Spin-Variable Reduction Method for Handling Linear Equality Constraints in Ising Machines

Tatsuhiko Shirai, *Member, IEEE*, and Nozomu Togawa, *Member, IEEE*

***Abstract*—We propose a spin-variable reduction method for Ising machines to handle linear equality constraints in a combinatorial optimization problem. Ising machines including quantum-annealing machines can effectively solve combinatorial optimization problems. They are designed to find the lowest-energy solution of a quadratic unconstrained binary optimization (QUBO), which is mapped from the combinatorial optimization problem. The proposed method reduces the number of binary variables to formulate the QUBO compared to the conventional penalty method. We demonstrate a sufficient condition to obtain the optimum of the combinatorial optimization problem in the spin-variable reduction method and its general applicability. We apply it to typical combinatorial optimization problems, such as the graph *k*-partitioning problem and the quadratic assignment problem. Experiments using simulated-annealing and quantum-annealing based Ising machines demonstrate that the spin-variable reduction method outperforms the penalty method. The proposed method extends the application of Ising machines to larger-size combinatorial optimization problems with linear equality constraints.**

***Index Terms*—Combinatorial optimization problem, Ising machine, Ising model, metaheuristics, quantum annealing, simulated annealing, variable reduction.**

I. Introduction

Combinatorial optimization problems find the optimal combination of decision variables to minimize or maximize an objective function under a set of given constraints [1]. Solving a combinatorial optimization problem with many decision variables is challenging because the number of solution candidates increases exponentially as the number of decision variables increases. Although typical examples found in textbooks are the satisfiability problem, quadratic assignment problem (QAP), and graph k-partitioning problem (GkPP), combinatorial optimization problems are ubiquitous in daily life. Examples include the logistics optimization, traffic route optimization, and drug discovery.

Ising machines are specialized computers for combinatorial optimization problems. Most Ising machine hardware operates on the basis of simulated annealing (SA) [2], [3], [4], [5], [6], [7] and quantum annealing (QA) [8], [9] (see the review of Ising machines in [10]). The quadratic unconstrained binary optimization (QUBO) is used to solve a combinatorial optimization problem in Ising machines [11], [12], [13], [14], [15], [16]. The QUBO is described by binary variables called spins, which take values of 0 or 1.[1] Then, the solution space of a combinatorial optimization problem is transformed into the spin configuration space $\{0,1\}^{\otimes N}$, where N is the number of spins. The QUBO problem finds the spin configuration to minimize the energy function defined by the QUBO. Ising machines search for the lowest-energy state of a QUBO, which is called the ground state. Efficient methods have been proposed for improving the performance of Ising machines [17], [18], [19], [20]. The optimal solution of a combinatorial optimization problem is obtained from the ground state. This paper develops a method for Ising machines.

The energy function of a QUBO is generally given as the sum of the terms for an objective function and the constraints. Feasible solutions (FSs) satisfy the constraints. The conventional method, which is called the penalty method, gives the constraint term by increasing the energy of infeasible solutions [12]. The spin configuration space is separated into many FS subspaces with lower energies by the infeasible solution space with higher energies. Fig. 1(a) applies the penalty method to a combinatorial optimization problem with two binary variables and a single linear equality constraint. The QUBO is expressed by $H_{\mathrm{penalty}} = x_1 - x_2 + 5(x_1 + x_2 - 1)^2$, where the first term $x_1 - x_2$ is the objective function and the second term $(x_1 + x_2 - 1)^2$ is the constraint term. The penalty 5 is given to the constraint term to increase the energy of infeasible solutions. Then, the two FSs $(x_1, x_2) = (0,1)$ and $(1,0)$ are separated by infeasible solutions with large energies. The transition probability between the FSs is low when the thermal or quantum fluctuation is small. In this situation, searching for the optimum using an Ising machine is difficult. SA-based or QA-based Ising machine studies have reported that the penalty method falls into a local minimum solution and fails to reach the optimum of the QUBO (e.g., Refs. [21], [22]). Therefore, developing an efficient constraint-handling method is important.

In this study, we propose a new constraint-handling method called the spin-variable reduction (SVR) method. The proposed method, which is based on the idea of variable reduction, expresses a spin variable using other spin variables in terms

Manuscript received 28 July 2022; revised 19 December 2022; accepted 1 January 2023. Date of publication 24 January 2023; date of current version 11 July 2023. Recommended for acceptance by Q. Li. *(Corresponding author: Tatsuhiko Shirai.)*

The authors are with the Department of Computer Science and Communications Engineering, Waseda University, Tokyo 169-8555, Japan (e-mail: tatsuhiko.shirai@aoni.waseda.jp; ntogawa@waseda.jp).

This article has supplementary downloadable material available at https://doi.org/10.1109/TC.2023.3239539, provided by the authors.

Digital Object Identifier 10.1109/TC.2023.3239539

[1] We herein adopt this definition. However spins are often denoted as binary variables with values of -1 or 1. Both definitions are equivalent since a spin $\sigma \in \{-1,1\}$ is expressed by a spin $x \in \{0,1\}$ as $\sigma = 2x - 1$.

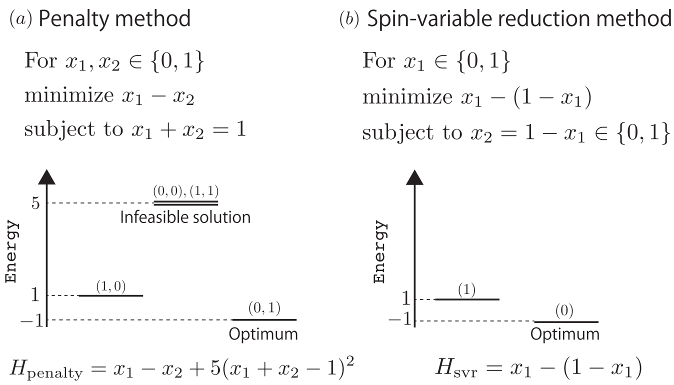

Fig. 1. Examples of the penalty method and the spin-variable reduction method in a simple combinatorial optimization problem with two binary variables and a single linear equality constraint. Both methods formulate the combinatorial optimization problem into QUBOs. (a) The penalty method gives a QUBO H_{penalty} as a function of (x_1, x_2). Two infeasible solutions have a large energy. (b) The spin-variable reduction method gives a QUBO H_{svr} as a function of x_1. H_{svr} is independent of x_2 and x_2 is obtained by x_1 using the variable relationship in the linear equality constraint (i.e., $x_2 = 1 - x_1$).

of the variable relationships in linear equality constraints. The SVR method reduces the number of spin variables compared to the penalty method and decreases the number of the infeasible solutions. Fig. 1(b) applies the SVR method to the combinatorial optimization problem given in (a). The QUBO is expressed by $H_{\text{svr}} = x_1 - (1 - x_1) = 2x_1 - 1$. The SVR method has one less spin than the penalty method, and thus H_{svr} depends on a single spin x_1. The reduced spin value x_2 is obtained by x_1 using the variable relationship in the linear equality constraint (i.e., $x_2 = 1 - x_1$). The SVR method induces a transition between the FSs $(x_1) = (0)$ and (1) without passing large-energy infeasible solutions. In this way, reducing the number of the infeasible solutions leads to an efficient search for the optimum.

The main contributions of this paper are as follows:

- We propose a new constraint-handling method called the SVR method for QUBO problems. The SVR method reduces the number of spins compared to the penalty method.
- We give a sufficient condition to obtain the optimum of the combinatorial optimization problem in the SVR method. This condition covers a wide range of constraints, which typically appear in QUBO problems.
- We apply the SVR method to two NP-hard combinatorial optimization problems: GkPP and QAP. Then we experimentally solve QUBO problems using an SA-based Ising machine and a QA-based Ising machine. The SVR method outperforms the penalty method.

The rest of this paper is organized as follows. Section II defines the constrained combinatorial optimization problem and reviews the penalty method. Section III proposes the SVR method. Section IV formulates the SVR method in GkPP and QAP. Section V details the experimental results using SA-based and QA-based Ising machines. Section VI discusses the experimental results. Section VII summarizes this paper. Appendices, which can be found on the Computer Society Digital Library at available online, give supplements on the application of the SVR method to d-dimensional systems, the domain-wall method for 2-dimensional systems, the results for extended version of GkPP, and the ideal QA simulation.

II. Preliminaries and Related Works

A. Definitions

This paper considers the following combinatorial optimization problem:

Problem 1: For $x_i \in \{0,1\}$ where $i \in \{1, 2, \ldots\} =: V$, find arg $\min_{\{x_i\}} Q(\{x_i\}_{i\in V})$ under the linear equality constraints such that for $k \in M$

$$C_k\left(\{x_i\}_{i\in V}\right) = \tilde{a}_{0,k} + \sum_{i\in V} \tilde{a}_{i,k} x_i = 0, \tag{1}$$

where $\tilde{a}_{i,k} \in \mathbb{R}$ for $i \in V \cup \{0\}$.

Here, $Q(\{x_i\}_{i\in V})$ and $C_k(\{x_i\}_{i\in V}) = 0$ denote the objective function and the k-th constraint, respectively. M is the set of the linear equality constraints.

Here, the equivalence of combinatorial optimization problems is defined as:

Definition 1: Given the set of the optimal solutions of problem a and problem b as $V_a^* \subseteq V$ and $V_b^* \subseteq V$, respectively, the two problems are equivalent if $V_a^* = V_b^*$.

B. QUBO and Penalty Method

QUBO [23] is a common input format for Ising machines and is defined on an undirected graph $G_{\text{qubo}} = (V_{\text{qubo}}, E_{\text{qubo}})$, where V_{qubo} and E_{qubo} are the vertex set and edge set, respectively. They are given as

$$H\left(\{x_i\}_{i\in V_{\text{qubo}}}\right) = \sum_{(i,j)\in E_{\text{qubo}}} Q_{i,j} x_i x_j + \sum_{i\in V_{\text{qubo}}} Q_{i,i} x_i + Q_0, \tag{2}$$

where $x_i \in \{0,1\}$ is a binary variable called the spin and $Q_{i,j}, Q_{i,i}, Q_0 \in \mathbb{R}$. The QUBO problem finds the ground state of $H(\{x_i\}_{i\in V_{\text{qubo}}})$.

The penalty method formulates Problem 1 as the QUBO problem in (2) [12], [14].

Problem 2: For $x_i \in \{0,1\}$ where $i \in V$, for a sufficiently large $\lambda \in \mathbb{R}$ find arg $\min_{\{x_i\}_{i\in V}} H_{\text{penalty}}(\{x_i\}_{i\in V})$ where

$$H_{\text{penalty}}\left(\{x_i\}_{i\in V}\right) = Q\left(\{x_i\}_{i\in V}\right) + \lambda \sum_{k\in M} \left[C_k\left(\{x_i\}_{i\in V}\right)\right]^2. \tag{3}$$

For a sufficiently large λ, $C_k(\{x_i\}_{i\in V})$ is zero for $k \in M$. Thus, the following theorem is obtained.

Theorem 1: Problem 1 is equivalent to Problem 2 [14].

$H_{\text{penalty}}(\{x_i\}_{i\in V})$ takes a QUBO form if $Q(\{x_i\}_{i\in V})$ is expressed in the QUBO and $C_k(\{x_i\}_{i\in V})$ is a linear equality constraint in (1). It should be noted that even if $Q(\{x_i\}_{i\in V})$ contains higher order terms (e.g., $x_1x_2x_3$), introducing auxiliary variables always reduces these to second order terms [24]. To the best of our knowledge, only the penalty method is known to generally formulate the constrained combinatorial optimization problems in Problem 1 by the QUBO form.

