IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 23, 2021, accepted May 3, 2021, date of publication May 19, 2021, date of current version June 10, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3081685

Performance Comparison of Typical Binary-Integer Encodings in an Ising Machine

KENSUKE TAMURA[1], TATSUHIKO SHIRAI[2], HOSHO KATSURA[1,3], SHU TANAKA[4,5], AND NOZOMU TOGAWA[2], (Member, IEEE)

[1]Department of Physics, Graduate School of Science, The University of Tokyo, Tokyo 113-0033, Japan
[2]Department of Computer Science and Communications Engineering, Waseda University, Tokyo 169-8555, Japan
[3]Institute for Physics of Intelligence, The University of Tokyo, Tokyo 113-0033, Japan
[4]Department of Applied Physics and Physico-Informatics, Keio University, Yokohama 223-8522, Japan
[5]Green Computing System Research Organization, Waseda University, Tokyo 162-0042, Japan

Corresponding author: Kensuke Tamura (tamura-kensuke265@g.ecc.u-tokyo.ac.jp)

ABSTRACT The differences in performance among binary-integer encodings in an Ising machine, which can solve combinatorial optimization problems, are investigated. Many combinatorial optimization problems can be mapped to find the lowest-energy (ground) state of an Ising model or its equivalent model, the Quadratic Unconstrained Binary Optimization (QUBO). Since the Ising model and QUBO consist of binary variables, they often express integers as binary when using Ising machines. A typical example is the combinatorial optimization problem under inequality constraints. Here, the quadratic knapsack problem is adopted as a prototypical problem with an inequality constraint. It is solved using typical binary-integer encodings: one-hot encoding, binary encoding, and unary encoding. Unary encoding shows the best performance for large-sized problems.

INDEX TERMS Ising machine, combinatorial optimization problem, Ising model, quadratic unconstrained binary optimization, binary-integer encoding, quadratic knapsack problem.

I. INTRODUCTION

A. MOTIVATION

Combinatorial optimization problems find the combination of decision variables that minimize or maximize an objective function under given constraints. Most problems are known as NP-hard or NP-complete [1], and the number of solution candidates exponentially increases with the number of decision variables. Because combinatorial optimization problems are ubiquitous in social life and industry [2]–[12], there is a growing interest in developing technologies that can efficiently and accurately find optimal or quasi-optimal solutions. Ising machines have recently attracted attention as efficient solvers that achieve faster computations than conventional digital computers with the von Neumann architecture [13]–[25]. Many underlying algorithms for Ising machines have been proposed [26]–[28]. Recently, proposals have also been made for efficient input formats to Ising machines based on the operation principle of Ising machines [29]. Examples include simulated annealing (SA)-based machines [13]–[17], quantum annealing machines [18]–[20], and photonics-based machines [21]–[24]. The features of the various Ising machines are summarized in [30].

In an Ising machine, each combinatorial optimization problem is mapped onto the Ising problem. The Ising problem searches for the lowest-energy (ground) state of the Ising model or its equivalent model, the Quadratic Unconstrained Binary Optimization (QUBO) model [31]. The Ising model was originally introduced in statistical mechanics to describe the nature of phase transition [32]. The energy functions of the Ising model and the QUBO model are described by the quadratic form of spin variables $\{+1, -1\}$ and binary variables $\{0, 1\}$, respectively. The objective function and the constraints in a combinatorial optimization problem are encoded in the Ising model or the QUBO model [33] and [34]. Since the Ising model and QUBO consist of binary variables, when solving combinatorial optimization problems involving integer values with Ising machines, the integer values must be expressed in binary variables.

The associate editor coordinating the review of this manuscript and approving it for publication was Alba Amato.

Herein combinatorial optimization problems subject to an inequality constraint are considered. To write the inequality constraint as an energy function of the Ising model, binary-integer encoding has been used to represent integers in terms of binary variables [3], [33], and [35]–[42]. Various types of binary-integer encoding exist. From a practical viewpoint, encoding with the best performance should be adopted. However, the performance of different binary-integer encodings has yet to be systematically investigated in Ising machines.

B. SUMMARY OF CONTRIBUTIONS

This paper compares the performance among typical binary-integer encodings in an SA-based Ising machine called Digital Annealer (DA) [15]. The main contributions are:

- We provide a systematic method to compare the binary-integer encoding performance. Our method is applied to the quadratic knapsack problem (QKP) [43], [44], which is a prototypical combinatorial optimization problem with an inequality constraint. The QKP is an extension of the well-known combinatorial optimization problem, the knapsack problem (KP) [35], [41], [45]–[47], except that it allows for quadratic terms in the objective function. The QKP is better suited than the KP to study the performance of binary-integer encodings since the objective function is given in the general function form of the QUBO model.
- We investigate the performance of common types of binary-integer encoding: one-hot encoding, binary encoding, and unary encoding. Unary encoding achieves the best performance for large-sized QKPs.

The rest of this paper is organized as follows. Section II introduces the QKP. Section III provides the three types of binary-integer encoding. Here, the QKP is rephrased as the QUBO model. Section IV shows methods to prepare the QKP instances and the setup for DA. Section V presents our results. Section VI discusses the results and section VII concludes our study.

II. QUADRATIC KNAPSACK PROBLEM

This section reviews the formulation of the QKP, or more precisely the $0-1$ QKP [43]. Informally, the QKP maximizes the overall profit in a knapsack so that the overall weight does not exceed a given knapsack's capacity. We assume that there are n items where the item i has a weight $w_i > 0$ for $i = 0, 1, \ldots, n-1$, and $n \times n$ nonnegative matrix $P = \{p_{ij}\}$, where p_{ii} is the profit achieved if the item i is selected and $p_{ij} (i < j)$ is the profit achieved if both items i and j are selected. For $i > j$, $p_{ij} = 0$. When P is a diagonal matrix (i.e., $p_{ij} = 0$ for all $i \neq j$), the QKP is reduced to the KP. Introducing binary variables $x_i \in \{0, 1\}$ provides a mathematical formulation of the QKP to maximize an objective function, which is expressed as

$$\sum_{i=0}^{n-1} \sum_{j=i}^{n-1} p_{ij} x_i x_j, \tag{1}$$

TABLE 1. Types of binary-integer encoding.

