Solving Constrained Slot Placement Problems Using an Ising Machine and Its Evaluations

Sho KANAMARU†††, Kazushi KAWAMURA†††, Shu TANAKA††††††, Yoshinori TOMITA††††, Nonmembers, and Nozomu TOGAWA††††, Senior Member

SUMMARY Ising machines have attracted attention, which is expected to obtain better solutions of various combinatorial optimization problems at high speed by mapping the problems to natural phenomena. A slot-placement problem is one of the combinatorial optimization problems, regarded as a quadratic assignment problem, which relates to the optimal logic-block placement in a digital circuit as well as optimal delivery planning. Here, we propose a mapping to the Ising model for solving a slot-placement problem with additional constraints, called a constrained slot-placement problem, where several item pairs must be placed within a given distance. Since the behavior of Ising machines is stochastic and we map the problem to the Ising model which uses the penalty method, the obtained solution does not always satisfy the slot-placement constraint, which is different from the conventional methods such as the conventional simulated annealing. To resolve the problem, we propose an interpretation method in which a feasible solution is generated by post-processing procedures. We measured the execution time of an Ising machine and compared the execution time of the simulated annealing in which solutions with almost the same accuracy are obtained. As a result, we found that the Ising machine is faster than the simulated annealing that we implemented.

key words: Ising machine, Ising model, slot-placement problem, quadratic assignment problem, quadratic unconstrained binary optimization

1. Introduction

1.1 Ising Machine

Recently, various Ising machines, which are also referred to as annealing machines, have been developed to solve combinatorial optimization problems efficiently [11]–[12]. The purpose of combinatorial optimization problem is to search for the optimal combination of decision variables that maximizes or minimizes the given objective function under satisfying the given constraints. As the number of given decision variables increases, the solution space becomes extremely large, thereby making it difficult to perform calculations with conventional von Neumann type computers. In contrast, Ising machines have attracted attention as they can obtain a nearly optimal solution at high speed by mapping the problems to natural phenomena. To utilize Ising machines, we express the objective function of the original combinatorial optimization problem by using the energy function of the Ising model. The Ising model is a theoretical model in statistical physics (see Sect. 3.1). Previously, various combinatorial optimization problems have been mapped to Ising models and solutions have been evaluated via Ising machines [13]–[15].

1.2 Constrained Slot Placement Problem

A slot-placement problem is one of the combinatorial optimization problems to find an optimum item assignment to lattice sockets. The objective function of the slot-placement problem is to minimize the total weighted wiring length (TWWL). The slot-placement problem plays an important role such as in logic-block placement in Field-Programmable Gate Array (FPGA) design [16]–[18]. In practice, several item pairs must be placed within a given distance when severe timing constraints are imposed on them in FPGA design. Furthermore, it also contributes to job assignment problems and relocating the departments in companies [19], [20]. In practice, to maximize work efficiency, several staffs and departments should be close. Thus, in order to take into account such a practical situation, we have to add the constraints above into the slot-placement problem, which is called a constrained slot-placement problem.

The slot-placement problem can be formulated as a quadratic assignment problem [21], [22], which is known as an NP-hard problem. The sub-optimal solutions are usually obtained by using simulated annealing (SA), branch and bound methods, genetic algorithm and tabu search [23], [24]. In addition, the performance of quantum annealing for quadratic assignment problems was investigated [25]. Recently an Ising model mapping of slot-placement problem was proposed [18]. In this paper, based on the previous study [18], Ising machines are applied to the constrained slot-placement problem.

To the best of our knowledge, no studies have applied
to the Ising model to solve the constrained slot-placement problem. In addition, since Ising machines behave stochastically, they do not necessarily obtain the feasible solutions which satisfy the given constraints. If an obtained solution by Ising machines (Ising solution) is not feasible, the Ising solution cannot be transformed to the solution of the original problem. Thus, it is important to develop a method for interpreting the Ising solution violating the given constraints.