TABLE I
KEY NOTATIONS USED IN THE PAPER

Symbol	Description
V	Set of spin variables
V_1	Set of independent spin variables
V_2	Set of dependent spin variables
M	Set of linear equality constraints
M_2	Set of eliminated linear equality constraints
M_1	$M \setminus M_2$
x_i	Spin variable
H, H'	QUBO
Q, Q'	Objective function
C_k, C'_k	Linear equality constraint
R_i	Description of dependent spin x_i as a function of independent spins

III. SPIN-VARIABLE REDUCTION METHOD

Here, a new constraint-handling method called the SVR method is formulated (Section III-A). The SVR method formulates Problem 1 as a QUBO problem. The QUBOs are explicitly given for combinatorial optimization problems with typical constraints (Section III-B). Table I lists the notations used in this section.

A. Spin-Variable Reduction Method

Our proposed method is based on the idea of variable reduction. First, we introduce subsets $V_1, V_2 \subseteq V$ and $M_1, M_2 \subseteq M$, where V and M were defined in Problem 1 (i.e., V is a set of the spin variables and M is a set of the linear equality constraints).

Independent Spins:

$i \in V_1$ when a spin variable is used to represent other spin variables in terms of the variable relationships in linear equality constraints. The spin variables in V_1 are called independent spins.

Dependent spins

$i \in V_2$ when a spin variable is expressed and calculated by independent spins. V_2 is given as $V_2 = V \setminus V_1$. The spin variables in V_2 are called dependent spins.

Eliminated equality constraints

$k \in M_2$ when the linear equality constraints $C_k(\{x_i\}_{i\in V})$ are eliminated along with the variable reduction. This is called the eliminated equality constraint.

The complement set of M_2 is defined as $M_1 := M \setminus M_2$. A one-to-one correspondence exists between V_2 and M_2 (i.e., $V_2 \simeq M_2$) since the one equality constraint is used to eliminate one spin.

Dependent spins can be expressed in terms of independent spins. For $k \in M_2$, the equality constraint gives

$$\sum_{i\in V_2} \tilde{a}_{i,k} x_i = -\tilde{a}_{0,k} - \sum_{i\in V_1} \tilde{a}_{i,k} x_i, \tag{4}$$

where $\{\tilde{a}_{i,k}\}$ for $i \in V_2$ and $k \in M_2$ can be regarded as a matrix with dimension $|V_2|$. When the constraints in $k \in M_2$ are linearly independent, the matrix is given by a regular square matrix. Here, the inverse matrix is introduced with a matrix element $\tilde{a}^{-1}_{k,i}$ for $k \in M_2$ and $i \in V_2$ satisfying $\sum_{k\in M_2} \tilde{a}_{i,k}\tilde{a}^{-1}_{k,j} = \delta_{i,j}$ for $i, j \in V_2$ and $\sum_{i\in V_2} \tilde{a}_{i,\ell}\tilde{a}^{-1}_{k,i} = \delta_{k,\ell}$ for $k, \ell \in M_2$, where $\delta_{i,j}$ is the Kronecker's delta. Then, (4) gives for $i \in V_2$

$$\begin{aligned} x_i &= -\sum_{k\in M_2} \left(\tilde{a}_{0,k} + \sum_{j\in V_1} \tilde{a}_{j,k} x_j \right) \tilde{a}^{-1}_{k,i} \\ &= a_{0,i} + \sum_{j\in V_1} a_{j,i} x_j =: R_i\left(\{x_j\}_{j\in V_1}\right), \end{aligned} \tag{5}$$

where $a_{j,i} = -\sum_{k\in M_2} \tilde{a}_{j,k}\tilde{a}^{-1}_{k,i}$ for $j \in V_1 \cup \{0\}$.

Next, a new objective function and constraints are redefined as functions of independent spins. They are obtained by replacing the dependent spins by $R_i(\{x_j\}_{j\in V_1})$ in $Q(\{x_i\}_{i\in V})$ and $C_k(\{x_i\}_{i\in V})$. Namely,

$$Q'\left(\{x_i\}_{i\in V_1}\right) := Q\left(\{x_i\}_{i\in V_1}, \{R_i\left(\{x_j\}_{j\in V_1}\right)\}_{i\in V_2}\right), \tag{6}$$

and for $k \in M_1$

$$C'_k\left(\{x_i\}_{i\in V_1}\right) := C_k\left(\{x_i\}_{i\in V_1}, \{R_i\left(\{x_j\}_{j\in V_1}\right)\}_{i\in V_2}\right). \tag{7}$$

It should be noted that $C'_k(\{x_i\}_{i\in V_1}) = 0$ maintains a linear equality constraint. Then, the combinatorial optimization problem can be reformulated as

Problem 3: For $x_i \in \{0, 1\}$ where $i \in V$, find $\arg\min_{\{x_i\}_{i\in V_1}} Q'(\{x_i\}_{i\in V_1})$ under the linear equality constraints such that for $k \in M_1$

$$C'_k\left(\{x_i\}_{i\in V_1}\right) = 0 \tag{8}$$

and the constraints of dependent spins that for $i \in V_2$

$$x_i = R_i\left(\{x_j\}_{j\in V_1}\right) \in \{0, 1\}. \tag{9}$$

Equation (9) gives the constraint for independent spins since $R_i(\{x_j\}_{j\in V_1})$ can take values other than 0 or 1.

Lemma 2: Problems 1 and 3 are equivalent.

Proof of Lemma 2 The equivalence of Problems 1 and 3 are proven in terms of Definition 1. We consider sets of FSs for Problems 1 and 3, which are denoted by $V^{(1)\mathrm{f}}$ and $V^{(3)\mathrm{f}}$, respectively. $\{x_i\}_{i\in V} \in V^{(1)\mathrm{f}}$ if $C_k(\{x_i\}_{i\in V}) = 0$ for all $k \in M$ and $\{x_i\}_{i\in V} \in V^{(3)\mathrm{f}}$ if $C'_k(\{x_i\}_{i\in V_1}) = 0$ for all $k \in M_1$ and $x_i = R_i(\{x_j\}_{j\in V_1})$ for $i \in V_2$. First, we show $V^{(1)\mathrm{f}} \subseteq V^{(3)\mathrm{f}}$. Suppose that $\{x_i\}_{i\in V} \in V^{(1)\mathrm{f}}$. Then, $C_k(\{x_i\}_{i\in V}) = 0$ for $k \in M_2$ gives $x_i = R_i(\{x_j\}_{j\in V_1})$ for $i \in V_2$. Next we find for $k \in M_1$

$$\begin{aligned} C'_k\left(\{x_i\}_{i\in V_1}\right) &= C_k\left(\{x_i\}_{i\in V_1}, \{R_i\left(\{x_j\}_{j\in V_1}\right)\}_{i\in V_2}\right) \\ &= C_k\left(\{x_i\}_{i\in V}\right) = 0. \end{aligned} \tag{10}$$

It indicates $\{x_i\}_{i\in V} \in V^{(3)\mathrm{f}}$, and consequently $V^{(1)\mathrm{f}} \subseteq V^{(3)\mathrm{f}}$. We also show $V^{(3)\mathrm{f}} \subseteq V^{(1)\mathrm{f}}$. Suppose that $\{x_i\}_{i\in V} \in V^{(3)\mathrm{f}}$. Then, $x_i = R_i(\{x_j\}_{j\in V_1})$ for $i \in V_2$ gives for $k \in M_1$

$$\begin{aligned} 0 &= C'_k\left(\{x_i\}_{i\in V_1}\right) \\ &= C_k\left(\{x_i\}_{i\in V_1}, \{R_i\left(\{x_j\}_{j\in V_1}\right)\}_{i\in V_2}\right) \\ &= C_k\left(\{x_i\}_{i\in V}\right), \end{aligned} \tag{11}$$

and

$$x_i = R_i\left(\{x_j\}_{j\in V_1}\right)$$

$$= a_{0,i} + \sum_{j\in V_1} a_{j,i} x_j$$

$$= -\sum_{\ell\in M_2} \left(\tilde{a}_{0,\ell} + \sum_{j\in V_1} \tilde{a}_{j,\ell} x_j \right) \tilde{a}_{\ell,i}^{-1}. \tag{12}$$

For $k \in M_2$

$$\sum_{i\in V_2} \tilde{a}_{i,k} x_i = -\sum_{i\in V_2} \tilde{a}_{i,k} \sum_{\ell\in M_2} \left(\tilde{a}_{0,\ell} + \sum_{j\in V_1} \tilde{a}_{j,\ell} x_j \right) \tilde{a}_{\ell,i}^{-1}$$

$$= -\sum_{\ell\in M_2} \left(\tilde{a}_{0,\ell} + \sum_{j\in V_1} \tilde{a}_{j,\ell} x_j \right) \delta_{k,\ell}$$

$$= -\tilde{a}_{0,k} - \sum_{i\in V_1} \tilde{a}_{i,k} x_i,$$

$$\Leftrightarrow \tilde{a}_{0,k} + \sum_{i\in V} \tilde{a}_{i,k} x_i = 0,$$

$$\Leftrightarrow C_k\left(\{x_i\}_{i\in V}\right) = 0. \tag{13}$$

Hence, $\{x_i\}_{i\in V} \in V^{(1)\mathrm{f}}$. Thus, $V^{(3)\mathrm{f}} \subseteq V^{(1)\mathrm{f}}$. Therefore, $V^{(1)\mathrm{f}} = V^{(3)\mathrm{f}} =: V^{\mathrm{f}}$.

We consider a FS $\{x_i\}_{i\in V} \in V^{\mathrm{f}}$. Then,

$$Q\left(\{x_i\}_{i\in V}\right) = Q\left(\{x_i\}_{i\in V_1}, \{R_i\left(\{x_j\}_{j\in V_1}\right)\}_{i\in V_2}\right)$$
$$= Q'\left(\{x_i\}_{i\in V_1}\right). \tag{14}$$

Let the sets of optimal solutions of Problems 1 and 3 be $V^{(1)*}$ and $V^{(3)*}$, respectively. Then, $V^{(1)\mathrm{f}} = V^{(3)\mathrm{f}}$ and $Q(\{x_i\}_{i\in V}) = Q'(\{x_i\}_{i\in V_1})$ for each FS indicate $V^{(1)*} = V^{(3)*}$. Thus, Problems 1 and 3 are equivalent.

Problem 3 is reformulated. We use the fact that constraints $R_i(\{x_j\}_{j\in V_1}) \in \{0,1\}$ in Problem 3 are expressed as $\arg\min_{\{x_j\}_{j\in V_1}} R_i(\{x_j\}_{j\in V_1})[R_i(\{x_j\}_{j\in V_1}) - 1]$ when $R_i(\{x_j\}_{j\in V_1}) \in \mathbb{Z}$. It gives the following Problem 4 and Lemma 3.

Problem 4: For $x_i \in \{0,1\}$ where $i \in V_1$, for sufficiently large $\lambda \in \mathbb{R}$ and $\tau \in \mathbb{R}$, find $\arg\min_{\{x_i\}_{i\in V_1}} H_{\mathrm{svr}}(\{x_i\}_{i\in V_1})$ where

$$H_{\mathrm{svr}}\left(\{x_i\}_{i\in V_1}\right) = Q'\left(\{x_i\}_{i\in V_1}\right) + \lambda \sum_{k\in M_1} \left[C'_k\left(\{x_i\}_{i\in V_1}\right)\right]^2 + \tau \sum_{i\in V_2} R_i(\{x_j\}_{j\in V_1})[R_i(\{x_j\}_{j\in V_1}) - 1]. \tag{15}$$

Then, for $i \in V_2$, $x_i = R_i(\{x_j\}_{j\in V_1})$.

