Fig. 3(b).

III. HYPERPARAMETER DETERMINATION BASED ON LOCAL ENERGY DISTRIBUTION

A. LOCAL ENERGY DISTRIBUTION

In the spin network, the spins interact with each other via pairwise interactions, which are often represented by the edge weights J_{ij} in a graph representation of the Ising model. The total energy of the spin network is defined as per (1), whereas

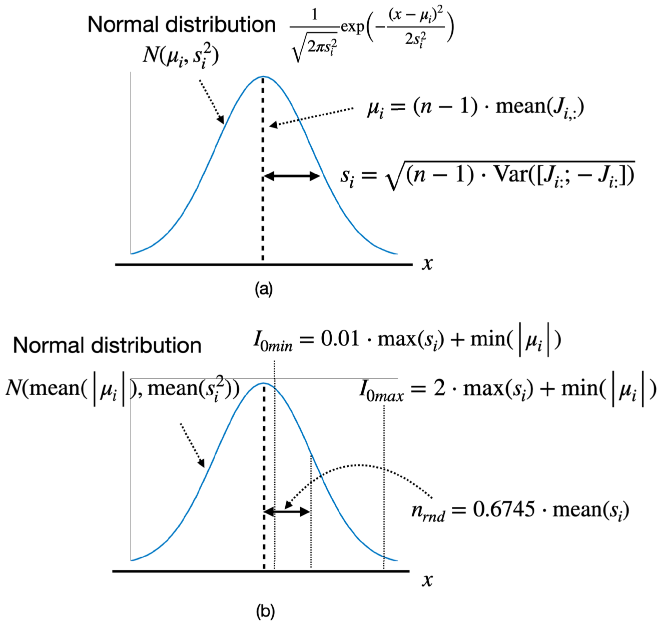

Normal distribution
$N(\mu_i, s_i^2)$
$\frac{1}{\sqrt{2\pi s_i^2}}\exp\left(-\frac{(x-\mu_i)^2}{2s_i^2}\right)$
$\mu_i = (n-1)\cdot \mathrm{mean}(J_{i:})$
$s_i = \sqrt{(n-1)\cdot \mathrm{Var}([J_{i:}; -J_{i:}])}$
x
(a)
Normal distribution
$N(\mathrm{mean}(|\mu_i|), \mathrm{mean}(s_i^2))$
$I_{0min} = 0.01\cdot \max(s_i) + \min(|\mu_i|)$
$I_{0max} = 2\cdot \max(s_i) + \min(|\mu_i|)$
$n_{rnd} = 0.6745\cdot \mathrm{mean}(s_i)$
x
(b)

FIGURE 4. Hyperparameter determination in SSA: (a) local energy distribution at *i*-th spin approximated by the normal distribution and (b) hyperparameter determination using the mean of all the local energy distributions. The three hyperparameters of $I_{0\min}$, $I_{0\max}$, and n_{rnd} are determined using the normal distribution $N(\mathrm{mean}(|\mu_i|), \mathrm{mean}(s_i^2))$.

the local energy at the i-th spin is defined as follows:

$$H_L(\sigma_l) = -h_i\sigma_i - \sum_{i\neq j} J_{ij}\sigma_j\sigma_i. \tag{3}$$

The local energy varies depending on the states of other spins σ_j that are connected to it with the edge weights J_{ij}. As the number of edges connected to a spin becomes larger, the distribution of the local energy becomes more and more Gaussian, which can be explained by the central limit theorem (CLT). In this case, the random variables are the edge weights J_{ij}, and the sum is over all the edges connected to a spin.

In Fig. 4(a), the local energy distribution at the spin is illustrated, where n is the total number of spins representing a combinatorial optimization problem. The i-th spin connects $(n-1)$ edges to other spins with the edge weights $J_{i:}$. Note that $J_{i:}$ is a vector containing all the edge weights connected to the i-th spin. Using CLT, the local energy distribution can be approximated by a normal distribution $N(\mu_i, s_i^2)$ with mean μ_i and standard deviation s_i calculated as follows:

$$\mu_i = (n-1)\cdot \mathrm{mean}(J_{i:}), \tag{4a}$$

$$s_i = \sqrt{(n-1)\cdot \mathrm{Var}([J_{i:}; -J_{i:}])}. \tag{4b}$$

where the semicolon (;) is used to indicte concatenation of the two vectors.

B. HYPERPARAMETER DETERMINATION

Fig. 4(b) illustrates how to determine the three hyperparameters of $I_{0\min}$, $I_{0\max}$, and n_{rnd} using the local energy distribution. As the three hyperparameters are common for all the spins, as shown in (2), the mean of the local energy distributions are considered. In other words, a normal distribution $N(\mathrm{mean}(|\mu_i|), \mathrm{mean}(s_i^2))$ is used to control I_0 that commonly controls for all the spins.