Encoding	$f(d)$	D	Notes
One-hot	d	$c+1$	One bit must be 1. All others must be 0.
Binary	2^d	$\lceil \log_2 c \rceil$	
Unary	1	c	

subject to an inequality constraint,

$$\sum_{i=0}^{n-1} w_i x_i \leq c, \tag{2}$$

where c is the knapsack's capacity. The binary variable x_i is 1 if the item i is selected. Otherwise, it is 0.

III. BINARY-INTEGER ENCODING AND QUBO MODEL

A. TYPICAL BINARY-INTEGER ENCODINGS

Binary-integer encoding represents integers by binary variables. Here, we introduce three types of well-known binary-integer encodings: one-hot encoding, binary encoding, and unary encoding [3]. Each encoding describes integer I by a linear function of binary variables,

$$I = \sum_{d=0}^{D-1} f(d) y_d, \tag{3}$$

where we introduce auxiliary bits $y_d \in \{0, 1\}$. The function $f(d)$ is called an encoding function and D is called the bit depth, where D depends on the choice of $f(d)$.

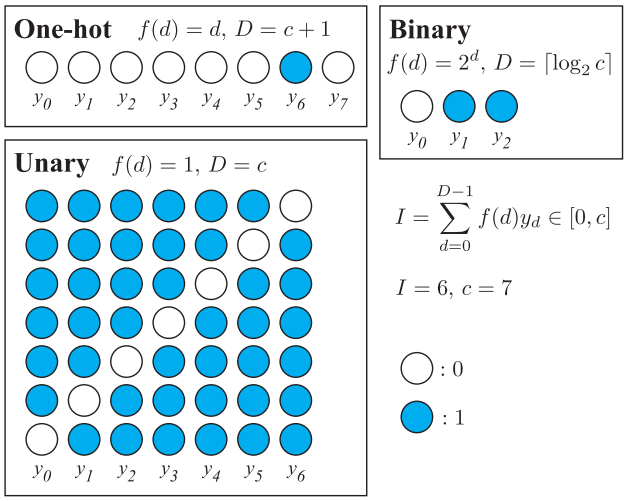

FIGURE 1. A schematic representation of the binary-integer encodings explained in Table 1 when $I = 6$ and $c = 7$.

The QKP must represent each integer $I \in [0, c]$ by binary variables to express the inequality constraint [eq. (2)] in the QUBO model. Table 1 shows the encoding function $f(d)$ and the bit depth D for each encoding. Figure 1 shows a schematic representation of the binary-integer encodings. First, one-hot encoding assumes that one bit takes a value of 1, and the

others are 0. Since the encoding function is given by $f(d) = d$, I is an integer between 0 and $D-1$. Thus, the bit length is given by $D = c+1$. One-hot encoding is one of the traditional encoding methods for representing any integers uniquely [36]. Second, in binary encoding, $f(d) = 2^d$. I is an integer between 0 and $2^D - 1$. Binary encoding is the informationally densest way to represent integers [36]. In most cases, integers from 0 to c are not encodable using only the term $\sum_{d=0}^{D-1} 2^d y_d$. For example, integers from 0 to 7 are encodable by this encoding with $D = 3$, but not integers from 0 to 10. For integers from 0 to 10, at least four auxiliary bits are needed. If $D = 4$, the maximum value of $\sum_{d=0}^{D-1} 2^d y_d$ is 15. Hence it can represent integers larger than 10. This is inappropriate for expressing QKPs because it allows for an overall weight of items that exceeds the knapsack's capacity. To avoid this, we set $D = \lceil \log_2 c \rceil$ and subtract a constant $C = 2^D - 1 - c$ from I so that the maximum value of $I - C$ is c [33]. Finally, in unary encoding, $f(d) = 1$. The number of bits with $y_d = 1$ gives the integer I, and thus, $D = c$. Note that the unary encoding is redundant, and the bit string for a given integer is not generally unique [37]. For example, the integer 2 is represented by three bits as $(y_0, y_1, y_2) = (0, 1, 1), (1, 0, 1)$, and $(1, 1, 0)$.

B. QUBO MODEL FOR QKP

This subsection shows the QUBO model for the QKP. The QUBO model is defined on an undirected graph, which is given by $G = (V, E)$, where V and E are the sets of vertices and edges on G, respectively. The energy function of the QUBO model is referred to as the Hamiltonian and is defined by

$$H_{\mathrm{QUBO}} = \sum_{(a,b)\in E} Q_{ab} z_a z_b, \tag{4}$$

where $z_a \in \{0, 1\}$ is a binary variable on the vertex $a \in V$. The matrix Q_{ab} is called the QUBO matrix.

The QKP provides the Hamiltonians representing the objective function and the inequality constraint. The QUBO model is given by

$$H_{\mathrm{QUBO}} = H_{\mathrm{objective}} + A H_{\mathrm{constraint}}, \tag{5}$$

where $H_{\mathrm{objective}}$ and $H_{\mathrm{constraint}}$ are the Hamiltonians for the objective function and the inequality constraint, respectively. The coefficient $A > 0$ is a hyperparameter, which should be appropriately chosen [see Sec. V-B]. In this study, the only hyperparameter included in the total Hamiltonian is A, regardless of the encoding method.