Lemma 3: Problem 3 is equivalent to Problem 4 under the conditions that

$$a_{j,i} \in \mathbb{Z} \text{ for } i \in V_2 \text{ and } j \in V_1 \cup \{0\}. \tag{16}$$

Proof of Lemma 3 For a sufficiently large λ, $C'_k(\{x_i\}_{i\in V_1})$ is zero for $k \in M_1$. Similarly, for a sufficiently large τ, for $i \in V_2$, $R_i(\{x_j\}_{j\in V_1})[R_i(\{x_j\}_{j\in V_1}) - 1]$ is zero under the conditions in (16). This is because the conditions restrict the range of $R_i(\{x_j\}_{j\in V_1})$ to an integer value. Then, $R_i(\{x_j\}_{j\in V_1})[R_i(\{x_j\}_{j\in V_1}) - 1] = 0$ indicates $R_i(\{x_j\}_{j\in V_1}) \in \{0,1\}$. Thus, Problem 4 is reduced to finding the sets of $\{x_i\}_{i\in V}$ that minimize $Q'(\{x_i\}_{i\in V_1})$ under the constraints that $C'_k(\{x_i\}_{i\in V_1}) = 0$ for $k \in M_1$ and $x_i = R_i(\{x_j\}_{j\in V_1})$ for $i \in V_2$, which is nothing but Problem 3.

Lemma 2 shows an equivalence of Problems 1 and 3 and Lemma 3 shows an equivalence of Problems 3 and 4 under the conditions in (16). Thus, we obtain the following theorem.

Theorem 4: Problems 1 and 4 are equivalent under the conditions in (16).

The linearity of $R_i(\{x_j\}_{j\in V_1})$ (see (5)) leads to the following proposition.

Proposition 5: $H_{\mathrm{svr}}(\{x_i\}_{i\in V_1})$ is a QUBO when $Q(\{x_i\}_{i\in V})$ is given in the QUBO form.

Theorem 4 and Proposition 5 give the corollary:

Corollary 6: (Optimality) The optimal solutions of Problem 1 are obtained by solving the QUBO problem of $H_{\mathrm{svr}}(\{x_i\}_{i\in V_1})$ in Problem 4 when the conditions in (16) are satisfied and $Q(\{x_i\}_{i\in V})$ is a QUBO.

Since $V_1 \subseteq V$, the SVR method requires fewer spins to formulate the QUBO than the penalty method.

Algorithm 1: How to Formulate Problem 4 from Problem 1.

1: $V_1 \leftarrow V$, $V_2 \leftarrow \emptyset$, $M_1 \leftarrow M$, and $M_2 \leftarrow \emptyset$
2: $C'_k(\{x_m\}_{m\in V_1}) \leftarrow C_k(\{x_m\}_{m\in V})$ for $k \in M$
3: **while** $\exists k \in M_1$ and $\exists i \in V_1$ s.t. $C'_k(\{x_m\}_{m\in V_1}) = 0$ is written as $x_i = a_0 + \sum_{j\in V_1} a_j x_j$ where $a_j \in \mathbb{Z}$ for $j \in V_1 \cup \{0\}$ **do**
4: $V_1 \leftarrow V_1 \setminus \{i\}$, $V_2 \leftarrow V_2 \cup \{i\}$, $M_1 \leftarrow M_1 \setminus \{k\}$, $M_2 \leftarrow M_2 \cup \{k\}$
5: $R_i(\{x_m\}_{m\in V_1}) \leftarrow a_0 + \sum_{j\in V_1} a_j x_j$
6: $C'_\ell(\{x_m\}_{m\in V_1}) \leftarrow C'_\ell(\{x_m\}_{m\in V_1}, R_i(\{x_m\}_{m\in V_1}))$ for $\ell \in M_1$
7: $R_j(\{x_m\}_{m\in V_1}) \leftarrow R_j(\{x_m\}_{m\in V_1}, R_i(\{x_m\}_{m\in V_1}))$ for $j \in V_2 \setminus \{i\}$
8: **end while**
9: return $C'_k(\{x_m\}_{m\in V_1})$ for $k \in M_1$ and $R_i(\{x_m\}_{m\in V_1}))$ for $i \in V_2$

Algorithm 1 practically formulates Problem 4 from Problem 1. It returns $\{C'_k(\{x_m\}_{m\in V_1})\}_{k\in M_1}$ and $\{R_i(\{x_m\}_{m\in V_1}))\}_{i\in V_2}$. First, set $V_1 = V$, $V_2 = \emptyset$, $M_1 = M$, $M_2 = \emptyset$, and $C'_k(\{x_m\}_{m\in V_1}) = C_k(\{x_m\}_{m\in V})$ for $k \in M$. Then, iteratively update the subsets and the constraints. Each update finds $k \in M_1$ and $i \in V_1$ that satisfy the condition

$$C'_k\left(\{x_m\}_{m\in V_1}\right) = 0 \Leftrightarrow x_i = a_0 + \sum_{j\in V_1} a_j x_j \tag{17}$$

where $a_j \in \mathbb{Z}$ for $j \in V_1 \cup \{0\}$ (Line 3). Then, move k from M_1 to M_2 and i from V_1 to V_2 (Line 4). Accordingly, $R_i(\{x_m\}_{m\in V_1})$ is given, and then $C'_\ell(\{x_m\}_{m\in V_1})$ for $\ell \in M_1$ and $R_j(\{x_m\}_{m\in V_1})$ for $j \in V_2 \setminus \{i\}$ are updated by replacing dependent spin x_i by $R_i(\{x_m\}_{m\in V_1})$ (Lines 5-7). The algorithm ends when $k \in M_1$ and $i \in V_1$ satisfying the condition in (17) does not exist. Then, $\{C'_k(\{x_m\}_{m\in V_1})\}_{k\in M_1}$ and $\{R_i(\{x_m\}_{m\in V_1}))\}_{i\in V_2}$ give H_{svr} (see (15)). Note that $R_i(\{x_m\}_{m\in V_1})$ satisfies the condition in (16).

TABLE II
EXAMPLE 1

(a)

x_2	x_3	x_1	$H_{\rm svr}$
0	0	2	4
0	1	0	0
1	1	1	1
1	1	-1	1

(b)

x_1	x_2	x_3	$H_{\rm svr}$
0	0	1	0
0	1	1/2	$-1/4$
1	0	1/2	$-1/4$
1	1	0	1

In Algorithm 1, the condition in (17) determines which spin variable can take as a dependent spin. The condition is easily checked by looking at the coefficients of a linear equality constraint. The choice of dependent spins is arbitrary as long as the conditions are satisfied and may affect the performances of Ising machines. However, as will be shown in experiments (see Table IX in Section V), the performances are almost independent of the choice of dependent spins.

Below, the SVR method is demonstrated in simple examples of Problem 1.

Example 1: For $x_1, x_2, x_3 \in \{0, 1\}$, minimize x_1 subject to a linear equality constraint $x_1 + x_2 + 2x_3 - 2 = 0$.

The optimal solution is given by $(x_1, x_2, x_3) = (0, 0, 1)$.

To solve the constrained combinatorial optimization problem in the SVR method, let x_1 be a dependent spin. The dependent spin is expressed by independent spins x_2 and x_3 as

$$R_1 (x_2, x_3) = 2 - x_2 - 2x_3. \tag{18}$$

This equation meets the condition in (16) since $a_{0,1} = 2$, $a_{2,1} = -1$, and $a_{3,1} = -2$ are integers. Then, the SVR method gives the QUBO as

$$H_{\rm svr} = R_1 (x_2, x_3) + \tau R_1 (x_2, x_3) [R_1 (x_2, x_3) - 1]. \tag{19}$$

$H_{\rm svr}$ is independent of dependent spin x_1. Problem 4 finds the ground state of $H_{\rm svr}$. Table II(a) lists the values of $H_{\rm svr}$ for each independent-spin configuration (x_2, x_3) when $\tau = 1$. $(x_2, x_3) = (0, 1)$ gives the lowest value of $H_{\rm svr} = 0$. The optimal solution $(x_1, x_2, x_3) = (0, 0, 1)$ is obtained from the lowest-value solution by using $x_1 = R_1(x_2, x_3)$.

Next, using this example, we show that the SVR method is inapplicable when (16) is not satisfied. Let x_3 be a dependent spin instead of x_1. Then,

$$R_3 (x_1, x_2) = 1 - \frac{1}{2}x_1 - \frac{1}{2}x_2 \tag{20}$$

does not meet the condition in (16) since $a_{1,3} = -1/2$ and $a_{2,3} = -1/2$ are not integers. The QUBO in the SVR method reads

$$H_{\rm svr} = x_1 + \tau R_3 (x_1, x_2) [R_3 (x_1, x_2) - 1]. \tag{21}$$

Table II(b) lists the values of $H_{\rm svr}$ for each independent-spin configuration (x_1, x_2) when $\tau = 1$. $H_{\rm svr}$ takes a minimum value of $-1/4$ when $(x_1, x_2) = (0, 1)$ or $(1, 0)$. However, the constraint $x_3 = R_3(x_1, x_2)$ indicates that x_3 takes an inappropriate value of $1/2$. Thus, (16) is generally necessary to obtain the optimal solution in the SVR method. The condition in (16) is hereafter referred to as the SVR condition.

Example 2: For $x_1, x_2 \in \{0, 1\}$, minimize x_1 subject to a linear equality constraint $x_1 + x_2 = 1$.

The optimal solution is given by $(x_1, x_2) = (0, 1)$.

Let x_1 be a dependent spin. Then, $R_1(x_2) = 1 - x_2$ meets the SVR condition. The SVR method gives the QUBO without the constraint term as $H_{\rm svr} = R_1(x_2)$ since

$$R_1(x_2) [R_1(x_2) - 1] = -(1 - x_2)x_2 = 0. \tag{22}$$

Here, we use $x_2^2 = x_2$. The constraint term disappears because the domain of $R_1(x_2)$ matches the domain of a spin variable. Problem 4 minimizes the value of $H_{\rm svr}$. The ground state of $H_{\rm svr}$ is $x_2 = 1$, and then $x_1 = R_1(x_2)$ gives the optimum $(x_1, x_2) = (0, 1)$.

B. Applications

Here, the SVR method is applied to combinatorial optimization problems with typical constraints in one-dimensional and two-dimensional systems. Application to the higher-dimensional systems is straightforward (see Appendix A, available in the online supplemental material). Note that Algorithm 1 explicitly gives a QUBO for the SVR method in other types of constrained combinatorial optimization problems of Problem 1.

1) One-Dimensional System: First, consider Problem 1 with $Q(\{x_i\}_{i\in V})$ in the QUBO form under a linear equality constraint given by

$$C (\{x_i\}_{i\in V}) = \sum_{i\in V} x_i - k = 0, \tag{23}$$

where $k \in \mathbb{N}$. The constraint is called the k-hot constraint. It appears in the G2PP, the job sequencing problem, and graph-coloring problem.

If the r-th spin is set as a dependent spin (i.e., $r \in V_2$), then

$$x_r = -\sum_{i\in V_1} x_i + k =: R_r (\{x_i\}_{i\in V_1}). \tag{24}$$

Here, $R_r(\{x_i\}_{i\in V_1})$ satisfies the SVR condition.

The SVR method and the penalty method respectively give the QUBO $H^{(1)}_{\rm svr}$ and $H^{(1)}_{\rm penalty}$ (see Problem 4 and Problem 2) as

$$\begin{aligned} H^{(1)}_{\rm svr} &= Q' (\{x_i\}_{i\in V_1}) \\ &\quad + \tau^{(1)} R_r (\{x_i\}_{i\in V_1}) [R_r (\{x_i\}_{i\in V_1}) - 1], \\ H^{(1)}_{\rm penalty} &= Q (\{x_i\}_{i\in V}) + \lambda^{(1)} [C (\{x_i\}_{i\in V})]^2, \end{aligned} \tag{25}$$

where $\tau^{(1)}$ and $\lambda^{(1)}$ are the constraint coefficients. The SVR method has one less spin than the penalty method. It should be noted that a different QUBO formulation called the domain-wall method gives a QUBO with $|V| - 1$ spins for $k = 1$ [21]. Regardless, our method is applicable to arbitrary k.