1.3 Our Proposal

In this paper, we propose an Ising model mapping to solve the constrained slot-placement problem and an interpretation method for the obtained Ising solutions so that they satisfy the slot-placement constraint. First, we propose an objective function term in the energy function or Hamiltonian of the Ising model that minimizes the TWWL to be placed to the slots. Secondly, we propose slot-placement constraint terms that minimize the energy function when satisfying the slot-placement constraint. Thirdly, we also propose an additional constraint term that increases the energy function when we violate the constraint that several item pairs must be placed within a given distance. It should be noted that Ising solutions do not necessarily satisfy the slot-placement constraint since Ising machines behave stochastically. If an Ising solution does not satisfy the slot-placement constraint, we cannot transform the Ising solution to the solution of the original problem. To resolve the problem, we newly propose an interpretation method for the obtained Ising solutions. In the method, we identify the spins which cause the violation of the slot-placement constraint and modify them so that the Ising solution satisfies the slot-placement constraint.

1.4 Contributions

The contributions of this paper are summarized as follows:

1. We propose an Ising model mapping to solve the constrained slot-placement problem (See Sect. 3).
2. To recover a feasible solution, i.e., a solution satisfying the slot-placement constraint defined in Sect. 2, from an Ising solution violating the hard constraints which must be satisfied, we newly propose an interpretation method (see Sect. 4). Note that, since the Ising machines behave stochastically, the hard constraints are sometimes violated.
3. We measured the execution time of an Ising machine and compared the execution time of the simulated annealing in which solutions with almost the same accuracy are obtained. As a result, we found that the speedup rate of the Ising machine increases as the problem size increases (See Sect. 5).

1.5 Organization of the Paper

The rest of this paper is organized as follows: Section 2 formulates the constrained slot placement problem; Sect. 3 proposes a Quadratic Unconstrained Binary Optimization (QUBO) mapping of the constrained slot-placement problem; Sect. 4 proposes an interpretation method of Ising solutions after annealing; Sect. 5 presents the experimental results and Sect. 6 gives several concluding remarks and future works.

2. Formulation of the Constrained Slot-Placement Problem

Let us define a set \( P \) of \( m \) items as \( P = \{ p_1, p_2, \ldots, p_m \} \) and a set \( S \) of \( n \) slots consisting of \( L_s \) columns and \( L_a \) rows as \( S = \{ s_1, s_2, \ldots, s_n \} \), where \( n = L_s \times L_a \) (see Fig. 1). Here, \( m \leq n \), which means the number of items is less than or equal to that of slots.

Let \( a \) and \( b \) be the slots located at \( a_{1\text{-th}} \) column and \( a_{2\text{-th}} \) column, and \( b_{1\text{-th}} \) column and \( b_{2\text{-th}} \) column, respectively, where \( 1 \leq a_1, b_1 \leq L_s \) and \( 1 \leq a_2, b_2 \leq L_a \). The Manhattan distance between the two slots \( a \) and \( b \) is defined by

\[
l(a, b) = |a_1 - b_1| + |a_2 - b_2|.
\]

Let \( w_{p_i,p_j} \) be the number of wires between two items \( p_i, p_j \in P \), where \( 1 \leq i, j \leq m \). Note that \( w_{p_i,p_j} \) is zero when there are no wires between \( p_i \) and \( p_j \). When an item \( p_i \) is assigned to a slot, the slot is denoted by \( s(p_i) \). The weighted wiring length between \( p_i \) and \( p_j \) is defined by \( (w_{p_i,p_j} \times l(s(p_i), s(p_j))) \). TWWL is defined by

\[
L = \sum_{i=1}^{m} \sum_{j=1}^{m} w_{p_i,p_j} \times l(s(p_i), s(p_j)).
\]

In the constrained slot-placement problem, every item \( p_i \in P \) is assigned to a slot \( s \in S \) such that the assignment satisfies the following three constraints: Firstly, each

| Input |
|----------------------------------|
| Given number of wires between the items and distance between slots \( l = 3, l_a = 2, m = 5 \) |

| Objective |
|----------------------------------|
| Minimize the total number of the wires distance |

| Output |
|----------------------------------|
| Arrangement of items |

Fig. 1 Constrained slot-placement problem \((L_s = 3, L_a = 2, m = 5\) and \(d_{\text{limit}} = 2\)).
item \( p_i \in P \) must be assigned to just one slot. This constraint is called an item assignment constraint. Secondly, at most one item can be assigned to every slot. This constraint is called a slot assignment constraint. Here, these constraints are called the slot-placement constraint. Thirdly, several item pairs must be placed within a given Manhattan distance \( d_{\text{limit}} \). This constraint is called an additional constraint.