The Hamiltonian for the objective function is simply given by

$$H_{\mathrm{objective}} = -\sum_{i=0}^{n-1}\sum_{j=i}^{n-1} p_{ij} x_i x_j. \tag{6}$$

TABLE 2. *n* and *c* values of the QKPs and the total number of binary variables *N* for each encoding.

n	c	N		
Items	Capacity	One-hot	Binary	Unary
20	30	51	25	50
50	100	151	57	150
100	200	301	108	300
200	300	501	209	500
400	600	1001	410	1000
800	1200	2001	811	2000

The Hamiltonians for the inequality constraint depend on the binary-integer encoding. They are given by

$$\begin{aligned}
H_{\mathrm{constraint}}^{(\mathrm{one-hot})} &= \left(\sum_{d=0}^{D-1} d y_d - \sum_{i=0}^{n-1} w_i x_i\right)^2 + \left(\sum_{d=0}^{D-1} y_d - 1\right)^2; \\
H_{\mathrm{constraint}}^{(\mathrm{binary})} &= \left(\sum_{d=0}^{D-1} 2^d y_d - (2^D - 1 - c) - \sum_{i=0}^{n-1} w_i x_i\right)^2; \\
H_{\mathrm{constraint}}^{(\mathrm{unary})} &= \left(\sum_{d=0}^{D-1} y_d - \sum_{i=0}^{n-1} w_i x_i\right)^2,
\end{aligned} \tag{7}$$

where D in each encoding is given in Table 1. The constraint Hamiltonian for one-hot encoding follows the one presented in [33]. The above Hamiltonians can be zero if and only if the total weight of the items in the knapsack is less than or equal to the knapsack's capacity (i.e., $\sum_{i=0}^{n-1} w_i x_i \le c$).

IV. METHOD

A. PREPARATION OF QKP INSTANCES

We created QKPs with various n and c values (Table 2). The total number of binary variables N is given by

$$N = n + D. \tag{8}$$

N value for the binary encoding is the smallest, while N values for the one-hot encoding and the unary encoding are almost the same. w_i and $p_{ij} (i \le j)$ are randomly chosen from uniform distributions on $\{1, 2, \ldots, 10\}$ and $\{0, 1, 2, \ldots, 10\}$, respectively. For each n and c, 50 random instances are generated.

Practically, w_i and p_{ij} may be distributed according to a more complex and wider distribution. However, it is unnecessary to solve such problems by Ising machines since the distribution of w_i and p_{ij} can be reduced to the one with a smaller deviation using the following pre-processes. It is not optimal to select items with extremely large weights but extremely small profits. By contrast, items with extremely small weights but extremely large profits should realize optimal solutions. In this way, items with extremely large or small profit per weight can be ignored since it can be trivially determined whether or not such items should be selected. After processing, problems can be broken down into subproblems, which have items with similar weights and profits.

TABLE 3. Digital Annealer parameters.

Number of iterations	10^7
Number of runs	100
Initial temperature	$\alpha_1 N \max_{ab} \lvert Q_{ab} \rvert$
Final temperature	$\alpha_2 \min_{Q_{ab} \neq 0} \lvert Q_{ab} \rvert$
Temperature interval	N

As such, we assume that w_i and p_{ij} are generated uniformly in a limited range of values.

B. SETUP OF AN ISING MACHINE

We used DA [15] as an Ising machine. DA is implemented on CMOS hardware. Its algorithm is based on SA, and it uses massive parallelization. The Ising machine has a maximum of 8192 binary variables on a complete graph. Table 3 shows the tuning parameters of DA. The number of iterations and the number of runs are set to 10^7 and 100, respectively. A mode based on SA is selected. The initial temperature is set sufficiently high for the Hamiltonian. Here, an energy scale of $N \max_{ab} |Q_{ab}|$ is used to characterize the high temperature since it gives an upper bound of the energy change for the flip of a single binary variable. We adjust the initial temperature by a factor α_1, where α_1 takes a value of 1, 10, or 100. On the other hand, the final temperature is set sufficiently low for the Hamiltonian. An energy scale of $\min_{Q_{ab} \neq 0} |Q_{ab}|$ characterizes the low temperature since it gives the minimum excitation energy. The final temperature is adjusted by a factor α_2, where α_2 takes a value of 1, 0.1, or 0.01. The temperature is lowered in every N iteration. The temperature at the n-th update, $T(n)$, is given by

$$T(n) = \exp(-nr)T_{\mathrm{i}}, \tag{9}$$

where T_{i} is the initial temperature and r is the decay rate. The decay rate is set to be consistent with the initial temperature, the final temperature, and the number of iterations.

V. RESULTS

A. A-DEPENDENCES OF FEASIBLE SOLUTION RATES AND AVERAGE ENERGIES

This subsection shows the A-dependences of the probability of obtaining feasible solutions (FSs) and their average energies for each encoding. The FSs denote the solutions that satisfy the inequality constraint in eq. (2). Here, we refer to the probability of obtaining FSs as the FS rate. Namely, the FS rate is defined by the ratio of the number of obtaining the FSs to the number of total runs. Here, the number of total runs is 100 [see Table 3]. The FS rate is obtained by calculating $H_{\mathrm{constraint}}$ since the inequality constraint is satisfied if and only if $H_{\mathrm{constraint}} = 0$. The average energy of the FSs is evaluated with the value of $H_{\mathrm{objective}}$.

Figure 2 shows the A-dependences of the FS rates for each encoding in the QKPs with $(n, c) = (20, 30)$, $(n, c) = (100, 200)$, and $(n, c) = (800, 1200)$. For binary encoding and unary encoding, the FS rates approach 1 as the parameter A increases. The same trends are observed for all other values n and c for both types of encoding. Moreover, the values of A necessary to obtain a high FS rate are smaller for unary encoding than binary encoding, except for $(n, c) = (20, 30)$. For the problem with $(n, c) = (800, 1200)$, the FS rate approaches 1 around $A = 1.6 \times 10^3$ in binary encoding [Fig. 2(h)] and around $A = 0.2 \times 10^3$ in unary encoding [Fig. 2(i)] when α_1 and α_2 are appropriately chosen. This feature becomes more pronounced as the values of n and c increase. On the other hand, in one-hot encoding, few FSs are obtained even when the parameter A is large except for $(n, c) = (20, 30)$. That is, FSs can be obtained with a high frequency in binary encoding and unary encoding at a sufficiently large A, whereas one-hot encoding cannot provide FSs for problems with large n and c.

Figure 3 shows the A-dependences of the average energies of the FSs. The average energies increase with A for binary encoding and unary encoding. The average energies at $\alpha_2 = 1$ are larger than those at $\alpha_2 = 0.01$ and 0.1 for the case of $(n, c) = (800, 1200)$. This indicates that tuning the final temperature is important to achieve a high-quality performance for large-sized problems. For small n and c, one-hot encoding shows qualitatively the same A-dependence of the average energy as binary encoding and unary encoding.