2) Two-Dimensional System: Next, consider Problem 1 with $Q(\{x_{i,s}\}_{(i,s)\in L^{(\rm r)}\otimes L^{(\rm c)}=V})$ in the QUBO form under constraints given by

$$\begin{aligned} C^{(\rm r)}_i (\{x_{j,t}\}) &= \sum_{s\in L^{(\rm c)}} x_{i,s} - k^{(\rm r)} = 0 \quad \forall i \in L^{(\rm r)}, \\ C^{(\rm c)}_s (\{x_{j,t}\}) &= \sum_{i\in L^{(\rm r)}} x_{i,s} - k^{(\rm c)} = 0 \quad \forall s \in L^{(\rm c)}, \end{aligned} \tag{26}$$

where $k^{(\rm r)} \in \mathbb{N}$ and $k^{(\rm c)} \in \mathbb{N}$. Here, the number of 1-valued spins in the FSs are set to $N_1 = k^{(\rm r)}|L^{(\rm r)}| = k^{(\rm c)}|L^{(\rm c)}|$. The constraints appear in the QAP, GkPP ($k \geq 3$), and the traveling salesman problem.

The QUBO obtained by the SVR method depends on the selection of dependent spins. Using a simple example where the spins in the p-th row or in the q-th column are set as dependent spins (i.e., $V_2 = \{(p,s)\}_{s\in L^{(c)}} \cup \{(i,q)\}_{i\in L^{(r)}}$), the dependent spins are represented by the independent spins as

$$x_{i,q} = k^{(r)} - \sum_{s\in L^{(c)}\setminus\{q\}} x_{i,s} := R_{i,q}\left(\{x_{j,t}\}\right) \text{ for } i \in L^{(r)} \setminus \{p\},$$
$$x_{p,s} = k^{(c)} - \sum_{i\in L^{(r)}\setminus\{p\}} x_{i,s} := R_{p,s}\left(\{x_{j,t}\}\right) \text{ for } s \in L^{(c)} \setminus \{q\}, \tag{27}$$

and

$$\begin{aligned} x_{p,q} &= k^{(c)} - \sum_{i\in L^{(r)}\setminus\{p\}} x_{i,q} \\ &= k^{(c)} - \sum_{i\in L^{(r)}\setminus\{p\}} \left(k^{(r)} - \sum_{j\in L^{(c)}\setminus\{q\}} x_{i,s} \right) \\ &= \sum_{(i,s)\in V_1} x_{i,s} + k^{(r)} + k^{(c)} - N_1 =: R_{p,q}\left(\{x_{j,t}\}\right). \end{aligned} \tag{28}$$

Equations (27) and (28) satisfy the SVR condition.

The SVR method and the penalty method give the QUBO $H^{(2)}_{\rm svr}$ and $H^{(2)}_{\rm penalty}$ (see Problems 4 and 2), respectively, as

$$\begin{aligned} H^{(2)}_{\rm svr} &= Q'\left(\{x_{i,s}\}_{(i,s)\in V_1}\right) \\ &\quad + \tau^{(2)} \sum_{(i,s)\in V_2} R_{i,s}\left(\{x_{j,t}\}\right)\left[R_{i,s}\left(\{x_{j,t}\}\right) - 1\right], \\ H^{(2)}_{\rm penalty} &= Q\left(\{x_{i,s}\}_{(i,s)\in V}\right) \\ &\quad + \lambda^{(2)} \sum_{\ell\in\{r,c\}} \sum_{i\in L^{(\ell)}} \left[C^{(\ell)}_i\left(\{x_{j,t}\}\right)\right]^2, \end{aligned} \tag{29}$$

where $\tau^{(2)}$ and $\lambda^{(2)}$ are the constraint coefficients. The SVR method gives the QUBO with $(|L^{(r)}|-1)(|L^{(c)}|-1)$ spins. By contrast, the penalty method gives the QUBO with $|L^{(r)}||L^{(c)}|$ spins. The domain-wall method formulates the QUBO with $(|L^{(r)}|-1)|L^{(c)}|$ spins when $k^{(r)} = 1$ [25] (see also Appendix B, available in the online supplemental material). The SVR method at $|L^{(r)}| = |L^{(c)}|$ and $k^{(r)} = k^{(c)} = 1$ is equivalent to the inserted method in [26]. Hence, our proposed method is superior to other known methods in terms of the number of spins.

IV. QUBO Formulation by Spin-Variable Reduction Method

This section formulates the QUBO of combinatorial optimization problems by the SVR method. We consider the G2PP as examples of the one-dimensional problems and the GkPP ($k \geq 3$) and the QAP as examples of the two-dimensional problems.

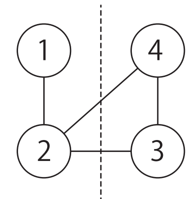

Fig. 2. Example of the four-vertex graph 2-partitioning problem. Dashed line denotes the optimum solution.

A. G2PP

The G2PP is specified by sets of vertices and edges, $V_{\rm g2p}$ and $E_{\rm g2p}$ with even $|V_{\rm g2p}|$. It partitions the vertex set into two subsets of equal size such that the number of edges connecting the two subsets is minimized. It is an NP-hard problem [27].

Here, the G2PP is transformed into Problem 1. First, we place a spin $x_i \in \{0,1\}$ for each vertex $i \in V_{\rm g2p}$ and set $x_i = 0$ when vertex i is in the first subset and $x_i = 1$ when vertex i is in the second subset. The objective function (i.e., the number of edges connecting the two subsets) is given as [12], [28]

$$Q_{\rm g2p}\left(\{x_i\}\right) = -2 \sum_{(i,j)\in E_{\rm g2p}} x_i x_j + \sum_{i\in V_{\rm g2p}} k_i x_i, \tag{30}$$

where k_i denotes the degree of vertex i. The constraint is expressed as

$$C_{\rm g2p} = \sum_{i\in V_{\rm g2p}} x_i - \frac{|V_{\rm g2p}|}{2} = 0. \tag{31}$$

The constraint is satisfied when half of the spins are 0 and the other half are 1. Equations (31) and (23) with $k = |V_{\rm g2p}|/2$ have the same form. Hence, (25) gives the QUBOs for the SVR method and the penalty method.

Next, we demonstrate the SVR method for an undirected graph with four spins (see Fig. 2). The optimal solutions are given by $(x_1, x_2, x_3, x_4) = (0,0,1,1)$ or (1,1,0,0). We set x_1 as a dependent spin. Namely,

$$R_1\left(\{x_i\}\right) = 2 - (x_2 + x_3 + x_4). \tag{32}$$

Then the objective function for the SVR method is obtained as

$$\begin{aligned} Q'_{\rm g2p}\left(\{x_i\}\right) &= Q_{\rm g2p}\left(R_1\left(\{x_i\}\right), x_2, x_3, x_4\right) \\ &= -2x_3x_4 + x_3 + x_4 + 2. \end{aligned} \tag{33}$$

The objective function is independent of dependent spin x_1. The constraint is described by

$$\begin{aligned} H_{\rm c,g2p} &= R_1\left(\{x_i\}\right)\left[R_1\left(\{x_i\}\right) - 1\right] \\ &= \left[2 - (x_2 + x_3 + x_4)\right]\left[1 - (x_2 + x_3 + x_4)\right]. \end{aligned} \tag{34}$$

Overall, the SVR method gives a QUBO as

$$\begin{aligned} H_{\rm g2p} = &-2x_3x_4 + x_3 + x_4 + 2 \\ &+ \tau^{(1)}\left[2 - (x_2 + x_3 + x_4)\right]\left[1 - (x_2 + x_3 + x_4)\right]. \end{aligned} \tag{35}$$

Table III lists the values of $Q'_{\rm g2p}$, $H_{\rm c,g2p}$ and $H_{\rm g2p}$ for each independent-spin configuration (x_2, x_3, x_4) when $\tau^{(1)} = 2$.

TABLE III
FOUR-SPIN G2PP

x_2	x_3	x_4	x_1	Q'_{g2p}	$H_{\mathrm{c,g2p}}$	H_{g2p}
0	0	0	2	2	2	6
0	0	1	1	3	0	3
0	1	0	1	3	0	3
0	1	1	0	2	0	2
1	0	0	1	2	0	2
1	0	1	0	3	0	3
1	1	0	0	3	0	3
1	1	1	−1	2	2	6

$(x_2, x_3, x_4) = (0, 1, 1)$ or $(1, 0, 0)$ gives the lowest value of $H_{\mathrm{g2p}} = 2$. Since the dependent-spin value is calculated by $x_1 = R_1(\{x_i\})$, the optimal solutions $(x_1, x_2, x_3, x_4) = (0, 0, 1, 1)$ and (1,1,0,0) are obtained from the lowest-value solutions.

B. GkPP

The GkPP ($k \geq 3$) is an extension of the G2PP. It partitions the vertex set of graph $G_{\mathrm{gkp}} = (V_{\mathrm{gkp}}, E_{\mathrm{gkp}})$ into k subsets of equal size such that the number of edges connecting the different subsets is minimized. Here, we assume $|V_{\mathrm{gkp}}|/k$ to an integer.

The GkPP is transformed into Problem 1. First, we place spin $x_{i,s} \in \{0, 1\}$ for $i \in V_{\mathrm{gkp}}$ and $s \in \{1, \ldots, k\} =: K$. The spins denote the subset of each vertex. When $x_{i,s} = 1$, vertex i belongs to the s-th subset. Then the objective function is given as

$$Q_{\mathrm{gkp}} = |E_{\mathrm{gkp}}| - \sum_{(i,j)\in E_{\mathrm{gkp}}} \sum_{s\in K} x_{i,s} x_{j,s}. \quad (36)$$

The constraints are expressed as

$$C^{(1)}_{\mathrm{gkp},i} = \sum_{s\in K} x_{i,s} - 1 = 0 \quad \forall i \in V_{\mathrm{gkp}},$$
$$C^{(2)}_{\mathrm{gkp},s} = \sum_{i\in V_{\mathrm{gkp}}} x_{i,s} - \frac{|V_{\mathrm{gkp}}|}{k} = 0 \quad \forall s \in K. \quad (37)$$

The constraints are satisfied if each vertex belongs to a subset and each subset size is equal to $|V_{\mathrm{gkp}}|/k$. Equation (37) satisfies the form in (26) with $k^{(\mathrm{r})} = 1$ and $k^{(\mathrm{c})} = |V_{\mathrm{gkp}}|/k$. Thus, (29) gives the QUBOs for the SVR method and the penalty method. See Appendix C, available in the online supplemental material, for an extended version of GkPP with inequality constraints.

C. QAP

The QAP assigns facilities to different locations to minimize the total transport cost [29]. The total transport cost is defined as

$$S(\pi) = \sum_{i\in L_{\mathrm{qap}}} \sum_{j\in L_{\mathrm{qap}}} f_{i,j} d_{\pi(i),\pi(j)} \quad (38)$$

where $|L_{\mathrm{qap}}| =: n_{\mathrm{fac}}$, π denotes a permutation, and $L_{\mathrm{qap}} \otimes L_{\mathrm{qap}} =: V_{\mathrm{qap}}$ represents the set of facilities and locations. Here, $f_{i,j}$ indicates the flow amount between facilities i and j, while $d_{s,t}$ denotes the distance between locations s and t. The QAP is an NP-hard problem [30].