TABLE 1. Summary of the Determined Hyperparameters for SSA

μ_i	$(n-1)\cdot \mathrm{mean}(J_{i:})$
s_i	$\sqrt{(n-1)\cdot \mathrm{Var}([J_{i:}; -J_{i:}])}$
n_{rnd}	$0.6745\cdot \mathrm{mean}(s_i)$
I_{0min}	$0.01\cdot \max(s_i) + \min(\lvert\mu_i\rvert)$
I_{0max}	$2\cdot \max(s_i) + \min(\lvert\mu_i\rvert)$
β	$\left(\frac{I_{0min}}{I_{0max}}\right)^{\left(\frac{1}{cycle-1}\right)}$

First, the magnitude of noise signals n_{rnd} is determined as follows:

$$n_{\mathrm{rnd}} = 0.6745\cdot \mathrm{mean}(s_i). \tag{5}$$

The reason to select 0.6745 is that the point where the area from the mean of the normal distribution equals 50% is at $\pm 0.6745\cdot\sigma$ from the mean. It means that the local energy distribution can be equally divided into two area by n_{rnd}. Second, the minimum pseudo inverse temperature $I_{0\min}$ is determined as follows:

$$I_{0\min} = 0.01\cdot \max(s_i) + \min(|\mu_i|). \tag{6}$$

$I_{0\min}$ is set as close to $\min(|\mu_i|)$ as possible to facilitate the flipping of the spin states. Third, the maximum pseudo inverse temperature $I_{0\max}$ is determined as follows:

$$I_{0\max} = 2\cdot \max(s_i) + \min(|\mu_i|). \tag{7}$$

It means that a cumulative probability within a distance of $2\cdot\max(s_i)$ from the mean is approximately 95%. The reasoning behind the selection of max/min functions originates from the local energy distribution. Each local distribution at its spin has different s_i and μ_i, varying from small to large values. The usage of the max and min functions ensures that I_0 can encompass all local energy distributions by changing $I_{0\min}$ to $I_{0\max}$ during the annealing process.

Table 1 summarizes the determined hyperparameters for SSA. β is determined by $(I_{0\min}/I_{0\max})^{\left(\frac{1}{cycle-1}\right)}$ as I_0 is updated by $I_0(t+1) = I_0(t)/\beta$, as shown in Fig. 2. Note that $cycle$ is the total number of cycles during the annealing process. The determined hyperparameters will be evaluated in Section V.

IV. SSA WITH UNIQUE NOISE MAGNITUDE (SSAU)

In SSA, the magnitude of the noise signals n_{rnd} is common for all spins, as shown in Fig. 5(a). However, each spin has a unique local energy distribution that is different from others. To properly control the noise magnitude for each spin, SSA with a unique noise magnitude (SSAU) is introduced as an extension of SSA. This approach aims to better control the noise level for each spin and improve the search performance.

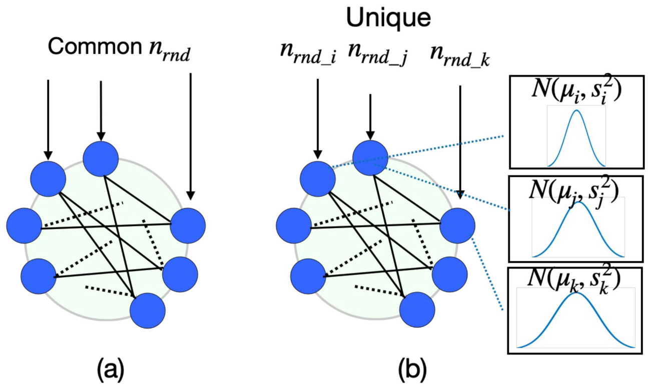


FIGURE 5. Magnitude of random signals: (a) SSA and (b) SSAU. In SSAU, a unique noise magnitude n_{rnd_i} is applied for each spin depending on its local energy distribution.

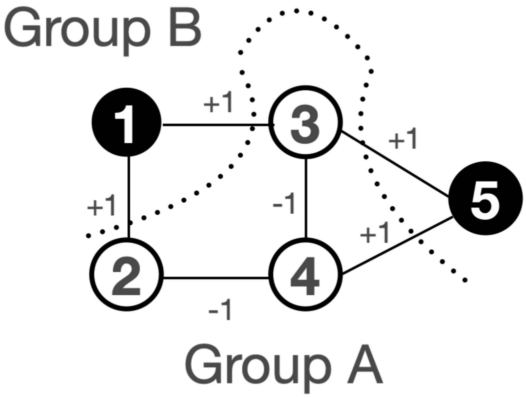


FIGURE 6. Example of a five-node MAX-CUT problem with edge weights of −1 and +1. The line cuts the edges to divide the graph into two groups while the sum of the edge weights is maximized.

Fig. 5(b) shows SSAU with a unique noise magnitude n_{rnd_i} for the i-th spin. The noise magnitude for SSA in (5) is updated for SSAU as follows:

$$n_{rnd_i} = 0.6745 \cdot s_i, \tag{8}$$

where s_i is the standard deviation of the local energy distribution at i-th spin in (4). The other hyperparameters for SSAU are the same as that of SSA as shown in Table 1.

Using n_{rnd_i}, (2a) is replaced by the following equation:

$$I_i(t+1) = h_i + \sum_j J_{ij} \cdot \sigma_j(t) + n_{rnd\_i} \cdot r_i(t). \tag{9}$$

The other equations of (2) are common for SSA and SSAU.

V. EVALUATION

A. SIMULATION SETUP

The proposed statistical methods for determining hyperparameters are evaluated on maximum-cut (MAX-CUT) problems, which are a typical class of combinatorial optimization problems [16]. An example of a five-node MAX-CUT problem with edge weights of −1 and +1 is shown in Fig. 6.