Then the constrained slot-placement problem is defined as follows:

**Definition 1.** Given a set of items and a set of slots, the constrained slot-placement problem is to determine item-to-slot assignment so as to minimize the total weighted wiring length (TWWL), under satisfying both the slot-placement constraint and the additional constraint.

**Example 1.** As in Fig. 1, assume that \( m = 5 \) items, \( L_x \times L_y = 3 \times 2 \) slots and \( d_{\text{limit}} = 2 \) are given. The number of wires between items are also given in this figure. An example of constrained slot assignment is given as in the Fig. 1, where the TWWL is minimized under satisfying the given constraints. The TWWL in this case is given by

\[
\begin{align*}
\sum_{i=1}^{m} \sum_{j=1}^{m} u_{p_i,p_j} l(s(p_i), s(p_j)) \\
= 5 \times 1 + 2 \times 2 + 3 \times 1 \\
+ 1 \times 2 + 2 \times 1 + 2 \times 1 + 2 \times 1 \\
= 20.
\end{align*}
\]

### 3. QUBO Model Mapping of the Constrained Slot-Placement Problem

#### 3.1 Ising Model

The Ising model is one of the basic models in statistical mechanics to investigate the cooperative phenomena of many-body systems [13]. This model is defined on an undirected graph \( M = (V,E) \), where \( V \) and \( E \) are a set of vertices and a set of connecting edges between vertices, respectively. When the two vertices \( i,j \in V \) are connected, \( (i,j) \in E \) represents the connecting edges between them. Here we define a degree of freedom called spin. Spin is located on every body system\[13\]. This model is defined on an undirected graph \( M \).

**Example 1.** Here, the interaction coefficient between the two spins \( \sigma_i \) and \( \sigma_j \), and \( \sigma_i \) is the coefficient of the linear term of the variable \( x_i \). As in Eq. (3), both \( q_{ij} \) and \( c_i \) are real values. Here, the third term of r.h.s. of Eq. (5) represents a constant value which does not depend on the binary variables \( \{x_i\} \). Since the constant value does not affect the efficiency of the computation, we omit it hereafter.

### 3.2 Constrained Slot Placement Mapping to the QUBO Model

In this subsection, the constrained slot-placement problem is mapped to the QUBO model. We firstly introduce a binary variable \( x_{i,a} \) \((1 \leq i \leq m \text{ and } 1 \leq a \leq n)\) as follows:

\[
x_{i,a} = \begin{cases} 
1 & \text{(if the item } p_i \text{ is assigned to the slot } a, \text{i.e., } a = s(p_i) \text{).} \\
0 & \text{(otherwise)}
\end{cases}
\]  

Using the binary variables, we map the constrained slot-placement problem onto the QUBO model as follows:

#### 3.2.1 Objective Function Term

Since we consider the cost function represented by Eq. (2), the objective function \( \mathcal{H}_1 \) of the QUBO model can be expressed as follows:

\[
\mathcal{H}_1 = \sum_{a=1}^{n} \sum_{b=1}^{m} \sum_{i=1}^{m} \sum_{j=1}^{m} w_{p_i,p_j} l(s(p_i), s(p_j)) x_{i,a} x_{j,b}.
\]  

#### 3.2.2 Item Assignment Constraint

An item must be assigned to one slot, which is expressed as

\[
\sum_{a=1}^{n} x_{i,a} = 1 \text{ for } 1 \leq i \leq m.
\]  

The following Hamiltonian, \( \mathcal{H}_2 \) takes minimum value of 0 when Eq. (8) is satisfied.

\[
\mathcal{H}_2 = \sum_{i=1}^{n} \left(1 - \sum_{a=1}^{n} x_{i,a}\right)^2
\]
able to introduce a term that increases the energy of the QUBO item pairs must be placed within a given distance. In order placement such as in FPGA logic-block placement, several As explained in Sect. 1, when we consider a practical slot-placement constraint and the corresponding interpretations.