Here, the QAP is transformed into Problem 1. First, we place spin $x_{i,s} \in \{0, 1\}$ for each $(i, s) \in V_{\mathrm{qap}}$. The spins denote the location of each facility. When $x_{i,s} = 1$, facility i is placed at location s. Then the objective function (i.e., $S(\pi)$) is given as [12], [14]

$$Q_{\mathrm{qap}} = \sum_{(i,s)\in V_{\mathrm{qap}}} \sum_{(j,t)\in V_{\mathrm{qap}}} f_{i,j} d_{s,t} x_{i,s} x_{j,t}. \quad (39)$$

The constraints are expressed as

$$C^{(\mathrm{c})}_{\mathrm{qap},i} = \sum_{s\in L_{\mathrm{qap}}} x_{i,s} - 1 = 0 \quad \forall i \in L_{\mathrm{qap}},$$
$$C^{(\mathrm{r})}_{\mathrm{qap},s} = \sum_{i\in L_{\mathrm{qap}}} x_{i,s} - 1 = 0 \quad \forall s \in L_{\mathrm{qap}}. \quad (40)$$

If all the constraints are satisfied, the permutation is described as $x_{i,s} = 1$ when $s = \pi(i)$. Otherwise $x_{i,s} = 0$. Equation (40) adopts the form in (26) with $k^{(\mathrm{r})} = k^{(\mathrm{c})} = 1$. Consequently, (29) gives the QUBOs for the SVR method and the penalty method.

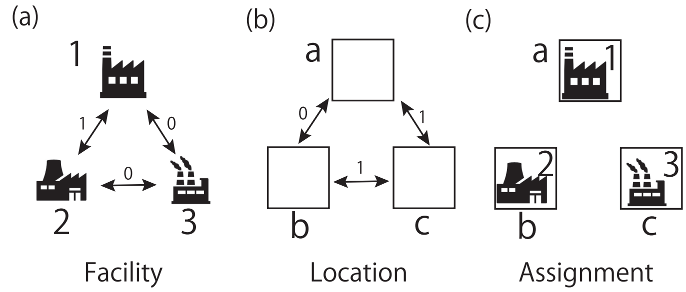

Fig. 3. Example of a QAP with three facilities and locations. Values on each arrow denote (a) $f_{i,j}$ for $i, j \in \{1, 2, 3\}$ and (b) $d_{s,t}$ for $s, t \in \{a, b, c\}$. (c) Optimal assignment for the location of the facilities.

As an example, we demonstrate the SVR method in a QAP with three facilities and locations. We set $(f_{1,2}, f_{2,3}, f_{1,3}) = (1, 0, 0)$ and $(d_{a,b}, d_{b,c}, d_{a,c}) = (0, 1, 1)$ (see Fig. 3). We assumed that $f_{i,j}$ and $d_{s,t}$ are symmetric (i.e., $f_{i,j} = f_{j,i}$ and $d_{s,t} = d_{t,s}$) and the diagonals are zero (i.e., $f_{i,i} = 0$ and $d_{s,s} = 0$). There are two optimal solutions. One allocates facility 1 to location a, facility 2 to location b, and facility 3 to location c and the other allocates facility 1 to location b, facility 2 to location a, and facility 3 to location c. The total transport cost in the optimal solutions is 0. Here, we set the dependent spins as $V_2 = \{(1, a), (1, b), (1, c), (2, a), (3, a)\}$. Namely,

$$\begin{aligned} R_{1,a}(\{x_{i,s}\}) &= x_{2,b} + x_{2,c} + x_{3,b} + x_{3,c} - 1, \\ R_{1,b}(\{x_{i,s}\}) &= -x_{2,b} - x_{3,b} + 1, \\ R_{1,c}(\{x_{i,s}\}) &= -x_{2,c} - x_{3,c} + 1, \\ R_{2,a}(\{x_{i,s}\}) &= -x_{2,b} - x_{2,c} + 1, \\ R_{3,a}(\{x_{i,s}\}) &= -x_{3,b} - x_{3,c} + 1. \end{aligned} \quad (41)$$

The objective function is obtained as

$$\begin{aligned} Q'_{\mathrm{qap}} &= Q_{\mathrm{qap}}\left(\{x_{i,s}\}_{(i,s)\in V_1}, \{R_{i,s}(\{x_{j,t}\})\}_{(i,s)\in V_2}\right) \\ &= 2\,(2x_{2,c}x_{3,c} - x_{3,c} + 1). \end{aligned} \quad (42)$$

The constraint term is described by

$$H_{\mathrm{c,qap}} = \sum_{(i,s)\in V_2} R_{i,s}(\{x_{j,t}\})\,[R_{i,s}(\{x_{j,t}\}) - 1]. \quad (43)$$

TABLE IV
QAP With Three Facilities and Locations

$x_{2,b}$	$x_{3,b}$	$x_{2,c}$	$x_{3,c}$	$x_{1,a}$	$x_{1,b}$	$x_{1,c}$	$x_{2,a}$	$x_{3,a}$	Q'_{qap}	$H_{\mathrm{c,qap}}$	H_{qap}
0	0	0	0	−1	1	1	1	1	2	2	6
0	0	0	1	0	1	0	1	0	0	0	0
0	0	1	0	0	1	0	0	1	2	0	2
0	0	1	1	1	1	−1	0	0	4	2	8
0	1	0	0	0	0	1	1	0	2	0	2
0	1	0	1	1	0	0	1	−1	0	2	4
0	1	1	0	1	0	0	0	0	2	0	2
0	1	1	1	2	0	−1	0	−1	4	6	16
1	0	0	0	0	0	1	0	1	2	0	2
1	0	0	1	1	0	0	0	0	0	0	0
1	0	1	0	1	0	0	−1	1	2	2	6
1	0	1	1	2	0	−1	−1	0	4	6	16
1	1	0	0	1	−1	1	0	0	2	2	6
1	1	0	1	2	−1	0	0	−1	0	6	12
1	1	1	0	2	−1	0	−1	0	2	6	14
1	1	1	1	3	−1	−1	−1	−1	4	14	32

Overall, the QUBO is given as

$$H_{\mathrm{qap}} = 2(2x_{2,c}x_{3,c} - x_{3,c} + 1) + \tau^{(2)} \sum_{(i,s)\in V_2} R_{i,s}(\{x_{j,t}\})[R_{i,s}(\{x_{j,t}\}) - 1]. \quad (44)$$

Table IV lists the values of Q'_{qap}, $H_{\mathrm{c,qap}}$, and H_{qap} for each independent-spin configuration $(x_{2,b}, x_{3,b}, x_{2,c}, x_{3,c})$ when $\tau^{(2)} = 2$. $(x_{2,b}, x_{3,b}, x_{2,c}, x_{3,c}) = (0, 0, 0, 1)$ or $(1, 0, 0, 1)$ gives the lowest value of $H_{\mathrm{qap}} = 0$. The lowest-value solutions give the optimal solutions by calculating the dependent-spin values using $x_{i,s} = R_{i,s}(\{x_{j,t}\})$ for $(i, s) \in V_2$.

V. Experimental Evaluation Using Ising Machines

This section compares the performances of the SVR method and the penalty method for GkPPs and QAPs when an SA-based Ising machine or a QA-based Ising machine is used. Additionally, we show the performance of the domain-wall method for GkPPs ($k \geq 3$) and QAPs (see the QUBO formulation in Appendix B, available in the online supplemental material).

A. Set Up of Ising Machines

SA [31], [32], [33] and QA [34], [35] are meta-heuristic algorithms that address combinatorial optimization problems. Ising machines based on SA or QA are designed to solve QUBO problems. Hereafter, the SA-based Ising machine [7] and the QA-based Ising machine [36] are referred to as classical IM and quantum IM, respectively. The classical and quantum IM hardware embed a maximum of 130,000 spins [2] and 180 spins on a complete graph, respectively [7], [37]. The IM experiments run on a MacBook Pro with a 2.8GHz quad core Intel Core i7 (16GB RAM) using Python 3.7.6 as the implementation language.

Table V lists the parameter sets for classical IM and quantum IM. The annealing time was set to 1 second for the classical IM and 20μ second for the quantum IM. The number of runs was set to 10 for the classical IM and 1,000 for the quantum IM. The remaining values were set to the default for each IM.

[2]The number of maximum spins depends on the users. Here, the number of spins available is restricted to 8,000 on a complete graph.

TABLE V
Parameters for Ising Machines

	Classical IM	Quantum IM
annealing time	1sec	20μsec
number of runs	10	$1,000$

B. Methods

We adopted GkPPs and QAPs to compare the performances of the SVR method, the penalty method, and the domain-wall method. Note that the domain-wall method is inapplicable to the G2PP. The G2PP instances are specified by the undirected graph $G_{\mathrm{g2p}} = (V_{\mathrm{g2p}}, E_{\mathrm{g2p}})$. For each problem size $|V_{\mathrm{g2p}}|$, 20 undirected graphs with edge density of 0.5 were generated. Connecting the vertices i and j in V_{g2p} with half probability created the undirected graphs. The QUBO for a G2PP has the constraint coefficient $\tau^{(1)}$ in the SVR method and $\lambda^{(1)}$ in the penalty method ((25)). The argument in [12] gives a sufficient condition to satisfy the constraint in the ground state of the QUBO as

$$\lambda^{(1)} = \min\left(\max_{i\in V_{\mathrm{g2p}}} k_i, \frac{|V_{\mathrm{g2p}}|}{2}\right). \quad (45)$$

The same argument holds for the SVR method. Thus, $\tau^{(1)}$ and $\lambda^{(1)}$ are set as the value of the right-hand side of (45) in both the classical and quantum IM experiments.

GkPP ($k \geq 3$) generated 5 undirected graphs with edge density of 0.5 for each graph size and QAP used instances in Refs. [38], [39]. The QAP instances are respectively called nug-n_{fac} and lipa-n_{fac}-a. The constraint coefficient in the QUBO for a GkPP and a QAP is $\tau^{(2)}$ in the SVR method, $\lambda^{(2)}$ in the penalty method (29), and κ in the domain-wall method (Appendix B, available in the online supplemental material). The constraint coefficients were determined using the procedure in [40] in classical IM experiments. We calculated the FS rate and the cost function. The FS rate represents the ratio of the number of obtained FSs to the number of runs. The cost function was calculated for the FSs. A smaller value of the cost function indicates a better performance. The FS rate and cost function typically increase with the constraint coefficients [15].

TABLE VI
RESULTS OF THE SVR METHOD AND THE PENALTY METHOD IN G2PPS USING A CLASSICAL IM

	SVR				Penalty			
$\lvert V_{\mathrm{g2p}}\rvert$	N_{L}	C_{ave}	$C_{\min}$	FS rate	N_{L}	C_{ave}	$C_{\min}$	FS rate
8	7	5.55(1)	5.55	1.0	8	5.55(1)	5.55	1.0
16	15	23.1(1)	23.1	1.0	16	23.1(1)	23.1	1.0
32	31	97.5(1)	97.5	1.0	32	97.5(1)	97.5	1.0
64	63	424.3(1)	424.3	1.0	64	424.3(1)	424.3	1.0
128	127	1795.5(1)	1795.5	1.0	128	1795.5(1)	1795.5	1.0
256	255	7445.3(1)	7445.3	1.0	256	7445.3(1)	7445.3	1.0
512	511	30647.4(0.9995)	30647.1	1.0	512	30661.5(1)	30648.8	1.0
1024	1023	124944.3(0.9987)	124940.8	1.0	1024	125103.1(1)	125029.1	1.0
2048	2047	506924.6(0.9986)	506869.2	1.0	2048	507652.2(1)	507513.6	1.0
4096	4095	2048288.7(0.9969)	2048097.0	1.0	4096	2054769.2(1)	2054039.0	1.0

Values in parentheses denote the rescaled average cost function when the value for the penalty method is set to 1.