B. PERFORMANCE COMPARISON

The results in subsection V-A indicate that for each encoding A has an optimal value, where the FS rate is high and the average energy of FSs is small. To systematically determine the optimal strength of A, we set a threshold value for the FS rate. In this study, we set the threshold value as 0.95. For a problem instance, we repeatedly solve it while varying the parameters α_1, α_2, and A. Then we find the sets of parameters that give the FS rate above the threshold value, and search the set of parameters for the optimal set to minimize the average energy over the FSs. Below, we discuss the performance difference among the three encodings by comparing the average energies of the FSs at the optimal values of α_1, α_2, and A.

Figure 4(a) compares the average energies of the FSs of all the 50 QKPs between one-hot encoding and unary encoding. The average energies are compared only when $(n, c) = (20, 30)$ since this is the only case where the FS rate in one-hot encoding reaches the threshold. The average energies in unary encoding are smaller than those in one-hot encoding. Namely, unary encoding shows a better performance than one-hot encoding.

Figure 4(b) compares the average energies between binary encoding and unary encoding. For a small-sized problem $[(n, c) = (20, 30)]$, the performances are similar. However, the average energies of unary encoding become smaller than those of binary encoding as the values of n and c increase. Furthermore, the data for $(n, c) = (400, 600)$ and $(800, 1200)$ almost overlap, implying that unary encoding shows the highest performance even for larger-sized problems.

We compare the encoding methods in terms of the computation time. The computation in DA is composed of processing done on CMOS hardware and CPU. The time required for the former is called annealing time, and the time required for

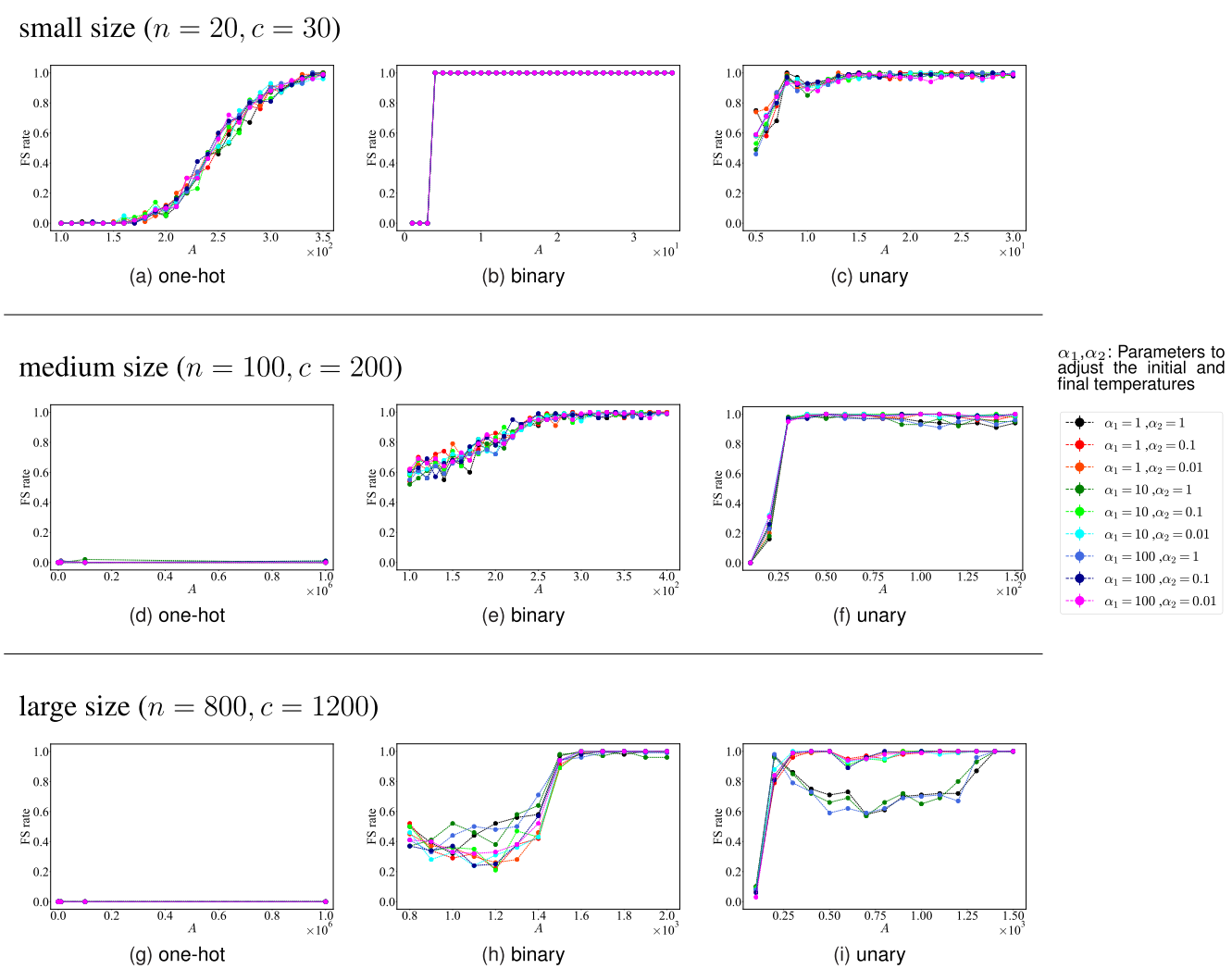

FIGURE 2. *A*-dependences of the FS rates: one-hot encoding [(a),(d),(g)], binary encoding [(b),(e),(h)], and unary encoding [(c),(f),(i)]. Top [(a),(b),(c)], middle [(d),(e),(f)], and bottom row [(g),(h),(i)] show the small-sized problems [$(n, c) = (20, 30)$], medium-sized problems [$(n, c) = (100, 200)$], and large-sized problems [$(n, c) = (800, 1200)$], respectively. For large A, the A-dependences are almost independent of α_1 and α_2.

the latter is called CPU time, and the sum of these is the actual computation time. The annealing time is almost the same for all the encoding methods since the number of iterations is fixed in our setup. On the other hand, the CPU time is longer as the number of bits increases. In the comparison of the CPU times, when $(n, c) = (800, 1200)$, unary encoding takes approximately 50 % longer than binary encoding, and almost the same time as one-hot encoding.