3.2.3 Slot Assignment Constraint

At most one item must be assigned to each slot, which is expressed as

$$\sum_{i=1}^{m} x_{i,a} \leq 1 \text{ for } 1 \leq a \leq n.$$  (10)

The following Hamiltonian, $\mathcal{H}_{3}$, takes minimum value of $n/4$ when Eq. (10) is satisfied.

$$\mathcal{H}_{3} = \frac{n}{2} \left( 1 - \sum_{i=1}^{m} x_{i,a} \right)^{2}$$
$$= \sum_{a=1}^{n} \sum_{i=1}^{m} x_{i,a} x_{j,a} - \sum_{a=1}^{n} \sum_{i=1}^{m} x_{i,a}$$
+ const.  (11)

3.2.4 Additional Constraint

As explained in Sect. 1, when we consider a practical slot-placement such as in FPGA logic-block placement, several item pairs must be placed within a given distance. In order to introduce a term that increases the energy of the QUBO model, when this constraint is not satisfied, the binary variable $z_{a,b,i,j}$ is defined as follows.

$$z_{a,b,i,j} = \begin{cases} 
1 & \text{(when the Manhattan distance between the slots } a = s(p_{i}) \text{ and } b = s(p_{j}) \text{ are placed exceeds } d_{\text{lim}}) \\
0 & \text{(otherwise)} 
\end{cases}$$  (12)

Using $z_{a,b,i,j}$, the additional constraint term is defined as follows.

$$\mathcal{H}_{4} = \sum_{a=1}^{n} \sum_{b=1}^{m} \sum_{i=1}^{m} z_{a,b,i,j} x_{i,a} x_{j,b}$$  (13)

Note that $z_{a,b,i,j}$ can be pre-calculated because we can enumerate all the patterns of $a$, $b$, $i$, and $j$ in Eq. (13) beforehand.

3.2.5 Total Energy Function

The total energy function is given by the weighted sum of the energy functions in Eqs. (7), (9), (11), and (13) as follows:

$$\mathcal{H} = \lambda_{1} \mathcal{H}_{1} + \lambda_{2} \mathcal{H}_{2} + \lambda_{3} \mathcal{H}_{3} + \lambda_{4} \mathcal{H}_{4}.$$  (14)

Note that $\lambda_{1}$, $\lambda_{2}$, $\lambda_{3}$, and $\lambda_{4}$ are the hyperparameters, where $\lambda_{1}, \lambda_{2}, \lambda_{3}, \lambda_{4} \geq 0$. Equation (14) successfully gives the QUBO model to the constrained slot-placement problem. When the coefficients of constraint terms is large enough, the constrained slot placement problem can be optimally solved with satisfying the given constraints by minimizing Eq. (14). The fact will be confirmed in Sect. 5.2.

4. Interpretation of the Ising Solution

4.1 Strategy

In general, Ising machines search the ground state but they cannot necessarily obtain the ground state since the behavior of Ising machines is stochastic and we map the problem to the Ising model which uses the penalty method. In addition, solutions which do not satisfy the given constraints are obtained with a finite probability. The situation occurs not only in our case but also in any combinatorial optimization problem with constraints if we use the current version of Ising machines. It is quite important to develop a method for interpreting non-feasible solutions obtained by Ising machines.

The purpose of the interpretation method is not to improve the solutions, but to recover feasible solutions from infeasible solutions. The interpretation method is just applied to the infeasible solution where the slot-placement constraint is not satisfied and modifies it so that it satisfies the slot-placement constraint.

Figure 2 shows examples of Ising solutions which do not satisfy the slot-placement constraint and the corresponding interpretations which will be described below. When slot-placement constraints are satisfied, just one (1)-bit appears in the slot direction and at most one (1)-bit appears in the item direction in the binary variables $\{x_{i,a}\}$ (see Eqs. (8) and (10)). However, it is not always possible to obtain the state that minimizes Eqs. (9) and (11).

Figure 2 shows three cases that do not satisfy the slot-placement constraint. We then propose an interpretation...
method corresponding to these cases as follows:

Resolve item direction overlap (Fig. 2 (a)):
Assume that there are two or more (1)-bits in the \(a\)-th row. Among these (1)-bits in the \(a\)-th row, we pick up one (1)-bit, flip all the other bits to (0) in the \(a\)-th row, and calculate the contribution of the cost function \(H_1\). We try all the patterns of these bit flips in the \(a\)-th row and accept the one which minimizes the contribution of the cost function \(H_1\).