TABLE 2. Summary of MAX-CUT Problems Used for Evaluation

Graph	# nodes	Structure	Weights (J_{ij})	# edges ($J_{ij} \neq 0$)
G1	800	random	$\{+1, 0\}$	19 176
G6	800	random	$\{+1, 0, -1\}$	19 176
G11	800	toroidal	$\{+1, 0, -1\}$	1600
G14	800	planar	$\{+1, 0\}$	4694
G18	800	planar	$\{+1, 0, -1\}$	4694
G22	2000	random	$\{+1, 0\}$	19 990
G34	2000	toroidal	$\{+1, 0, -1\}$	4000
G38	2000	planar	$\{+1, 0\}$	11 779
G39	2000	planar	$\{+1, 0, -1\}$	11 778
G47	1000	random	$\{+1, 0\}$	9990
G48	3000	toroidal	$\{+1, 0, -1\}$	6000
G54	1000	random	$\{+1, 0\}$	5916
G55	5000	random	$\{+1, 0\}$	12 498
G56	5000	random	$\{+1, 0, -1\}$	12 498
G58	5000	planar	$\{+1, 0\}$	29 570
K2000	2000	full	$\{+1, -1\}$	1 999 000

The objective of MAX-CUT is to maximize the sum of edge weights by dividing the graph into two groups through a line cut. The annealing process changes the spin states in an attempt to reach the global minimum energy described in (1), where the maximum cut value is corresponding to the global minimum energy. The black circle illustrates a spin state of '+1', while the white circle illustrates a spin state of '−1'. In this example, the graph is divided into Group A (nodes 2, 3, and 4) and Group B (nodes 1 and 5), with a sum of edge weights equal to 4. The edge weights are represented using J in the Hamiltonian shown in (1).

Table 2 summarizes the benchmarks for the MAX-CUT problems that are used to evaluate the proposed method. The Gset includes the Gx graphs with different sizes, shapes, and weights [17], while K2000 is another benchmark that consists of fully-connected graphs with 2,000 nodes [18]. All simulation results are obtained using Python 3.9.6 on an Apple M1 Ultra with 128 GB of memory.

B. DETERMINED HYPERPARAMETERS

Table 3 summarizes the hyperparameters used for SSA and SSAU. The equations for calculating these hyperparameters are provided in Table 1 and (8). To illustrate how these values are determined, let us consider the example of G1. The matrix J of G1 is an 800×800 matrix. It consists of edge weights either '+1' or '0', with the count of '+1' edge weights being 19,176. The total number of edges in the matrix amounts to 319,600. First, $|\mu_i|$ and s_i are calculated for each of the 800 vectors in J, which yields a unique set of values for each spin. As an example, two normal distributions with the minimum and maximum values of μ are shown in Fig. 7(a). The range of $|\mu_i|$ is from 26.97 to 66.92, and the range of s_i is from 5.19 to 8.18, as shown in Fig. 7(b).

TABLE 3. Summary of Hyperparameters Determined for SSA and SSAU

Graph	$\|\mu_i\|$	s_i	SSA			SSAU
			n_{rnd}	I_{0min}	I_{0max}	n_{rnd_i}
G1	[26.97, 66.92]	[5.19, 8.18]	4.66	27.05	43.33	[3.50, 5.52]
G6	[0.00, 28.96]	[5.19, 8.18]	4.66	0.08	16.36	[3.50, 5.52]
G11	[0.00, 3.99]	[1.99, 1.99]	1.35	0.02	3.99	[1.35, 1.35]
G14	[4.99, 131.84]	[2.23, 11.48]	2.18	5.11	27.96	[1.50, 7.74]
G18	[0.00, 17.98]	[2.23, 11.48]	2.18	0.11	22.96	[1.50, 7.74]
G22	[6.99, 36.98]	[2.64, 6.08]	2.99	7.05	19.16	[1.78, 4.10]
G34	[0.00, 3.99]	[1.99, 1.99]	1.35	0.02	3.99	[1.35, 1.35]
G38	[3.99, 248.88]	[1.99, 15.78]	2.17	4.16	35.55	[1.35, 10.64]
G39	[0.00, 42.98]	[1.99, 14.49]	2.17	0.14	28.97	[1.35, 9.77]
G47	[7.99, 33.97]	[2.83, 5.83]	2.99	8.05	19.64	[1.90, 3.93]
G48	[3.99, 3.99]	[1.99, 1.99]	1.35	4.02	7.99	[1.35, 1.35]
G54	[4.99, 135.86]	[2.23, 11.66]	2.18	5.11	28.30	[1.51, 7.86]
G55	[0.00, 14.99]	[0.00, 3.87]	1.46	0.03	7.75	[0.00, 2.61]
G56	[0.00, 9.99]	[0.00, 3.87]	1.46	0.03	7.75	[0.00, 2.61]
G58	[3.99, 560.88]	[1.99, 23.68]	2.17	4.24	51.36	[1.35, 15.97]
K2000	[0.99, 168.92]	[44.70, 44.70]	30.15	1.45	90.40	[30.15, 30.15]

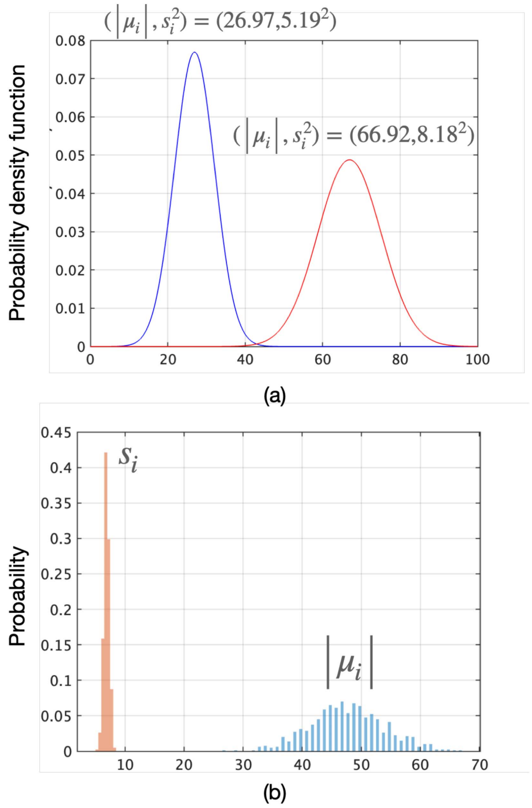


FIGURE 7. $|\mu_i|$ and s_i for G1: (a) two normal distribution with the minimum and the maximum $|\mu_i|$ as an example and (b) the histogram of $|\mu_i|$ and s_i for all 800 spins.