The dependences indicate that the constraint coefficient has an optimal value. To systematically determine the optimal value of the constraint coefficient, we set the threshold value of the FS rate as $r_{\mathrm{th}} = 0.8$ and repeatedly solved each problem instance while varying the constraint coefficient. The precision threshold was set to 10. The quantum IM experiments used the optimal value of the constraint coefficient obtained in classical IM experiments.

The classical IM hardware has physical spins on a complete graph architecture [7]. Hence, the number of physical spins on the architecture is identical to the number of logical spins in a QUBO (i.e., $|V_{\mathrm{qubo}}|$). On the other hand, the quantum IM hardware has physical spins on a sparse graph architecture [37]. Therefore, minor embedding is generally required to map G_{qubo} onto the architecture [41], [42]. The minor-embedding process constructs chains of physical spins to represent logical spin states. The physical spins in a chain interact via ferromagnetic coupling. The quantum IM experiments calculated the chain break fraction, which is the ratio of chains whose spins have different values. When the chain-break fraction averaged over the 1,000 runs is greater than 0.1, the ferromagnetic coupling strength in a chain was increased. Here, the number of logical spins and physical spins are denoted by N_{L} and N_{P}, respectively.

Below, the average cost function, the minimum cost function, the FS rate, and N_{P} are used. The average cost function is the value averaged over the FSs, while the minimum cost function is the lowest value among the FSs. In GkPPs, each value was averaged over problem instances. The average cost function and the minimal cost function are denoted by C_{ave} and $C_{\min}$, respectively.

C. Performance Comparison in a Classical IM

Table VI gives the results of G2PP in a classical IM for different $|V_{\mathrm{g2p}}|$ using the SVR method and the penalty method. The SVR method selects x_1 as a dependent spin. Both methods find $C_{\mathrm{ave}} = C_{\min}$ for small sized problems, indicating that the classical IM obtains the optimum for all 20 problem instances. By contrast, the SVR method outperforms the penalty method for large-sized problems. The parenthesis denotes the relative value of C_{ave} in the SVR method when C_{ave} in the penalty method is set to 1. The decrease in the relative value with $|V_{\mathrm{g2p}}|$ indicates the superior performance of the SVR method for larger-sized problems. The FS rate is always 1 in both methods.

Table VII shows the results of GkPP ($k \geq 3$) for various k. The SVR method selects the first-raw spins $\{x_{1,s}\}_{s\in K}$ and the first-column spins $\{x_{i,1}\}_{i\in V_{\mathrm{gkp}}}$ as dependent spins. The small-sized problem (i.e., $|V_{\mathrm{gkp}}| = 12$) finds $C_{\mathrm{ave}} = C_{\min}$ for all the methods. On the other hand, for a large-sized problem (i.e., $|V_{\mathrm{gkp}}| = 360$), the penalty method has worst performance among them. The SVR method is best for large k, and the domain-wall method is best for small k. The constraint coefficients in the penalty method are larger than the others.

Table VIII shows the results of QAP in a classical IM with different n_{fac}. The SVR method selects the first-raw spins $\{x_{1,s}\}_{s\in L_{\mathrm{qap}}}$ and the first-column spins $\{x_{i,1}\}_{i\in L_{\mathrm{qap}}}$ as dependent spins. The SVR method finds the optimum when $n_{\mathrm{fac}} \leq 25$, whereas the penalty method finds the optimum when $n_{\mathrm{fac}} \leq 20$ and the domain-wall method finds the optimum when $n_{\mathrm{fac}} \leq 15$. For larger n_{fac}, the SVR method outperforms the other methods. The domain-wall method fails to yield a FS rate above the threshold value in lipa80a even at $\kappa = 10^4$. The constraint coefficients in the SVR method $\tau^{(2)}$ are smaller than those in the penalty method $\lambda^{(2)}$, especially for lipa-n_{fac}-a problem instances.

Table IX shows the G2PP results of $|V_{\mathrm{g2p}}| = 1024$, the G8PP results of $|V_{\mathrm{g8p}}| = 360$, and the QAP results of lipa50a for 10 different choices of dependent spins. The G2PP randomly selects dependent spin r from the vertex set V_{g2p} and the G8PP and the QAP randomly sets p-th row spins and q-th column spins to dependent spins. In all cases, the SVR method produces smaller average cost functions and minimum cost functions than those of the penalty method and the domain-wall method, indicating that the results shown in Tables VI, VII, and VIII are independent of the choice of dependent spins. In G8PP and QAP, the optimal values of $\tau^{(2)}$ are almost independent of the choice of dependent spins.

D. Performance Comparison in a Quantum IM

Table X gives the results of G2PP in a quantum IM for different $|V_{\mathrm{g2p}}|$ using the SVR method and the penalty method. Both methods provide the same value of $C_{\min}$ as that in a classical IM for $|V_{\mathrm{g2p}}| = 8$. For larger $|V_{\mathrm{g2p}}|$, the quantum IM gives larger values of C_{ave} and $C_{\min}$ than the classical IM. C_{ave} and $C_{\min}$ are almost the same in both methods. By contrast, the FS rate in the SVR method is higher than that in the penalty method. The SVR method has smaller N_{P} than the penalty method.

TABLE VII
RESULTS OF THE SVR METHOD, THE PENALTY METHOD, AND THE DOMAIN-WALL METHOD IN GkPPS ($k \geq 3$) USING A CLASSICAL IM

	SVR					Penalty				
	N_L	$\tau^{(2)}$	C_{ave}	C_{min}	FS rate	N_L	$\lambda^{(2)}$	C_{ave}	C_{min}	FS rate
$\|V_{gkp}\| = 12$										
$k = 3$	22	10	15.4(1)	15.4	1.0	36	10	15.4(1)	15.4	1.0
$k = 4$	33	10	18.8(1)	18.8	1.0	48	10	18.8(1)	18.8	1.0
$k = 6$	55	10	24.0(1)	24.0	1.0	72	10	24.0(1)	24.0	1.0
$\|V_{gkp}\| = 360$										
$k = 3$	718	20	20064.8(0.9959)	20061.4	1.0	1080	40	20146.8(1)	20127.8	1.0
$k = 4$	1077	20	22728.8(0.9982)	22720.0	1.0	1440	30	22768.7(1)	22754.0	0.88
$k = 5$	1436	20	24373.9(0.9890)	24358.6	1.0	1800	30	24645.7(1)	24623.0	1.0
$k = 6$	1795	20	25497.6(0.9816)	25482.6	1.0	2160	30	25974.6(1)	25948.2	1.0
$k = 8$	2513	10	26947.9(0.9930)	26926.0	1.0	2880	20	27137.0(1)	27110.8	1.0
$k = 9$	2872	10	27461.0(0.9888)	27446.0	1.0	3240	20	27772.2(1)	27744.8	1.0
$k = 10$	3231	10	27871.1(0.9863)	27855.2	1.0	3600	20	28258.6(1)	28224.8	0.98
	Domain wall									
	N_L	κ	C_{ave}	C_{min}	FS rate					
$\|V_{gkp}\| = 12$										
$k = 3$	22	10	15.4(1)	15.4	1.0					
$k = 4$	33	10	18.8(1)	18.8	1.0					
$k = 6$	55	10	24.0(1)	24.0	1.0					
$\|V_{gkp}\| = 360$										
$k = 3$	718	10	20059.8(0.9957)	20059.4	1.0					
$k = 4$	1077	10	22711.9(0.9975)	22702.0	1.0					
$k = 5$	1436	10	24352.5(0.9881)	24342.0	1.0					
$k = 6$	1795	10	25485.9(0.9812)	25474.2	1.0					
$k = 8$	2513	10	26958.2(0.9934)	26941.6	1.0					
$k = 9$	2872	10	27470.9(0.9892)	27455.2	1.0					
$k = 10$	3231	10	27887.4(0.9869)	27877.4	1.0					

Values in parentheses denote the rescaled average cost function when the value for the penalty method is set to 1 sentence case.

TABLE VIII
RESULTS OF THE SVR METHOD, THE PENALTY METHOD, AND THE DOMAIN-WALL METHOD IN QAPS USING A CLASSICAL IM. OPT. DENOTES THE OPTIMUM FOUND IN REFS. [43], [44]

		SVR					Penalty				
Problem	Opt.	N_L	$\tau^{(2)}$	C_{ave}	C_{min}	FS rate	N_L	$\lambda^{(2)}$	C_{ave}	C_{min}	FS rate
nug5	50	16	10	50.0(1)	50	1.0	25	20	50.0(1)	50	1.0
nug6	86	25	30	86.0(1)	86	1.0	36	30	86.0(1)	86	1.0
nug7	148	36	40	148.0(1)	148	1.0	49	40	148.0(1)	148	1.0
nug8	214	49	50	214.0(1)	214	1.0	64	60	214.0(1)	214	1.0
nug12	578	121	70	578.0(1)	578	1.0	144	80	578.0(1)	578	1.0
nug15	1150	196	110	1150.0(1)	1150	1.0	225	130	1150.0(1)	1150	1.0
nug20	2570	361	160	2587.2(0.9987)	2570	1.0	400	230	2590.5(1)	2570	0.8
nug25	3744	576	210	3756.6(0.9917)	3744	1.0	625	300	3788.0(1)	3762	0.9
nug30	6124	841	270	6181.8(0.9768)	6146	0.9	900	410	6328.4(1)	6202	1.0
lipa10a	473	81	10	473.0(1)	473	1.0	100	70	473.0(1)	473	1.0
lipa20a	3683	361	20	3683.0(0.9784)	3683	1.0	400	240	3764.1(1)	3730	1.0
lipa30a	13178	841	70	13419.7(0.9975)	13385	1.0	900	530	13453.2(1)	13422	0.9
lipa40a	31538	1521	90	32050.6(0.9973)	31991	1.0	1600	960	32136.7(1)	32089	1.0
lipa50a	62093	2401	110	62943.1(0.9952)	62874	1.0	2500	1490	63243.3(1)	63190	1.0
lipa60a	107218	3481	180	108821.0(0.9969)	108745	1.0	3600	2060	109155.0(1)	109073	0.9
lipa70a	169755	4761	200	172148.9(0.9964)	172044	1.0	4900	2850	172767.2(1)	172694	0.8
lipa80a	253195	6241	250	256531.8(0.9971)	256304	1.0	6400	3580	257283.3(1)	257164	0.8
		Domain wall									
Problem	Opt.	N_L	κ	C_{ave}	C_{min}	FS rate					
nug5	50	20	20	50.0(1)	50	1.0					
nug6	86	30	20	86.0(1)	86	1.0					
nug7	148	42	30	148.0(1)	148	1.0					
nug8	214	56	50	214.0(1)	214	1.0					
nug12	578	132	50	578.0(1)	578	1.0					
nug15	1150	210	80	1150.0(1)	1150	1.0					
nug20	2570	380	140	2628.0(1.0145)	2606	1.0					
nug25	3744	600	210	3874.9(1.0229)	3808	0.9					
nug30	6124	570	270	6388.0(1.0094)	6298.0	0.9					
lipa10a	473	90	40	473.0(1)	473	1.0					
lipa20a	3683	380	90	3770.7(1.0018)	3683	1.0					
lipa30a	13178	570	150	13475.8(1.0017)	13452	0.9					
lipa40a	31538	1560	300	32193.6(1.0018)	32161	1.0					
lipa50a	62093	2450	410	63337.8(1.0015)	63257	0.9					
lipa60a	107218	3540	450	109326.0(1.0016)	109203	1.0					
lipa70a	169755	4830	1530	173524.4(1.0044)	173321	0.9					
lipa80a	253195	6320	10000	–	–	0.0					

Values in parentheses denote the rescaled average cost function when the value for the penalty method is set to 1. FS rates are greater than the threshold value 0.8 except for LIPA80A in domain-wall method.