VI. DISCUSSION

Here the results are discussed from two perspectives: unary encoding and the FS rate in one-hot encoding.

Unary encoding outperforms binary encoding, although the total number of binary variables N in unary encoding is larger than that in binary encoding [Tables 1 and 2]. This is counterintuitive since the dimension of the solution space rapidly increases with N. There are two possible reasons for the higher performance of unary encoding. One is that the value of the coefficient A required to obtain a high FS rate in unary encoding is smaller than that in binary encoding. A small value of A improves performance since the average energy increases with A. The other is its redundancy. The number of FSs is larger in unary encoding than in the other encodings because there are multiple ways to represent an integer in unary encoding. This may make it easier for the Ising machine to find the FSs, resulting in a better performance.

One-hot encoding fails to obtain the FSs even for large A [Figs. 2(d) and 2(g)]. Although the number of FSs for one-hot encoding is the same as that for binary encoding, this behavior is observed only in one-hot encoding. To clarify the reason, we conducted additional analysis based on the local-minimum structure of the energy function. Here, a solution in a local minimum is the one whose energy does not decrease under the flip of any single binary variable. From the argument in ref. [33], all the FSs are in local minima for large A in our experiment. In the case of $(n, c) = (800, 1200)$, we have approximately 70 solutions, which are not feasible but are in a local minimum in the 100 runs of DA. (The remaining 30 solutions are neither feasible nor in a local

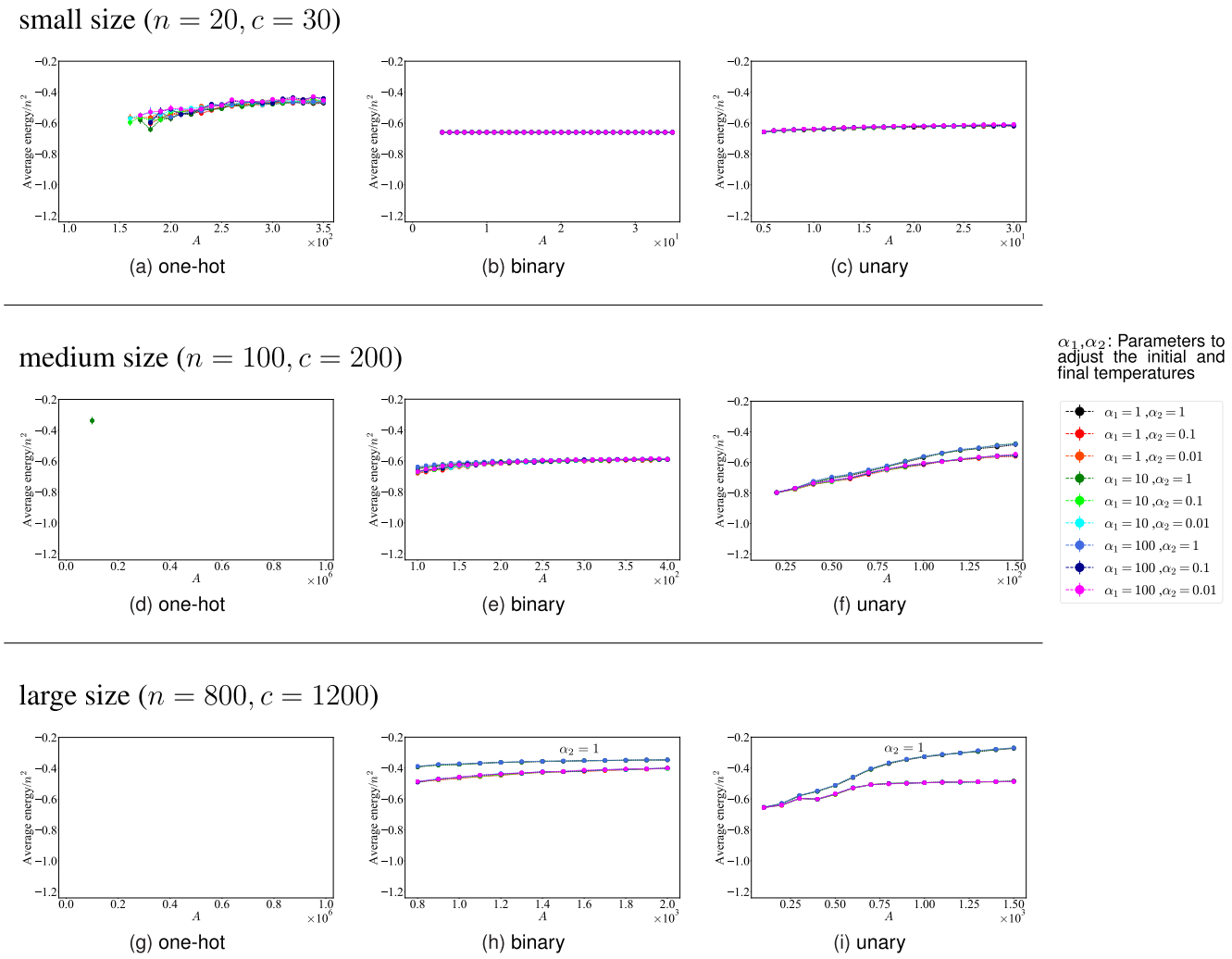

FIGURE 3. *A*-dependences of the average energies of the FSs. Types of binary-integer encodings and the values of *n* and *c* are the same as those in Fig. 2. Fig. (g) lacks data since the FSs cannot be obtained.