Next, $|\mu_i|$ and s_i are used to calculate the hyperparameters of n_{rnd}, $I_{0\mathrm{min}}$, and $I_{0\mathrm{max}}$. In the case of SSAU, each spin has a unique n_{rnd_i} value, which corresponds to its local energy distribution. The range of n_{rnd_i} is from 3.50 to 5.52.

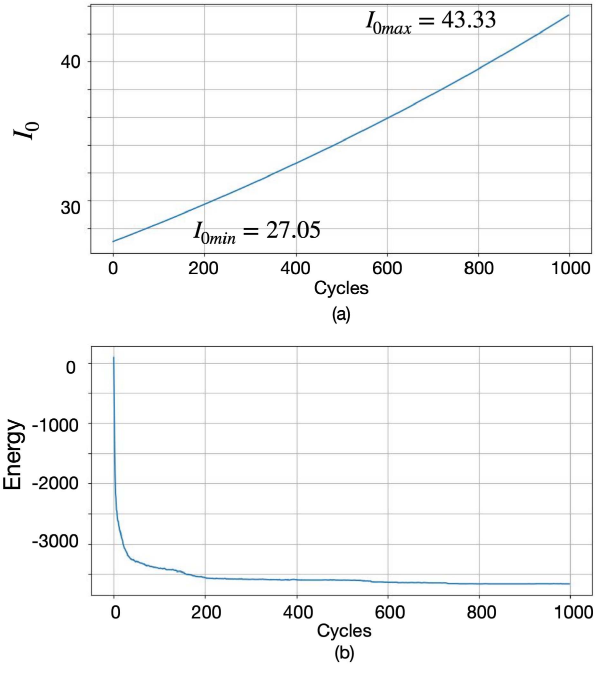


FIGURE 8. Simulation example of SSA with determined hyperparameters for G1: (a) I_0 vs. cycles and (b) energy vs. cycles. The energy decreases to the global minimum energy as I_0 is increased from $I_{0\mathrm{min}}$ and $I_{0\mathrm{max}}$.

To verify the hyperparameters, SSA with G1 is simulated as an example. Fig. 8(a) illustrates the pseudo inverse-temperature transition for 1,000 cycles. I_0 is gradually increased from $I_{0\mathrm{min}} = 27.05$ to $I_{0\mathrm{max}} = 43.33$. The value of I_0 is updated by $I_0(t+1) = I_0(t)/\beta$, where $\beta = 0.99952$. The energy, defined by (1), decreases to the global minimum as I_0 increases. Once the simulation completes 1,000 cycles, the spin states σ_i are extracted to compute the cut value using J.

TABLE 4. **Comparisons of Performance on the MAX-CUT Benchmarks for 1000 Cycles**

Graph	Best known	Mean of cut values			Standard deviation of cut values			Ratio of mean cut value		
	cut value	SA [5]	SSA	SSAU	SA	SSA	SSAU	SSA	SSAU	SSAU
								vs. SA	vs. SA	vs. SSA
G1	800	10754.52	11427.05	11428.13	47.63	32.83	33.28	6.25%	6.26%	0.01%
G6	800	1276.96	2159.59	2160.63	45.56	12.29	12.44	69.12%	69.20%	0.05%
G11	800	334.98	549.60	549.47	13.16	4.12	4.10	64.07%	64.03%	-0.02%
G14	800	2803.34	3009.71	3013.20	15.55	8.13	7.70	7.36%	7.49%	0.12%
G18	800	590.68	972.49	974.72	25.60	8.34	7.56	64.64%	65.02%	0.23%
G22	2000	11161.3	13099.78	13102.26	50.47	31.63	31.34	17.37%	17.39%	0.02%
G34	2000	469.26	1346.64	1346.67	27.97	6.34	6.39	186.97%	186.98%	0.00%
G38	2000	6642.68	7546.93	7554.4	31.09	15.82	15.71	13.61%	13.73%	0.10%
G39	2000	862.1	2352.47	2362.03	46.48	14.01	12.41	172.88%	173.99%	0.41%
G47	1000	5854.9	6536.24	6537.83	33.10	21.62	21.15	11.64%	11.66%	0.02%
G48	3000	3562.2	5724.25	5724.29	30.94	38.20	38.15	60.69%	60.70%	0.00%
G54	1000	3484.95	3780.36	3784.71	19.66	8.73	8.54	8.48%	8.60%	0.11%
G55	5000	6973.55	9994.42	10037.44	45.21	22.33	21.02	43.32%	43.94%	0.43%
G56	5000	702.21	3930.36	3947.27	55.74	14.72	12.08	459.71%	462.12%	0.43%
G58	5000	15791.8	18930.60	18949.24	68.81	27.41	30.40	19.88%	19.99%	0.10%
K2000	2000	11267.73	32931.54	32932.38	566.15	117.59	117.71	192.26%	192.27%	0.00%

C. COMPARISONS

Table 4 compares SSA and SSAU with a conventional SA in the 16 different MAX-CUT problems for 1,000 cycles. Let us briefly explain the simulation conditions for SA, as detailed in [5]. In this method, the temperature T is gradually decreased by Δ_{IT} as $T \leftarrow 1/(1/T + \Delta_{IT})$ at each cycle. The initial temperature is set to 1, and the final temperature is set to 1/1000. The number of cycles is 1,000, which is the same as for SSA and SSAU. During each cycle, a spin state is randomly flipped, and a new state is accepted if the new energy (E_{new}) is lower than the current energy (E_{cur}) or if it is higher with a probability of $\exp(-(E_{new} - E_{cur})/T)$.