TABLE IX
PERFORMANCE DEPENDENCES ON THE CHOICE OF DEPENDENT SPINS IN GkPP AND QAP

G2PP ($\|V_{\mathrm{g2p}}\| = 1024$)				G8PP ($\|V_{\mathrm{g8p}}\| = 360$)					QAP (lipa50a)				
r	C_{ave}	$C_{\min}$	FS rate	(p,q)	$\tau^{(2)}$	C_{ave}	$C_{\min}$	FS rate	(p,q)	$\tau^{(2)}$	C_{ave}	$C_{\min}$	FS rate
2	124944.6	124941.0	1.0	$(55,6)$	10	26951.3	26940.8	1.0	$(10,42)$	100	63009.5	62942	1.0
52	124943.9	124940.3	1.0	$(80,4)$	10	26947.8	26929.0	1.0	$(14,11)$	110	63010.2	62815	1.0
66	124945.6	124940.4	1.0	$(105,2)$	10	26945.8	26931.6	1.0	$(15,10)$	110	62901.6	62849	0.8
319	124945.3	124941.6	1.0	$(112,5)$	10	26948.0	26938.0	0.96	$(15,18)$	130	63020.0	62877	1.0
568	124945.9	124941.1	1.0	$(192,3)$	10	26948.2	26932.0	0.98	$(19,39)$	110	62995.9	62913	1.0
573	124943.2	124940.4	1.0	$(204,4)$	10	26948.3	26934.4	0.98	$(26,40)$	110	63023.5	62846	1.0
617	124944.1	124940.2	1.0	$(271,3)$	10	26947.9	26932.2	1.0	$(28,10)$	100	62992.5	62854	1.0
902	124942.8	124940.1	1.0	$(271,6)$	10	26946.4	26931.2	0.98	$(43,45)$	110	63039.1	63001	1.0
1003	124944.0	124940.2	1.0	$(303,5)$	10	26945.3	26931.0	1.0	$(45,24)$	120	63021.8	62917	0.9
1020	124943.7	124940.6	1.0	$(348,2)$	10	26949.0	26935.0	0.98	$(46,38)$	110	63045.4	62964	1.0
Penalty	125103.1	125029.1	1.0	penalty	–	27137.0	27110.8	1.0	Penalty	–	63243.3	63190	1.0
–	–	–	–	Domain-wall	–	26958.2	26941.6	1.0	Domain-wall	–	63337.8	63257	0.9

See the definitions of r in (24) and (p, q) in (27). Last two lines denote the average and minimum cost functions in the penalty method and the domain-wall method for comparison.

TABLE X
RESULTS OF THE SVR METHOD AND THE PENALTY METHOD IN G2PPS USING A QUANTUM IM

	SVR					Penalty				
$\|V_{\mathrm{g2p}}\|$	N_{L}	N_{P}	C_{ave}	$C_{\min}$	FS rate	N_{L}	N_{P}	C_{ave}	$C_{\min}$	FS rate
8	7	10.00	7.1315(1.008)	5.5500	0.9949	8	12.00	7.0777(1)	5.5500	0.9026
16	15	35.75	31.522(1.006)	23.500	0.9353	16	39.55	31.325(1)	23.500	0.6514
32	31	132.25	127.03(0.9975)	109.55	0.7464	32	138.85	127.35(1)	111.50	0.4180
64	63	511.05	515.42(1.001)	482.29	0.4611	64	527.15	514.99(1)	483.20	0.2484
128	127	2064.9	2059.5(1.000)	2003.1	0.1645	128	2209.50	2059.9(1)	2006.1	0.0848

Values in parentheses denote the rescaled average cost function when the value for the penalty method is set to 1.

TABLE XI
RESULTS OF THE SVR METHOD, THE PENALTY METHOD, AND THE DOMAIN-WALL METHOD IN GkPPS USING A QUANTUM IM. $|V_{\mathrm{GKP}}|$ IS SET TO 12

	SVR						Penalty					
k	N_{L}	N_{P}	$\tau^{(2)}$	C_{ave}	$C_{\min}$	FS rate	N_{L}	N_{P}	$\lambda^{(2)}$	C_{ave}	$C_{\min}$	FS rate
3	22	67.6	10	21.63(1.010)	16.6	0.357	36	180.0	10	21.41(1)	18.2	0.224
4	33	148.6	10	24.66(1.020)	21.8	0.035	48	310.0	10	24.18(1)	22.0	0.007
6	55	398.4	10	–	–	0.001	72	657.6	10	–	–	0.000
	Domain wall											
k	N_{L}	N_{P}	κ	C_{ave}	$C_{\min}$	FS rate						
3	24	81.0	10	21.56(1.007)	16.4	0.277						
4	36	175.8	10	24.58(1.017)	21.8	0.024						
6	60	459.2	10	–	–	0.000						

Values in parentheses denote the rescaled average cost function when the value for the penalty method is set to 1.

Table XI shows the results of GkPP ($k \geq 3$) for $k \in \{3, 4, 6\}$ and $|V_{\mathrm{gkp}}| = 12$. When $k = 6$, any method does not find a FS of all problem instances. When k is 3 or 4, all methods produce approximately the same values of C_{ave} and $C_{\min}$, but the SVR method gives the highest FS rate. The SVR method has the lowest N_{P} of the three methods for all k.

Table XII shows the results of QAP in a quantum IM for different n_{fac}. When n_{fac} is 10 or 12, any method fails to find an FS with the quantum IM. For a smaller number of facilities, all methods generate almost the same values of C_{ave} and $C_{\min}$, but the SVR method gives the highest FS rate. The SVR method has the smallest N_{P} among the three methods.

E. Preprocessing-Time Comparison

Table XIII shows the preprocessing times for generating QUBOs of GkPP and QAP using the SVR method and the penalty method. We used a software package from [7]. For all problems, the SVR method takes longer to preprocess than the penalty method. For G2PPs, the increase rate is approximately 5% regardless of problem size $|V_{\mathrm{g2p}}|$, and preprocessing time scales as $|V_{\mathrm{g2p}}|^2$. The increase rate of GkPPs ($k \geq 4$) increases with k. The processing time scales as k^2 for the SVR method and k for the penalty method. These are explained by QUBO edge number scalings (i.e., $|E_{\mathrm{qubo}}| \sim k^2$ for H_{svr} and $|E_{\mathrm{qubo}}| \sim k$ for H_{penalty}). For QAPs, the increase rate is approximately 30% regardless of problem size n_{fac}, and preprocessing time scales as n_{fac}^4.

VI. DISCUSSION

In a classical IM, the SVR method generates smaller cost functions in both GkPP and QAP than the penalty method. We explain the results in terms of the energy landscape structure. We define the set of FS subspaces $\{S_\alpha\}$ such that S_α is a set of FSs and two FSs are in the same S_α if the single-spin flips can make a transition between the two FSs without passing an infeasible solution. A single-spin flip denotes the change of a spin value from 0 to 1 or vice versa. For a G2PP, the penalty method needs at least two-spin flips to induce a transition between the FSs. Each FS belongs to the different S_α, and thus $|S_\alpha|$ exponentially increases with the number of spins $|V_{\mathrm{g2p}}|$

TABLE XII
RESULTS OF THE SVR METHOD, THE PENALTY METHOD, AND THE DOMAIN-WALL METHOD IN QAPS USING A QUANTUM IM. OPT. DENOTES THE OPTIMUM

		SVR						Penalty					
Problem	Opt.	$N_{\rm L}$	$N_{\rm P}$	$\tau^{(2)}$	$C_{\rm ave}$	$C_{\rm min}$	FS rate	$N_{\rm L}$	$N_{\rm P}$	$\lambda^{(2)}$	$C_{\rm ave}$	$C_{\rm min}$	FS rate
nug5	50	16	40	10	66.74(0.9982)	50	0.321	25	86	20	66.86(1)	50	0.065
nug6	86	25	88	30	114.40(1.014)	86	0.091	36	166	30	112.81(1)	92	0.042
nug7	148	36	173	40	201.40(1.079)	178	0.010	49	343	40	186.67(1)	170	0.006
nug8	214	49	314	50	306.00(0.9839)	270	0.004	64	553	60	311.00(1)	286	0.002
nug12	578	121	2074	70	–	–	0.000	144	2580	80	–	–	0.000
lipa10a	473	81	911	10	–	–	0.000	100	1268	70	–	–	0.000
		Domain wall											
Problem	Opt.	$N_{\rm L}$	$N_{\rm P}$	κ	$C_{\rm ave}$	$C_{\rm min}$	FS rate						
nug5	50	20	53	20	65.84(0.9847)	50	0.283						
nug6	86	30	119	20	107.05(0.9489)	86	0.061						
nug7	148	42	241	30	198.33(1.062)	184	0.006						
nug8	214	56	437	50	264.00(0.8489)	264	0.001						
nug12	578	132	2164	50	–	–	0.000						
lipa10a	473	90	1167	40	–	–	0.000						

Values in parentheses denote the rescaled average cost function when the value for the penalty method is set to 1.

TABLE XIII
PREPROCESSING TIME FOR QUBO GENERATION IN THE SVR METHOD AND THE PENALTY METHOD

G2PP	SVR	penalty	G*k*PP	SVR	penalty
$\lvert V_{\rm g2p}\rvert$	time [s]	time [s]	k	time [s]	time [s]
512	0.976(1.031)	0.946(1)	4	3.091(1.355)	2.282(1)
1024	4.213(1.074)	3.922(1)	6	5.660(1.672)	3.386(1)
2048	17.46(1.036)	16.86(1)	8	9.247(2.009)	4.602(1)
4096	70.47(1.068)	65.99(1)	10	14.07(2.480)	5.673(1)
QAP	SVR	penalty			
$n_{\rm fac}$	time [s]	time [s]			
50	69.41(1.270)	54.66(1)			
60	148.7(1.271)	117.0(1)			
70	275.2(1.342)	205.0(1)			
80	477.8(1.355)	352.5(1)			

Values in parentheses denote the rescaled time when the value for the penalty method is set to 1. G k PP ($K \geq 4$) sets $\lvert V_{\rm GKP}\rvert = 360$.

as $|S_\alpha| = \binom{|V_{\rm g2p}|}{|V_{\rm g2p}|/2} \sim |V_{\rm g2p}|^{-1/2} 2^{|V_{\rm g2p}|}$. By contrast, the SVR method gives $|S_\alpha| = 1$. For example, in the four-spin G2PP (Table III), the transition from FS $(x_2, x_3, x_4) = (0, 0, 1)$ to FS $(0, 1, 1)$ is induced by a single-spin flip of x_3. For a G*k*PP ($k \geq 3$) and QAP, the penalty method needs at least four-spin flips to induce a transition between the FSs. The SVR method reduces $|S_\alpha|$ compared to the penalty method. For example, in a QAP with $n_{\rm fac} = 3$ (Table IV), $(x_{2,b}, x_{3,b}, x_{2,c}, x_{3,c}) = (0, 0, 0, 1)$, $(1, 0, 0, 0)$, and $(1, 0, 0, 1)$ are within the same FS subspace. The reduction of $|S_\alpha|$ efficiently searches lower-energy solutions. In a quantum IM, the SVR method gives a higher FS rate in both G*k*PP and QAP than the other methods. This is attributed to the smaller number of infeasible solutions.