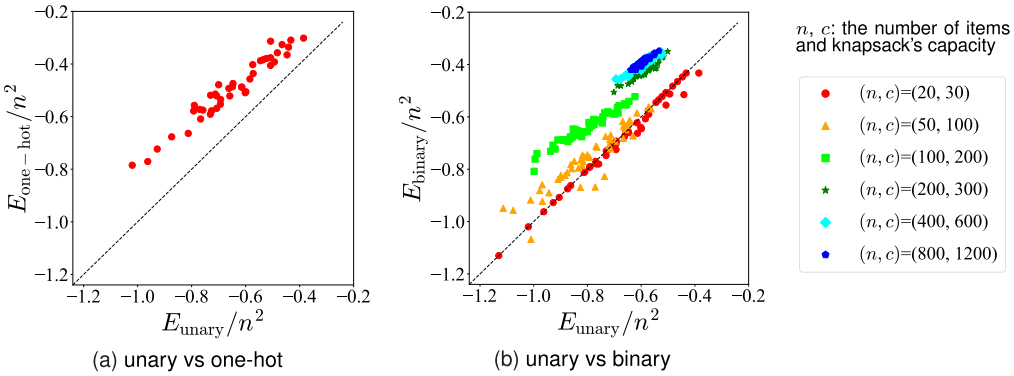

FIGURE 4. Comparisons of the average energies at the optimal values of *A*, α_1, and α_2 (a) between the one-hot encoding and the unary encoding for $(n, c) = (20, 30)$ and (b) between the binary encoding and the unary encoding for various values of *n* and *c*.

minimum.) None of the 70 solutions satisfy the one-hot constraint: $\sum_{d=0}^{D-1} y_d = 1$. This implies that the one-hot constraint creates a number of infeasible local-minimum solutions. Multiple local minima cause the spin-flip dynamics in SA to be slow. Thus, one-hot encoding gives a stronger initial-state dependence than the other encodings, making it difficult to find the FSs by DA. In addition, although the studied model only has one parameter characterizing the constraint term, another parameter can be introduced for one-hot encoding such as $H_{\mathrm{constraint}}^{(\mathrm{one-hot})} = \left(\sum_{d=0}^{D-1} d y_d - \sum_{i=0}^{n-1} w_i x_i\right)^2 + B\left(\sum_{d=0}^{D-1} y_d - 1\right)^2$. In this case, the structure of the local

minima will depend on the additional parameter B. Future analysis of the local-minimum structure while varying B should be considered.

VII. CONCLUSION

The performances of three types of binary-integer encodings (one-hot encoding, binary encoding, and unary encoding) were evaluated. The QKPs were formulated into Ising problems by utilizing each binary-integer encoding. Then the QKPs were solved using an SA-based Ising machine, DA. To compare their performances, the optimal values of the hyperparameters (α_1, α_2, and A) are systematically determined. Unary encoding provided the best results, especially for large-sized problems, whereas one-hot encoding failed to find FSs even for sufficiently large A.

There are several potential reasons for the high performance of unary encoding. In the future, we plan to clarify the important properties of unary encoding to achieve the high performance. In addition, we will investigate whether the performance is independent of the type of combinatorial optimization problem or the choice of Ising machine.

APPENDIX A
OPTIMAL VALUES OF THE HYPERPARAMETERS

The appendix provides the details of the optimal values of the hyperparameters A, α_1, and α_2 for all sizes of the problems we studied. Table 4 shows the average values over the 50 problems for each encoding. Table 4 also shows the mode values of α_1 and α_2. Note that, in one-hot encoding, optimal hyperparameters cannot be determined since a sufficient number of FSs were not obtained except for the case of $(n, c) = (20, 30)$.

TABLE 4. The optimal values of A, α_1, and α_2.

(n, c)	One-hot			Binary			Unary		
	A	α_1	α_2	A	α_1	α_2	A	α_1	α_2
$(20, 30)$	287.2	1	1	11.98	1	1	12.0	1	1
$(50, 100)$	-	-	-	105.4	1	0.1	24.4	1	0.1
$(100, 200)$	-	-	-	245.4	1	0.1	30.8	1	0.1
$(200, 300)$	-	-	-	381.0	1	0.1	67.4	1	0.1
$(400, 600)$	-	-	-	790.0	1	0.1	103.8	1	1
$(800, 1200)$	-	-	-	1514.0	1	0.1	208.0	1	1

ACKNOWLEDGMENT

This article is based on the results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO). The work of Shu Tanaka was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant 19H01553. Kensuke Tamura and Hosho Katsura acknowledge Yutaka Akagi for fruitful discussions.

REFERENCES

[1] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*. Boston, MA, USA: Springer, 1972, pp. 85–103, doi: 10.1007/978-1-4684-2001-2_9.

[2] F. Neukart, G. Compostella, C. Seidel, D. von Dollen, S. Yarkoni, and B. Parney, "Traffic flow optimization using a quantum annealer," *Frontiers ICT*, vol. 4, p. 29, Dec. 2017, doi: 10.3389/fict.2017.00029.

[3] G. Rosenberg, P. Haghnegahdar, P. Goddard, P. Carr, K. Wu, and M. L. de Prado, "Solving the optimal trading trajectory problem using a quantum annealer," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 6, pp. 1053–1060, Sep. 2016, doi: 10.1109/JSTSP.2016.2574703.

[4] H. Neven, V. S. Denchev, G. Rose, and W. G. Macready, "Training a binary classifier with the quantum adiabatic algorithm," 2008, *arXiv:0811.0416*. [Online]. Available: http://arxiv.org/abs/0811.0416

[5] K. Terada, D. Oku, S. Kanamaru, S. Tanaka, M. Hayashi, M. Yamaoka, M. Yanagisawa, and N. Togawa, "An ising model mapping to solve rectangle packing problem," in *Proc. Int. Symp. VLSI Design, Autom. Test (VLSI-DAT)*, Apr. 2018, pp. 1–4.

[6] M. Ohzeki, A. Miki, M. J. Miyama, and M. Terabe, "Control of automated guided vehicles without collision by quantum annealer and digital devices," *Frontiers Comput. Sci.*, vol. 1, p. 9, Nov. 2019. [Online]. Available: https://www.frontiersin.org/article/10.3389/fcomp.2019.00009

[7] N. Yoshimura, M. Tawada, S. Tanaka, J. Arai, S. Yagi, H. Uchiyama, and N. Togawa, "Efficient Ising model mapping for induced subgraph isomorphism problems using Ising machines," in *Proc. IEEE 9th Int. Conf. Consum. Electron. (ICCE-Berlin)*, Sep. 2019, pp. 227–232, doi: 10.1109/ICCE-Berlin47944.2019.8966218.