The mean and the standard deviation of cut values are obtained by running 100 trials for each benchmark. When comparing SSA and SSAU with SA, the ratios of the mean cut values are positive in all benchmarks. In particular, the ratios are significantly positive in the case of graphs with $\{+1, 0, -1\}$ weights. The reason is that SA takes more cycles to achieve better cut values in the case, where the mean cut values are significantly smaller than the best-known values. The mean of normalized cut values is 65.0% for SA.

When comparing SSA and SSAU, the mean cut values of SSAU are larger than that of SSA in most of the benchmarks. The ratios are relatively large in the case of graphs with $\{+1, 0\}$ weights. The standard deviation of the cut values is similar for both methods, with these values significantly smaller than that of SA. On average, across all benchmarks for 1,000 cycles, the mean cut values of SSA and SSAU are 97.9% and 98.0% of the best-known cut values, respectively. In terms of computation cost, SSAU takes almost the same simulation time as SSA. For instance, the simulation time on K2000 for 1,000 cycles is 1.43 seconds for both SSA and SSAU. The best cut values are summarized in Table 5. It shows that SSA and SSAU achieve 99.2% of the best-known cut values while SA only attains 67.7%.

TABLE 5. **Summary of Best Cut Values for 1,000 Cycles**

	SA	SSA	SSAU
G1	10859	11550	11560
G6	1389	2178	2178
G11	364	564	564
G14	2836	3040	3040
G18	647	990	992
G22	11325	13214	13221
G34	536	1368	1368
G38	6715	7603	7606
G39	993	2399	2400
G47	5947	6647	6615
G48	3642	5934	5904
G54	3530	3820	3815
G55	7088	10073	10121
G56	831	3982	3996
G58	15919	19014	19033
K2000	12413	33235	33225

Fig. 9 displays the mean of normalized cut values in relation to the number of cycles for 16 benchmarks. Both SSA and SSAU quickly achieve an average of 95.4% of the normalized best-known value within just 100 cycles, while SA only reaches 48.2%. For 10,000 cycles, SA, SSA, and SSAU achieve 90.0%, 98.8%, and 98.9% respectively. Compared to related works on K2000, the mean cut value of SSAU is 32,932, which surpasses the 32,458 and 32,768 values from [18] and [30], respectively.

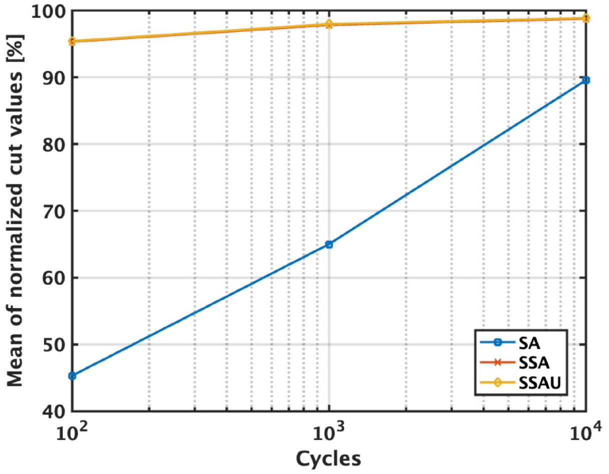


FIGURE 9. Mean of normalized cut values versus cycles for 16 benchmarks. It shows that both SSA and SSAU rapidly attain an average of 95.4% of the best-known normalized value within just 100 cycles. In comparison, SA only achieves 48.2%.

TABLE 6. Time Complexity of Hyperparameter Search in SSA

	Conventional [10], [24]	This work
Time complexity	$\mathcal{O}(n^3)$	$\mathcal{O}(1)$

VI. DISCUSSION

A. COMPARISON WITH HYPERPARAMETER SEARCH

The comparison result between SA and SSA for G11 was previously presented in [10]. SSA achieved superior mean cut values compared to SA and QA, even though the previous study employed hyperparameter searching. Table 6 provides a comparison of the time complexity of hyperparameter searching for SSA. In the conventional method [10], [24], the hyperparameters of n_{rnd}, $I_{0\min}$, and $I_{0\max}$ were selected through searching, leading to a time complexity of $\mathcal{O}(n^3)$. In contrast, the proposed method determines these hyperparameters statistically without the need for searching, thereby resulting in a time complexity of $\mathcal{O}(1)$.

Let us discuss the impact of determined hyperparameters in SSA. As summarized in Table 6, the conventional method necessitates the search for hyperparameters prior to simulated annealing. Table 7 compares the search and annealing time, along with the normalized mean cut values for the all 16 benchmarks using SSA, where the annealing takes 1,000 cycles. In the conventional method, the hyperparameters of $I_{0\min}$, $I_{0\max}$, and n_{rnd} are randomly searched for across 1,000 trials, with each value ranging from 0 to 1000 with a condition of $I_{0\min} \geq I_{0\min}$. Based on these search results, the best hyperparameters are selected to evaluate the mean cut values. The outcome indicates that the proposed method, which excludes the hyperparameter search, yields results that are superior to the conventional method in most benchmarks. Moreover, the proposed method completely eliminates the time-consuming process of hyperparameter searching.