Due to the minor embedding, the SVR method and the penalty method in the quantum IM have the same number of FS subspaces because a multi-spin flip of physical spins is necessary to flip a single logical spin. Therefore, the results for an ideal QA simulation without minor embedding (see the simulation details in Appendix D, available in the online supplemental material) should be compared with the quantum IM results. Fig. 4 shows the annealing-time dependences of the average cost function and the FS rate in the G2PPs with $|V_{\rm g2p}| = 16$. For the ideal QA, the average cost function approaches the optimum and the FS rate approaches 1 with the annealing time. For a long annealing time, the average cost function is smaller in the SVR method than the penalty method, implying that reducing the number of FS subspaces improves the performance of the ideal QA. Moreover, the SVR method has a higher FS rate than the penalty method at a short annealing time, but at a long annealing time they have the same FS rate. By contrast, the quantum IM results weakly depend on the annealing time. The average cost

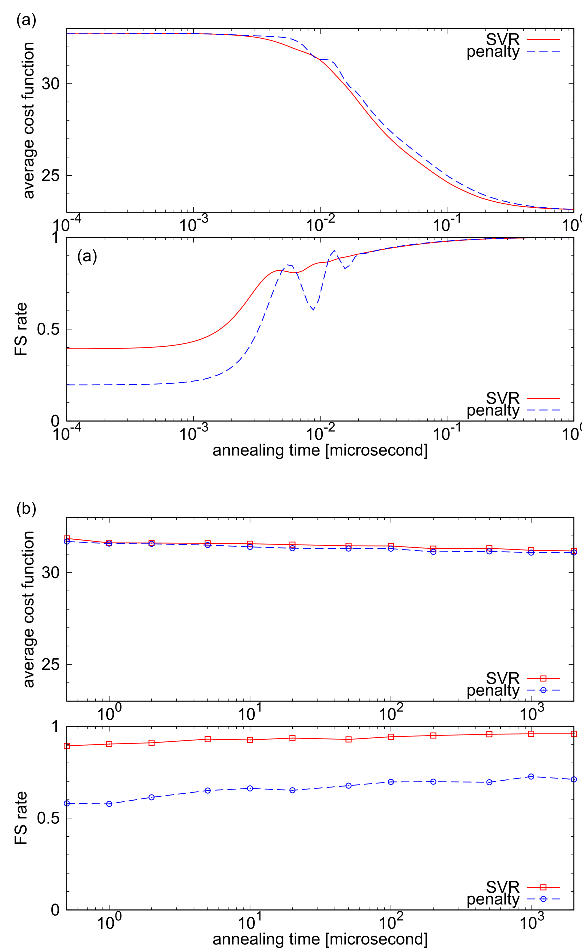

Fig. 4. Annealing-time dependences of the average cost function and the FS rate for (a) ideal QA and (b) quantum IM. Solid and dashed lines denote the results of the SVR method and the penalty method, respectively.

function is smaller and the FS rate is higher as the annealing time increases. Note that ideal QA finds better solutions in less time than the quantum IM. We expect this to be due to the minor embedding because the minor embedding reduces quantum fluctuation effects [17]. An interesting future work is to theoretically and experimentally investigate the effect of minor embedding on the performance.

VII. Conclusion

We propose the SVR method to handle the linear equality constraints for combinatorial optimization problems in the QUBO form. The SVR method reduces the number of spin variables in the QUBO problem compared to the penalty method and the other known methods. Here, we determine the sufficient condition to obtain the optimum of the combinatorial optimization problem in the SVR method and provide QUBO formulations for combinatorial optimization problems with typical constraints in one-dimensional and two-dimensional systems. Therefore, the SVR method extends the application of Ising machines to larger-size combinatorial optimization problems with linear equality constraints. Additionally, we compare the performances of the SVR method, the penalty method, and the domain-wall method in GkPP and QAP using an SA-based Ising machine and a QA-based Ising machine. The SVR method outperforms the other methods in terms of the cost function in an SA-based Ising machine and the FS rate in a QA-based Ising machine.

Acknowledgments

This work was supported in part by JST CREST under Grant JPMJCR19K4, Japan and in part by KAKENHI under Grant 21K03391. The authors thank Shu Tanaka for his fruitful comments. T. S. thanks the Supercomputer Center, the Institute for Solid State Physics, The University of Tokyo for use of the facilities.

References

[1] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Chelmsford, MA, USA: Courier Corporation, 1998.

[2] 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," *Front. Phys.*, vol. 7, 2019, Art. no. 48.

[3] 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.

[4] 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.

[5] T. Okuyama, T. Sonobe, K.-I. Kawarabayashi, and M. Yamaoka, "Binary optimization by momentum annealing," *Phys. Rev. E*, vol. 100, Jul. 2019, Art. no. 012111.

[6] K. Yamamoto et al., "7.3 STATICA: A 512-spin 0.25M-weight full-digital annealing processor with a near-memory all-spin-updates-at-once architecture for combinatorial optimization with complete spin-spin interactions," in *Proc. IEEE Int. Solid- Statist. Circuits Conf.*, 2020, pp. 138–140.

[7] Fixstars Amplify AE. 2022. [Online]. Available: https://amplify.fixstars.com/en/engine

[8] M. W. Johnson et al., "Quantum annealing with manufactured spins," *Nature*, vol. 473, no. 7346, pp. 194–198, 2011.

[9] M. Maezawa et al., "Toward practical-scale quantum annealing machine for prime factoring," *J. Phys. Soc. Jpn.*, vol. 88, no. 6, 2019, Art. no. 061012.

[10] N. Mohseni, P. L. McMahon, and T. Byrnes, "Ising machines as hardware solvers of combinatorial optimization problems," *Nature Rev. Phys.*, vol. 4, no. 6, pp. 363–379, 2022.

[11] H. Neven, V. S. Denchev, G. Rose, and W. G. Macready, "Training a binary classifier with the quantum adiabatic algorithm," 2008, *arXiv:0811.0416*.

[12] A. Lucas, "Ising formulations of many NP problems," *Front. Phys.*, vol. 2, 2014, Art. no. 5.

[13] F. Neukart, G. Compostella, C. Seidel, D. von Dollen, S. Yarkoni, and B. Parney, "Traffic flow optimization using a quantum annealer," *Front. ICT*, vol. 4, 2017, Art. no. 29.

[14] F. Glover, G. Kochenberger, and Y. Du, "A tutorial on formulating and using QUBO models," 2018, *arXiv: 1811.11538*.

[15] 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, 2019, Art. no. 061010.

[16] K. Kitai et al., "Designing metamaterials with quantum annealing and factorization machines," *Phys. Rev. Res.*, vol. 2, Mar. 2020, Art. no. 013319. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevResearch.2.013319

[17] E. Andriyash, Z. Bian, F. A. Chudak, A. D. King, and W. G. Macready, "Boosting integer factoring performance via quantum annealing offsets," D-Wave, Burnaby, Canada, Tech. Rep. 14-1002A-B, 2016.

[18] 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.*, vol. 71, no. 1, pp. 223–234, Jan. 2022.

[19] T. Shirai, S. Tanaka, and N. Togawa, "Guiding principle for minor-embedding in simulated-annealing-based Ising machines," *IEEE Access*, vol. 8, pp. 210 490–210 502, 2020.

[20] T. Shirai and N. Togawa, "Multi-spin-flip engineering in an Ising machine," *IEEE Trans. Comput.*, May 2022, doi: 10.1109/TC.2022.3178325.

[21] J. Chen, T. Stollenwerk, and N. Chancellor, "Performance of domain-wall encoding for quantum annealing," *IEEE Trans. Quantum Eng.*, vol. 2, Jul. 2021, Art. no. 3102714.

[22] O. Şeker, N. Tanoumand, and M. Bodur, "Digital annealer for quadratic unconstrained binary optimization: A comparative performance analysis," *Appl. Soft Comput.*, vol. 127, 2022, Art. no. 109367. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S156849462200521X

[23] G. Kochenberger et al., "The unconstrained binary quadratic programming problem: A survey," *J. Combinatorial Optim.*, vol. 28, no. 1, pp. 58–81, 2014.

[24] N. Dattani, "Quadratization in discrete optimization and quantum mechanics," 2019, *arXiv: 1901.04405*.

[25] J. Berwald, N. Chancellor, and R. Dridi, "Understanding domain-wall encoding theoretically and experimentally," *Philosoph. Trans. Roy. Soc. A: Math. Phys. Eng. Sci.*, vol. 381, no. 2241, 2023, Art. no. *20210410*. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2021.0410

[26] M. Seelbach Benkner, V. Golyanik, C. Theobalt, and M. Moeller, "Adiabatic quantum graph matching with permutation matrix constraints," in *Proc. IEEE Int. Conf. 3D Vis.*, 2020, pp. 583–592.

[27] S. Edelkamp and S. Schrödl, "Constraint search," in *Heuristic Search*, S. Edelkamp and S. Schrödl, Eds., San Francisco, CA, USA: Morgan Kaufmann, 2012, ch. 13, pp. 571–631. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123725127000134

[28] T. Yoshimura, T. Shirai, M. Tawada, and N. Togawa, "QUBO matrix distorting method for consumer applications," in *Proc. IEEE Int. Conf. Consum. Electron.*, 2022, pp. 01–06.

[29] T. C. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," *Econometrica*, vol. 25, no. 1, pp. 53–76, 1957. [Online]. Available: http://www.jstor.org/stable/1907742

[30] S. Sahni and T. Gonzalez, "P-complete approximation problems," *J. ACM*, vol. 23, no. 3, pp. 555–565, Jul. 1976.

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

[32] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon, "Optimization by simulated annealing: An experimental evaluation; Part I, graph partitioning," *Operations Res.*, vol. 37, no. 6, pp. 865–892, 1989.

[33] 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, 1991.

[34] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," *Phys. Rev. E*, vol. 58, pp. 5355–5363, Nov. 1998.

[35] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Quantum computation by adiabatic evolution," 2000, *arXiv:quant-ph/0001106*.

[36] C. McGeoch and P. Farré, "The advantage system: Performance update," D-Wave, Burnaby, Canada, Tech. Rep. 14-1054A-A, 2021.

[37] K. Boothby, P. Bunyk, J. Raymond, and A. Roy, "Next-generation topology of D-Wave quantum processors," D-Wave, Burnaby, Canada, Tech. Rep. 14-1026A-C, 2019.

[38] C. E. Nugent, T. E. Vollmann, and J. Ruml, "An experimental comparison of techniques for the assignment of facilities to locations," *Operations Res.*, vol. 16, no. 1, pp. 150–173, 1968.

[39] Y. Li and P. M. Pardalos, "Generating quadratic assignment test problems with known optimal permutations," *Comput. Optim. Appl.*, vol. 1, no. 2, pp. 163–184, 1992.

[40] 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. 81 032–81 039, 2021.

[41] V. Choi, "Minor-embedding in adiabatic quantum computation: I. The parameter setting problem," *Quantum Inf. Process.*, vol. 7, no. 5, pp. 193–209, 2008.

[42] V. Choi, "Minor-embedding in adiabatic quantum computation: II. Minor-universal graph design," *Quantum Inf. Process.*, vol. 10, no. 3, pp. 343–353, 2011.

[43] R. E. Burkard, S. E. Karisch, and F. Rendl, "QAPLIB–A quadratic assignment problem library," *J. Glob. Optim.*, vol. 10, no. 4, pp. 391–403, 1997. [Online]. Available: https://www.opt.math.tugraz.at/qaplib/inst.html

[44] É. Taillard, "Quadratic assignment instances," 2022. [Online]. Available: http://mistic.heig-vd.ch/taillard/problemes.dir/qap.dir/qap.html

Tatsuhiko Shirai (Member, IEEE) received the BSci, MSci, and DrSci degrees from the University of Tokyo, in 2011, 2013, and 2016, respectively. He is presently an assistant professor with the Department of Computer Science and Communications Engineering, Waseda University. His research interests include quantum dynamics, statistical mechanics, and computational science. He is a member of the JPS.

Nozomu Togawa (Member, IEEE) received the BEng, MEng, and DrEng degrees from Waseda University, in 1992, 1994, and 1997, respectively, all in electrical engineering. He is presently 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 IPSJ.