[8] N. Nishimura, K. Tanahashi, K. Suganuma, M. J. Miyama, and M. Ohzeki, "Item listing optimization for e-commerce websites based on diversity," *Frontiers Comput. Sci.*, vol. 1, p. 2, Jul. 2019. [Online]. Available: https://www.frontiersin.org/article/10.3389/fcomp.2019.00002

[9] D. Venturelli and A. Kondratyev, "Reverse quantum annealing approach to portfolio optimization problems," *Quantum Mach. Intell.*, vol. 1, nos. 1–2, pp. 17–30, May 2019.

[10] K. Kitai, J. Guo, S. Ju, S. Tanaka, K. Tsuda, J. Shiomi, and R. Tamura, "Designing metamaterials with quantum annealing and factorization machines," *Phys. Rev. Res.*, vol. 2, no. 1, Mar. 2020, Art. no. 013319, doi: 10.1103/PhysRevResearch.2.013319.

[11] S. Kanamaru, K. Kawamura, S. Tanaka, Y. Tomita, and N. Togawa, "Solving constrained slot placement problems using an Ising machine and its evaluations," *IEICE Trans. Inf. Syst.*, vol. E104.D, no. 2, pp. 226–236, Feb. 2021, doi: 10.1587/transinf.2019EDP7254.

[12] D. Inoue, A. Okada, T. Matsumori, K. Aihara, and H. Yoshida, "Traffic signal optimization on a square lattice with quantum annealing," *Sci. Rep.*, vol. 11, no. 1, pp. 1–12, Dec. 2021.

[13] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, "A 20k-spin Ising chip to solve combinatorial optimization problems with CMOS annealing," *IEEE J. Solid-State Circuits*, vol. 51, no. 1, pp. 303–309, Jan. 2016, doi: 10.1109/JSSC.2015.2498601.

[14] T. Okuyama, M. Hayashi, and M. Yamaoka, "An ising computer based on simulated quantum annealing by path integral Monte Carlo method," in *Proc. IEEE Int. Conf. Rebooting Comput. (ICRC)*, Nov. 2017, pp. 1–6, doi: 10.1109/ICRC.2017.8123652.

[15] 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 Phys.*, vol. 7, p. 48, Apr. 2019, doi: 10.3389/fphy.2019.00048.

[16] T. Okuyama, T. Sonobe, K.-I. Kawarabayashi, and M. Yamaoka, "Binary optimization by momentum annealing," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 100, no. 1, Jul. 2019, Art. no. 012111, doi: 10.1103/PhysRevE.100.012111.

[17] C. Yoshimura, M. Hayashi, T. Okuyama, and M. Yamaoka, "Implementation and evaluation of FPGA-based annealing processor for Ising model by use of resource sharing," *Int. J. Netw. Comput.*, vol. 7, no. 2, pp. 154–172, 2017, doi: 10.15803/ijnc.7.2_154.

[18] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, and E. M. Chapple, "Quantum annealing with manufactured spins," *Nature*, vol. 473, no. 7346, pp. 194–198, May 2011, doi: 10.1038/nature10012.

[19] M. Maezawa, K. Imafuku, M. Hidaka, H. Koike, and S. Kawabata, "Design of quantum annealing machine for prime factoring," in *Proc. 16th Int. Superconductive Electron. Conf. (ISEC)*, Jun. 2017, pp. 1–3, doi: 10.1109/ISEC.2017.8314195.

[20] M. Maezawa, G. Fujii, M. Hidaka, K. Imafuku, K. Kikuchi, H. Koike, K. Makise, S. Nagasawa, H. Nakagawa, M. Ukibe, and S. Kawabata, "Toward practical-scale quantum annealing machine for prime factoring," *J. Phys. Soc. Jpn.*, vol. 88, no. 6, Jun. 2019, Art. no. 061012, doi: 10.7566/JPSJ.88.061012.

[21] Z. Wang, A. Marandi, K. Wen, R. L. Byer, and Y. Yamamoto, "Coherent Ising machine based on degenerate optical parametric oscillators," *Phys. Rev. A, Gen. Phys.*, vol. 88, no. 6, Dec. 2013, Art. no. 063853, doi: 10.1103/PhysRevA.88.063853.

[22] A. Marandi, Z. Wang, K. Takata, R. L. Byer, and Y. Yamamoto, "Network of time-multiplexed optical parametric oscillators as a coherent Ising machine," *Nature Photon.*, vol. 8, no. 12, pp. 937–942, Oct. 2014, doi: 10.1038/nphoton.2014.249.

[23] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. Mcmahon, T. Umeki, K. Enbutsu, O. Tadanaga, H. Takenouchi, K. Aihara, K.-I. Kawarabayashi, K. Inoue, S. Utsunomiya, and H. Takesue, "A coherent Ising machine for 2000-node optimization problems," *Science*, vol. 354, no. 6312, pp. 603–606, Nov. 2016, doi: 10.1126/science.aah4243.

[24] P. L. Mcmahon, A. Marandi, Y. Haribara, R. Hamerly, C. Langrock, S. Tamate, T. Inagaki, H. Takesue, S. Utsunomiya, K. Aihara, R. L. Byer, M. M. Fejer, H. Mabuchi, and Y. Yamamoto, "A fully programmable 100-spin coherent Ising machine with all-to-all connections," *Science*, vol. 354, no. 6312, pp. 614–617, Nov. 2016, doi: 10.1126/science.aah5178.

[25] H. Goto, K. Tatsumura, and A. R. Dixon, "Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems," *Sci. Adv.*, vol. 5, no. 4, Apr. 2019, Art. no. eaav2372, doi: 10.1126/sciadv.aav2372.

[26] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 58, no. 5, p. 5355, 1998, doi: 10.1103/PhysRevE.58.5355.

[27] K. Hukushima and K. Nemoto, "Exchange Monte Carlo method and application to spin glass simulations," *J. Phys. Soc. Jpn.*, vol. 65, no. 6, pp. 1604–1608, Jun. 1996, doi: 10.1143/JPSJ.65.1604.

[28] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983, doi: 10.1126/science.220.4598.671.