Fig. 10 illustrates the normalized cut values using random search with 1,000 cycles on G58. In G58, the proposed method achieves a normalized mean cut value of 98.2%, which is superior to the 97.6% of the conventional method. Both the best hyperparameters identified through searching and the determined hyperparameters are plotted. The plotted values for n_{rnd} and $I_{0\min}$ are similar between the conventional and proposed methods; however, the values for $I_{0\max}$ differ significantly, potentially leading to performance loss. As the search process iterates 1,000 times, more iterations might yield better hyperparameters, but this would also increase the search time.

B. HYPERPARAMETER SELECTION

Let us discuss the effect of using 0.01 for $I_{0\min}$ in (6). In this study, the constant 0.01 is utilized to position $I_{0\min}$ close to $\min(|\mu_i|)$. This results in a mean normalized cut value of 97.9% on average in SSA. When 0.01 is changed to 10^{-6} or 0.05, the respective averages become 97.6% and 97.8%. Hence, the constant for $I_{0\min}$ does not significantly impact the results. Note that a value of 0 is not permissible. The range of this value is associated with the equation $\beta = \left(\frac{I_{0\min}}{I_{0\max}}\right)^{\left(\frac{1}{cycle-1}\right)}$ as shown in Table 1. If the value is 0, $I_{0\min}$ becomes 0 in several benchmarks, especially when $\min(|\mu_i|)$ is 0. This makes it impossible to determine β. Therefore, $I_{0\min}$ is set to a value as low as 10^{-6}.

Additionally, the number of standard deviations set to 2 for $I_{0\max}$ in (7) is examined. To evaluate the impact of this standard deviation number, it is altered to 3 and 4. As a result, the mean normalized cut value averages 97.8% in SSA for both cases. Therefore, the constant for $I_{0\max}$ also does not have a significant effect on the results.

There might be a constraint of the proposed determined hyperparameters based on CLT when each spin contains a few connections. To assess the performance of the proposed method based on the number of connections, new benchmarks are created using K2000 as a reference. K2000 is fully connected with weights of +1 or −1, and the ratio of +1 to −1 weights is 50%. These benchmarks encompass spin counts ranging from 5 to 1000, with each spin being fully connected and assigned weights of either +1 or −1. The mean cut values, taken from 100 trials, are contrasted between the conventional search and the proposed method in Fig. 11. The conventional method searches the best hyperparameters for 1,000 trials. The findings indicate that the proposed method outperforms the conventional approach when there is a larger number of connections. However, with a smaller number of connections, the assumptions underpinning the Central Limit Theorem (CLT) might not be met, even though the mean cut values remain nearly identical. A possible explanation is that annealing is effective with a smaller number of spins, even in the absence of precise hyperparameters.

TABLE 7. Comparisons of Search and Annealing Time With Normalized Mean Cut Values for 1000 Cycles on All 16 Benchmarks Using SSA

	Conventional			This work	Normalized mean cut value		
	Search time [s]	Annealing time [s]	Total time [s]	Annealing time (total time) [s]	Conventional	This work	Ratio
G1	310.61	0.28	310.88	0.40	98.58%	98.31%	-0.28%
G6	286.76	0.26	287.02	0.28	89.51%	99.15%	10.78%
G11	313.34	0.28	313.62	0.40	88.1%4	97.45%	10.56%
G14	312.60	0.25	312.85	0.39	97.44%	98.23%	0.81%
G18	316.05	0.28	316.32	0.29	87.90%	98.03%	11.53%
G22	1523.29	1.45	1524.74	1.45	97.4%2	98.06%	0.66%
G34	1494.86	1.45	1496.30	1.80	94.66%	97.30%	2.79%
G38	1483.77	1.45	1485.21	1.83	96.95%	98.17%	1.26%
G39	1473.18	1.45	1474.63	1.83	87.2%3	97.69%	12.00%
G47	433.96	0.41	434.37	0.61	96.92%	98.18%	1.31%
G48	3250.20	3.26	3253.46	4.19	88.33%	95.40 %	8.01%
G54	418.73	0.40	419.12	0.61	97.81%	98.14%	0.34%
G55	8812.68	8.96	8821.64	10.71	94.68%	97.04%	2.49%
G56	8869.97	8.80	8878.77	9.30	94.24%	97.84%	3.83%
G58	8794.97	8.82	8803.79	9.24	97.57%	98.12%	0.57%
K2000	1451.41	1.44	1452.85	1.52	98.24%	98.78%	0.56%

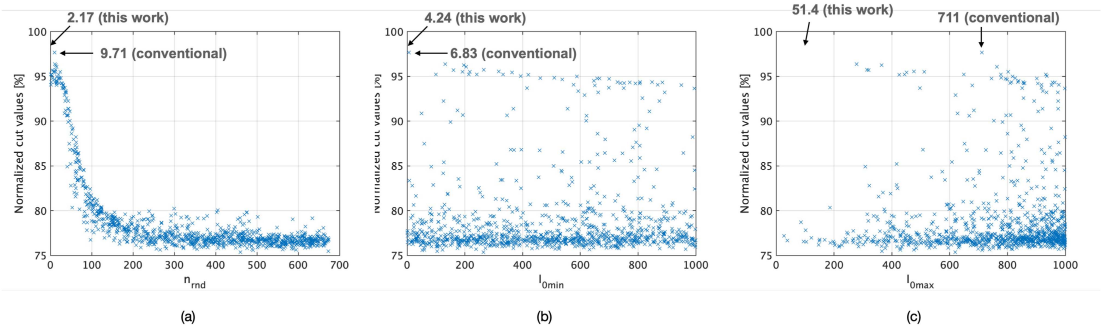


FIGURE 10. 1000 normalized cut values using random search for 1000 cycles on G58 are presented as: (a) n_{rnd}, (b) $I_{0\mathrm{min}}$, and (c) $I_{0\mathrm{max}}$. The conventional method achieves a normalized mean cut value of 97.6% with the best hyperparameters searched, while the proposed method achieves 98.2% with the determined hyperparameters. Both sets of parameters are plotted in the figures.