Resolve slot direction overlap (Fig. 2 (b)):
Assume that there are two or more (1)-bits in the \(i\)-th column. Among these (1)-bits in the \(i\)-th column, we pick up one (1)-bit, flip all the other bits to (0) in the \(i\)-th column, and calculate the contribution of the cost function \(H_1\). We try all the patterns of these bit flips in the \(i\)-th column and accept the one which minimizes the contribution of the cost function \(H_1\).

Add an item in slot direction (Fig. 2 (c)):
Assume that all bits are (0) in the \(j\)-th column. We pick up one bit in the \(j\)-th column, flip it to (1), and calculate the contribution of the cost function \(H_1\). We try such a bit flip in the \(j\)-th column in all the feasible patterns and accept the one which minimizes the contribution of the cost function \(H_1\).

4.2 Proposed Method

Assume that we obtain a solution by using the Ising machine for the QUBO model described in Eq. (14).

A set of vertices on which the value of binary variable is 1 in every column in the QUBO model can be expressed by

\[
V_i^+ = \{(i, a) | x_{i,a} = 1, (i, a) \in V, \text{ fixed } i\}
\]

for 1 \(≤ i \leq m\). (15)

A set of vertices on which the value of binary variable is 1 in every row in the QUBO model can be expressed by

\[
V_a^+ = \{(i, a) | x_{i,a} = 1, (i, a) \in V, \text{ fixed } a\}
\]

for 1 \(≤ a \leq n\). (16)

Then the proposed method is described in List 1.

5. Experiments

In this section, several constrained slot-placement problems on the QUBO model in Eq. (14) were solved using an Ising machine, Fujitsu Digital Annealer unit 1 [6]. In addition, we implemented our proposed interpretation method described in List 1 in C and applied to the solutions obtained by the Ising machine. For comparison, we also implemented a simulated annealing method for the constrained slot-placement problem (see the discussion below in detail).

5.1 Setup

In our experiment, we consider the case that \(L_t = L_d = L\) and the number of items is set to be \(m = n/2, m = 3n/4\), or \(m = n\), where \(n = L \times L (3 \leq L \leq 6)\). The largest problem size is 6 columns and 6 rows and 27 items, in which we require \(6 \times 6 \times 27 = 972\) binary variables. The Ising machine used in this experiment has 1,024 spins on the complete graph and hence 972 binary variables are almost the maximum size. The number of wires between any two items \(p_i\) and \(p_j\) is set to \(w_{p_i,p_j} \in [0, 10]\) randomly. 100 trials were conducted for each problem. Furthermore, the additional constraints introduced in Sect. 3.2.4 are applied to randomly selected item pairs, where the number of additional constraints are \(100\) for 1,000 trials. The final temperature

| Table 1 Parameters in the Ising machine. |
|----------------------------------------|
| Parameter | Value |
| Initial spin value | Random |
| Initial temperature | max(weight) \(\times 1,000\) |
| Final temperature | 0.1 |
| Annealing iterations | 50,000 – 100,000 |
| Decay rate | Calculated from above |

weight: Coefficient of QUBO for each problem.
was set to 0.1 which is sufficiently smaller than the coefficient of QUBO for each problem. The number of annealing iterations \( \tau \) was set in the range of 50,000 to 100,000 for each problem. Let \( T_{\text{start}} \) and \( T_{\text{end}} \) be the initial temperature and final temperature, respectively. The temperature decays \( T(t) = rT_{\text{start}} \), where \( r \) is the iteration count. By using the number of annealing iterations \( \tau \), the decay rate \( r \) is represented by

\[
\begin{align*}
  r &= \left( \frac{T_{\text{end}}}{T_{\text{start}}} \right)^{\frac{1}{\tau}}. \\
\end{align*}
\]