[29] T. Shirai, S. Tanaka, and N. Togawa, "Guiding principle for minor-embedding in simulated-annealing-based Ising machines," *IEEE Access*, vol. 8, pp. 210490–210502, 2020, doi: 10.1109/ACCESS.2020.3040017.

[30] D. Oku, M. Tawada, S. Tanaka, and N. Togawa, "How to reduce the bit-width of an Ising model by adding auxiliary spins," *IEEE Trans. Comput.*, early access, Dec. 15, 2020, doi: 10.1109/TC.2020.3045112.

[31] E. Boros and P. L. Hammer, "Pseudo-Boolean optimization," *Discrete Appl. Math.*, vol. 123, nos. 1–3, pp. 155–225, Nov. 2002, doi: 10.1016/S0166-218X(01)00341-9.

[32] H. Nishimori and G. Ortiz, *Elements of Phase Transitions and Critical Phenomena*. Oxford, U.K.: Oxford Univ. Press, 2010, doi: 10.1093/acprof:oso/9780199577224.001.0001.

[33] A. Lucas, "Ising formulations of many NP problems," *Frontiers Phys.*, vol. 2, p. 5, Feb. 2014, doi: 10.3389/fphy.2014.00005.

[34] S. Tanaka, R. Tamura, and B. K. Chakrabarti, *Quantum Spin Glasses, Annealing and Computation*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[35] M. W. Coffey, "Adiabatic quantum computing solution of the knapsack problem," 2017, *arXiv:1701.05584*. [Online]. Available: http://arxiv.org/abs/1701.05584

[36] N. Chancellor, "Domain wall encoding of discrete variables for quantum annealing and QAOA," *Quantum Sci. Technol.*, vol. 4, no. 4, Aug. 2019, Art. no. 045004.

[37] S. Karimi and P. Ronagh, "Practical integer-to-binary mapping for quantum annealers," *Quantum Inf. Process.*, vol. 18, no. 4, p. 94, Feb. 2019, doi: 10.1007/s11128-019-2213-x.

[38] T. Vyskočil, S. Pakin, and H. N. Djidjev, "Embedding inequality constraints for quantum annealing optimization," in *Quantum Technology and Optimization Problems*. Cham, Switzerland: Springer, 2019, pp. 11–22, doi: 10.1007/978-3-030-14082-3_2.

[39] K. Tanahashi, S. Takayanagi, T. Motohashi, and S. Tanaka, "Application of Ising machines and a software development for Ising machines," *J. Phys. Soc. Jpn.*, vol. 88, no. 6, Jun. 2019, Art. no. 061010, doi: 10.7566/JPSJ.88.061010.

[40] F. Leon and P. Caşcaval, "01IP and QUBO: Optimization methods for redundancy allocation in complex systems," in *Proc. 23rd Int. Conf. Syst. Theory, Control Comput. (ICSTCC)*, Oct. 2019, pp. 877–882, doi: 10.1109/ICSTCC.2019.8885826.

[41] L. Pusey-Nazzaro and P. Date, "Adiabatic quantum optimization fails to solve the knapsack problem," 2020, *arXiv:2008.07456*. [Online]. Available: http://arxiv.org/abs/2008.07456

[42] M. Zaman, K. Tanahashi, and S. Tanaka, "PyQUBO: Python library for mapping combinatorial optimization problems to QUBO form," 2021, *arXiv:2103.01708*. [Online]. Available: https://arxiv.org/abs/2103.01708

[43] G. Gallo, P. L. Hammer, and B. Simeone, *Quadratic Knapsack Problems*. Berlin, Germany: Springer, 1980, pp. 132–149, doi: 10.1007/BFb0120892.

[44] K. Yonaga, M. J. Miyama, and M. Ohzeki, "Solving inequality-constrained binary optimization problems on quantum annealer," 2020, *arXiv:2012.06119*. [Online]. Available: http://arxiv.org/abs/2012.06119

[45] H. M. Salkin and C. A. De Kluyver, "The knapsack problem: A survey," *Nav. Res. Logistics Quart.*, vol. 22, no. 1, pp. 127–144, Mar. 1975, doi: 10.1002/nav.3800220110.

[46] V. Boyer, D. E. Baz, and M. Elkihel, "Solving knapsack problems on GPU," *Comput. Oper. Res.*, vol. 39, no. 1, pp. 42–47, Jan. 2012.

[47] J. Jaros, "Multi-GPU island-based genetic algorithm for solving the knapsack problem," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–8.

KENSUKE TAMURA received the B.Sc. degree from Waseda University, in 2018, and the M.Sc. degree from The University of Tokyo, in 2020, where he is currently pursuing the Ph.D. degree in physics with the Department of Physics, Graduate School of Science. His research interests include quantum many-body systems and mathematical physics. He is a member of the JPS.

TATSUHIKO SHIRAI received the B.Sc., M.Sc., and Dr.Sc. degrees from The University of Tokyo, in 2011, 2013, and 2016, respectively. He is currently an Assistant Professor with the Department of Computer Science and Communications Engineering, Waseda University. His research interests include quantum dynamics and statistical mechanics. He is a member of the JPS.

HOSHO KATSURA received the B.Sc., M.Eng., and Dr.Eng. degrees from The University of Tokyo, in 2004, 2006, and 2008, respectively. He is currently an Associate Professor with the Department of Physics, The University of Tokyo. His research interests include strongly correlated systems, topological phases of matter, and exactly solvable models. He is a member of the JPS.

SHU TANAKA received the B.Sc. degree from the Tokyo Institute of Technology, in 2003, and the M.Sc. and Dr.Sc. degrees from The University of Tokyo, in 2005 and 2008, respectively. He is currently an Associate Professor with the Department of Applied Physics and Physico-Informatics, Keio University. His research interests include quantum annealing, Ising machine, statistical mechanics, and materials science. He is a member of the JPS.

NOZOMU TOGAWA (Member, IEEE) received the B.Eng., M.Eng., and Dr.Eng. degrees from Waseda University, in 1992, 1994, and 1997, respectively, all in electrical engineering. He is currently a Professor with the Department of Computer Science and Communications Engineering, Waseda University. His research interests include VLSI design, graph theory, and computational geometry. He is a member of the IEICE and the IPSJ.