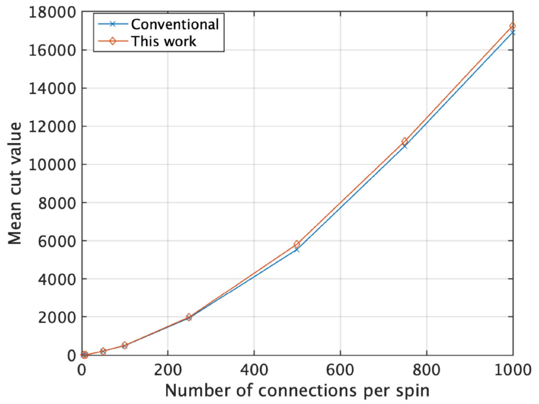


FIGURE 11. Mean cut values versus the number of connections per spins in fully-connected benchmark graphs with weights of +1 or −1.

VII. CONCLUSION

This paper proposes the local energy distribution based hyperparameter determination in SSA, which is a faster solving method for combinatorial optimization problems than SA. The method is based on the local energy distributions of spins and utilizes the CLT-based normal distribution for hyperparameter determination, significantly reducing the time complexity for hyperparameter search. Additionally, using the local energy distributions, the unique magnitude for the random signal at each spin has been presented to further improve SSA. The proposed method is evaluated in the MAX-CUT problems, where it achieves high accuracy while reducing time costs compared to conventional SSA with hyperparameter search.

Future research directions include applying the proposed method to other optimization problems, exploring the impact of hyperparameters on the SSA performance, developing more efficient hardware.

REFERENCES

[1] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, “Quantum annealing for industry applications: Introduction and review,” *Rep. Prog. Phys.*, vol. 85, no. 10, Sep. 2022, Art. no. 104001, doi: 10.1088/1361-6633/ac8c54.

[2] E. E. Reiter and C. M. Johnson, *Limits of Computation: An Introduction to the Undecidable and the Intractable*. Boca Raton, FL, USA: CRC Press, 2012.

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

[4] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon, “Optimization by simulated annealing: An experimental evaluation; part II, graph coloring and number partitioning,” *Operations Res.*, vol. 39, no. 3, pp. 378–406, 1981.

[5] T. Mykleburst, “Solving maximum cut problems by simulated annealing,” 2015, *arXiv:1505.03068*.

[6] T. Kadowaki and H. Nishimori, “Quantum annealing in the transverse Ising model,” *Phys. Rev. E*, vol. 58, no. 5, pp. 5355–5363, 1998.

[7] S. Boixo et al., “Evidence for quantum annealing with more than one hundred qubits,” *Nature Phys.*, vol. 10, no. 3, pp. 218–224, 2014, doi: 10.1038/nphys2900.

[8] H. Neven, “When can quantum annealing win?,” Jan. 2016. [Online]. Available: https://ai.googleblog.com/2015/12/when-can-quantum-annealing-win.html

[9] A. D. King, “Quantum critical dynamics in a 5,000-qubit programmable spin glass,” *Nature*, vol. 617, no. 7959, pp. 61–66, 2023.

[10] N. Onizawa, K. Katsuki, D. Shin, W. J. Gross, and T. Hanyu, “Fast-converging simulated annealing for Ising models based on integral stochastic computing,” *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 28, 2022, doi: 10.1109/TNNLS.2022.3159713.

[11] K. Camsari, R. Faria, B. Sutton, and S. Datta, “Stochastic p-bits for invertible logic,” *Phys. Rev. X*, vol. 7, Jul. 2017, Art. no. 031014.

[12] B. R. Gaines, “Stochastic computing systems,” *Adv. Inf. Syst. Sci.*, vol. 2, no. 2, pp. 37–172, 1969.

[13] B. D. Brown and H. C. Card, “Stochastic neural computation. I. Computational elements,” *IEEE Trans. Comput.*, vol. 50, no. 9, pp. 891–905, Sep. 2001.

[14] D. Shin, N. Onizawa, W. J. Gross, and T. Hanyu, “Memory-efficient FPGA implementation of stochastic simulated annealing,” *IEEE Trans. Emerg. Sel. Topics Circuits Syst.*, vol. 13, no. 1, pp. 108–118, Mar. 2023.

[15] A. Lucas, “Ising formulations of many NP problems,” *Front. Phys.*, vol. 2, 2014, Art. no. 5. [Online]. Available: https://www.frontiersin.org/article/10.3389/fphy.2014.00005

[16] S. Burer, R. D. C. Monteiro, and Y. Zhang, “Rank-two relaxation heuristics for MAX-CUT and other binary quadratic programs,” *SIAM J. Optim.*, vol. 12, no. 2, pp. 503–521, 2001.

[17] Y. Ye, “Computational optimization laboratory,” 1999. [Online] .Available: http://web.stanford.edu/yyye/Col.htm

[18] T. Inagaki et al., “A coherent Ising machine for 2000-node optimization problems,” *Science*, vol. 354, no. 6312, pp. 603–606, 2016.

[19] H. Gyoten, M. Hiromoto, and T. Sato, “Enhancing the solution quality of hardware Ising-model solver via parallel tempering,” in *Proc. Int. Conf. Comput.-Aided Des.*, 2018, pp. 1–8, doi: 10.1145/3240765.3240806.