In Eq. (14), we set \( \lambda_1 = 1 \), \( \lambda_2 = \lambda_3 = 40 \), and \( \lambda_4 = 20 \) when \( m = 5 \) and \( L_z = 3 \), and as the numbers of items and slots increased, the value of \( \lambda_2 \), \( \lambda_3 \) and \( \lambda_4 \) increased. In contrast, the values of \( \lambda_1 \) was fixed. The value of hyperparameters is shown in Fig. 3. The reason of the tuning \( \lambda_2 \), \( \lambda_3 \) and \( \lambda_4 \) is as follows:

\( \lambda_2 \), \( \lambda_3 \): The number of product terms in the Hamiltonians corresponding to the constraints, that is, Eqs. (9) and (11), are \( O(n^2m) \) and \( O(nm^2) \), respectively. In contrast, the number of product terms in the Hamiltonians corresponding to the cost function of Eq. (7) is \( O(nm^2) \), which is dominant since we consider the case that \( n \sim m \). Then we increase the value of \( \lambda_2 \) and \( \lambda_3 \) as the problem size increases.

\( \lambda_4 \): Equation (13) gives a penalty when the additional constraint is not satisfied, and as the size of the problem increases, TWWL also increases, and the cost also needs to be increased accordingly. Then we increase the value of \( \lambda_4 \) as the problem size increases.

5.1.2 Simulated Annealing for Constrained Slot Placement Problem

For comparison, a simulated annealing (SA) method was implemented in C running on CentOS 7.5 on an Intel Xeon 2.5 GHz CPU with 187GB of memory.

In the simulated annealing, the cost function is given by Eq. (2). Also, in the problem of \( L \) rows and \( L \) columns \( \times \) slots (\( n = L \times L \)), with \( m \) items, if the additional constraint is not satisfied, the following term is added to the cost function by using Eqs. (1) and (12).

### Table 2

| Parameter      | Value           |
|----------------|-----------------|
| Initial solution| Random          |
| Initial temperature | \( m \times n/10 \) |
| Cooling rate    | 0.95            |
| Inner loop      | 1.000           |
| Outer loop      | 100             |

*: Solutions cannot be obtained due to long execution time.

\[
\begin{align*}
  \lambda_4 &= \sum_{a=1}^{n} \sum_{b=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{m} \zeta_{a,b,i,j}(s(p_i), s(p_j)) \\
&\times \delta_{a,s(p_i)}\delta_{b,s(p_j)},
\end{align*}
\]

where \( \delta_{a,s(p_i)} \) is the kronecker delta,

\[
\begin{align*}
  \delta_{a,s(p)} &= \begin{cases} 
  1 & (a = s(p)) \\
  0 & (a \neq s(p)) 
\end{cases}.
\end{align*}
\]

\( \lambda_4 \) is the hyperparameter used in Eq. (14), and \( \lambda_4 \geq 0 \).

Based on [26], the parameters used is summarized in Table 2. The SA method is performed as follows:

**Step 1:** Set the temperature parameter \( T \) to be the initial temperature. Assign all the items to slots randomly satisfying the item and slot constraints.

**Step 2:** Pick up two items randomly and exchange them based on the Metropolis method. The Metropolis method is as follows; Let \( \Delta E \) be the difference between the cost \( E_{\text{trial}} \) when the two items are exchanged, and the cost of the current state, \( E_{\text{current}} \), that is, \( \Delta E = E_{\text{trial}} - E_{\text{current}} \). When \( \Delta E < 0 \), we exchange the two items. On the contrary, when \( \Delta E \geq 0 \), we change the two items with the probability \( p = \exp(-\Delta E/T) \), where \( T \) is the temperature.

**Step 3:** Repeat Step 2, “inner loop” times.

**Step 4:** Reduce the temperature parameter \( T \leftarrow r \times T \), where \( r \) is a cooling rate.

**Step 5:** Repeat Step 2–Step 4, “outer loop” times until a converged result is obtained.

5.2 Results on the Proposed Method

To check the quality of the solution and how to set parameters properly, we have implemented a brute-force search algorithm in Python on CentOS 7.3.1611 and Intel Xeon CPU E7-8855 2.10GHz with 1056GB memory. The optimal solutions obtained by our brute-force search algorithm are shown in Table 3. Our brute-force search algorithm can obtain the optimal solution up to \( 4 \times 4 \) slots and 8 items within.
Table 4  The results on the Ising machine w/o and w/ interpretation of Ising solutions.