[20] K. Y. Camsari, B. M. Sutton, and S. Datta, “P-bits for probabilistic spin logic,” *Appl. Phys. Rev.*, vol. 6, 2019, Art. no. 011305.

[21] S. C. Smithson, N. Onizawa, B. H. Meyer, W. J. Gross, and T. Hanyu, “Efficient CMOS invertible logic using stochastic computing,” *IEEE Trans. Circuits Syst. I, Regular Papers*, vol. 66, no. 6, pp. 2263–2274, Jun. 2019.

[22] N. Onizawa, S. C. Smithson, B. H. Meyer, W. J. Gross, and T. Hanyu, “In-hardware training chip based on CMOS invertible logic for machine learning,” *IEEE Trans. Circuits Syst. I, Regular Papers*, vol. 67, no. 5, pp. 1541–1550, May 2020.

[23] G. E. Hinton, T. J. Sejnowski, and D. H. Ackley, “Boltzmann machines: Constraint satisfaction networks that learn,” Dept. Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-84-119, 1984.

[24] K. Katsuki, D. Shin, N. Onizawa, and T. Hanyu, “Fast solving complete 2000-node optimization using stochastic-computing simulated annealing,” in *Proc. 29th IEEE Int. Conf. Electron. Circuits Syst.*, 2022, pp. 1–4.

[25] K. M. Zick, O. Shehab, and M. French, “Experimental quantum annealing: Case study involving the graph isomorphism problem,” *Sci. Rep.*, vol. 5, no. 1, 2015, Art. no. 11168, doi: 10.1038/srep11168.

[26] A. Ardakani, F. Leduc-Primeau, N. Onizawa, T. Hanyu, and W. J. Gross, “VLSI implementation of deep neural network using integral stochastic computing,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 10, pp. 2588–2599, Oct. 2017.

[27] V. C. Gaudet and A. C. Rapley, “Iterative decoding using stochastic computation,” *Electron. Lett.*, vol. 39, no. 3, pp. 299–301, Feb. 2003.

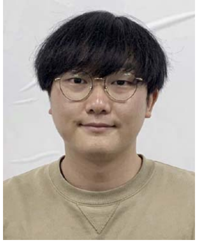
[28] P. Li, D. J. Lilja, W. Qian, K. Bazargan, and M. D. Riedel, “Computation on stochastic bit streams digital image processing case studies,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 3, pp. 449–462, Mar. 2014.

[29] Y. Liu and K. K. Parhi, “Architectures for recursive digital filters using stochastic computing,” *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3705–3718, Jul. 2016.

[30] H. Goto, K. Tatsumura, and A. R. Dixon, “Combinatorial optimization by simulating adiabatic bifurcations in nonlinear hamiltonian systems,” *Sci. Adv.*, vol. 5, no. 4, 2019, Art. no. eaav2372. [Online]. Available: https://www.science.org/doi/abs/10.1126/sciadv.aav2372

NAOYA ONIZAWA (Member, IEEE) received the D.E. degree in electrical and communication engineering from Tohoku University, Sendai, Japan, in 2009. He is currently an Associate Professor and a Distinguished Researcher with the Research Institute of Electrical Communication, Tohoku University. He was a Postdoctoral Fellow with the University of Waterloo, Waterloo, ON, Canada, in 2011 and McGill University, Montreal, QC, Canada, from 2011 to 2013. He was a Visiting Associate Professor with the University of Southern Brittany, Lorient, France, and IMT Atlantique, France in 2015 and 2022, respectively. His research interests include the energy-efficient VLSI design based on asynchronous circuits and probabilistic computation, and their applications, such as brain-like computers. He was the recipient of the Best Paper Award in 2010 IEEE ISVLSI, Best Paper Finalist in 2014 IEEE ASYNC, Kenneth C. Smith Early Career Award for Microelectronics Research in 2016 IEEE ISMVL, MEXT Young Scientists’ Prize, Japan in 2020, and 16th Aoba Engineering Promotion Society Award in 2022.

KYO KUROKI received the B.E. degree from the School of Engineering, Tohoku University, Sendai, Japan, in 2023. His research interests include stochastic simulated annealing and related applications.



DUCKGYU SHIN received the B.E. and M.E. degrees in electrical and communication engineering in 2019 and 2021, respectively, from Tohoku University, Sendai, Japan, where he is currently working toward the D.E. degree with the Research Institute of Electrical Communication. His research interests include high speed VLSI design based on stochastic simulated annealing and related applications such as implementation of a fast Ising processor.

TAKAHIRO HANYU (Senior Member, IEEE) received the B.E., M.E. and D.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1984, 1986 and 1989, respectively. He is currently a Professor and Vice Director with the Research Institute of Electrical Communication, Tohoku University. His research interests include nonvolatile logic circuits and their applications to ultra-low-power and/or highly dependable VLSI processors, and post-binary computing and its application to brain-inspired VLSI systems. He was the recipient of the Sakai Memorial Award from the Information Processing Society of Japan in 2000, the Judge’s Special Award at the 9th LSI Design of the Year from the Semiconductor Industry News of Japan in 2002, Special Feature Award at the University LSI Design Contest from ASP-DAC in 2007, APEX Paper Award of Japan Society of Applied Physics in 2009, Excellent Paper Award of IEICE, Japan, in 2010, Ichimura Academic Award in 2010, Best Paper Award of IEEE ISVLSI 2010, the Paper Award of SSDM 2012, Best Paper Finalist of IEEE ASYNC 2014, and Commendation for Science and Technology by MEXT, Japan in 2015.