| m   | n       | #spins | Ours w/o interpretation | Ours w/ interpretation |
|-----|---------|--------|--------------------------|------------------------|
|     |         |        | TWWL (best)   | TWWL (avg.) | Prob. [%] (slot-placement) | Prob. [%] (additional) | TWWL (best) | TWWL (avg.) | Prob. [%] (additional) |
| 5   | 3 × 3   | 45     | 69           | 70.8        | 99.0                  | 100.0                 | 69           | 71.3         | 100.0                 |
| 7   | 3 × 3   | 63     | 127          | 131.2       | 20.0                  | 100.0                 | 127          | 131.8        | 95.0                  |
| 9   | 3 × 3   | 81     | 277          | 295.5       | 73.0                  | 100.0                 | 277          | 294.8        | 98.0                  |
| 8   | 4 × 4   | 128    | 188          | 195.1       | 75.0                  | 100.0                 | 188          | 195.7        | 86.0                  |
| 12  | 4 × 4   | 192    | 637          | 678.3       | 97.0                  | 100.0                 | 637          | 679.2        | 99.0                  |
| 16  | 4 × 4   | 256    | 1316         | 1365.9      | 49.0                  | 100.0                 | 1316         | 1381.4       | 96.0                  |
| 12  | 5 × 5   | 300    | 631          | 683.7       | 100.0                 | 100.0                 | 631          | 683.7        | 100.0                 |
| 18  | 5 × 5   | 450    | 1610         | 1684.2      | 98.0                  | 100.0                 | 1610         | 1685.2       | 99.0                  |
| 25  | 5 × 5   | 625    | 3919         | 4045.3      | 65.0                  | 100.0                 | 3919         | 4045.4       | 84.0                  |
| 18  | 6 × 6   | 648    | 1593         | 1652.1      | 87.0                  | 100.0                 | 1604         | 1655.2       | 100.0                 |
| 27  | 6 × 6   | 972    | 4700         | 4804.4      | 83.0                  | 100.0                 | 4698         | 4819.7       | 98.0                  |

5 hours. For problems more than 4 × 4 slots and 8 items, we cannot obtain a solution within 5 hours in the brute-force search algorithm. As in Table 3, the TWWL values obtained by the Ising machine are the same as the brute-force search method, up to 4 × 4 slots and 8 items.

Table 4 summarizes the results on the Ising machine without/with performing the interpretation method described in List 1. In the former case, the obtained results do not always satisfy the slot-placement constraint. In the latter case, the final solution always satisfies the slot-placement constraint and can be transformed to the solution of the original problem.

For our proposed QUBO model without and with interpretation method (“Ours w/o interpretation” and “Ours w/ interpretation”, in short, respectively), “TWWL (best)” and “TWWL (avg.)” show the minimum and average TWWL that satisfy the additional constraints over 100 trials, respectively. “Prob. (slot-placement) [%]” shows the probability that “Ours w/o interpretation” can obtain a solution satisfying the slot-placement constraint. “Prob. (additional) [%]” shows the probability that “Ours w/ interpretation” and “Ours w/ interpretation” can obtain a solution satisfying the additional constraint.

As in Table 4, we can obtain a feasible solution in all the cases with a finite probability even if we do not apply the interpretation method to it. The experimental results show that the slot-placement constraint may be satisfied without performing the interpretation method, but it is expected that the probability of satisfying the slot-placement constraint decreases as the problem size increases. If it occurs, the interpretation method greatly contributes to solving the constrained slot-placement problem.

The upper/lower panel of Fig. 4 shows the histogram of TWWL for m = 27 and n = 6 × 6. obtained by the Ising machine without/with performing the interpretation method. The vertical axis represents the number of solutions satisfying the slot-placement constraint. Also, Fig. 5 shows the merged histogram with the upper/lower panel of Fig. 4. The blue bars are obtained by applying the interpretation method in addition to the original results. As can be seen in Fig. 5, the solutions of w/ interpretation do not always lead to larger TWWL.

Note that, since the original infeasible solution before applying the interpretation method does not satisfy the slot-placement constraint, we cannot obtain its TWWL and hence we cannot evaluate how much TWWL is changed after the interpretation method is applied.
5.3 Execution Time Comparison

Now we compare the estimated execution time of the Ising machine based on our proposed QUBO model and SA (See Table 5).

**Ours w/ interpretation:** The execution time of “Ours w/ interpretation” is estimated as follows: The “Annealing time” column shows the average time to execute one annealing for 100 trials. The “Interpretation time” column shows the average CPU time to execute the interpretation method for 100 trials. “Total time” column shows the sum of “Annealing time” and “Interpretation time”.

**SA(avg.):** The execution time of “SA(avg.)” is estimated as follows: In “SA(avg.)”, we perform 100 trials. In the i-th trial, let $L(j)$ be the TWWL in the j-th step. Every time $L(j) \leq \text{TWWL (avg.)}$ is satisfied, we increment the count of satisfying $L(j) \leq \text{TWWL (avg.)}$ by one in every inner loop. Here, TWWL (avg.) shows TWWL obtained by our method w/ interpretation. In order to obtain a converged result, we terminate the loop when the half of the inner loop steps satisfies $L(j) \leq \text{TWWL (avg.)}$ and record the elapsed CPU time at this point. If the half of the inner loop steps does not satisfy $L(j) \leq \text{TWWL (avg.)}$, the execution time of “SA(avg.)” shows the CPU time for 100,000 steps. We average the CPU time obtained above, which is shown in the “SA(avg.)” column. In a similar way, the execution time of “SA(best)” can be calculated from “TWWL(best)”. The estimated execution time of “Ours w/ interpretation”, “SA(avg.)”, and “SA(best)” is shown in Fig. 6, where the execution time of “Ours w/ interpretation” is the sum of “Annealing time” and “Interpretation time”. As can be seen in Table 5 and Fig. 6, when the simple SA parameter in Table 2 is given, “Ours w/ interpretation” can solve the constrained slot-placement problem 18.5 times faster compared to “SA(avg.)” and 40 times faster compared to “SA(best)” as the problem size increases.

### Table 5
Execution time of the proposed method and the SA method.

| $m$ | $n$ | SA(avg) | SA(best) | Ours w/ interpretation |
|-----|-----|---------|----------|------------------------|
| 5   | 3 x 3 | $1.0 \times 10^{-2}$ | $1.1 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $1.7 \times 10^{-5}$ |
| 7   | 3 x 3 | $1.3 \times 10^{-2}$ | $1.5 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $1.7 \times 10^{-5}$ |
| 9   | 3 x 3 | $1.1 \times 10^{-2}$ | $1.8 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $1.1 \times 10^{-5}$ |
| 8   | 4 x 4 | $2.1 \times 10^{-2}$ | $2.3 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $4.0 \times 10^{-5}$ |
| 12  | 4 x 4 | $2.3 \times 10^{-2}$ | $3.0 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $3.8 \times 10^{-5}$ |
| 16  | 4 x 4 | $2.6 \times 10^{-2}$ | $3.2 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $2.6 \times 10^{-5}$ |
| 12  | 5 x 5 | $2.2 \times 10^{-2}$ | $4.1 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $0.0$ |
| 18  | 5 x 5 | $3.1 \times 10^{-2}$ | $5.4 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $5.8 \times 10^{-5}$ |
| 25  | 5 x 5 | $6.1 \times 10^{-2}$ | $8.3 \times 10^{-2}$ | $4.8 \times 10^{-3}$ | $4.8 \times 10^{-3}$ | $3.7 \times 10^{-5}$ |
| 18  | 6 x 6 | $7.8 \times 10^{-2}$ | $1.1 \times 10^{-1}$ | $4.8 \times 10^{-3}$ | $4.8 \times 10^{-3}$ | $4.5 \times 10^{-5}$ |
| 27  | 6 x 6 | $9.8 \times 10^{-2}$ | $2.1 \times 10^{-1}$ | $5.3 \times 10^{-3}$ | $5.1 \times 10^{-3}$ | $2.1 \times 10^{-4}$ |

5.4 SA Parameter Optimization

In this subsection, we optimize SA parameters and perform a re-experiment for a fairer comparison in execution time between the SA method and an actual Ising machine.

5.4.1 Parameters Setting for Simulated Annealing

It is important to investigate appropriate scheduling of the SA method. In the SA parameter setting, it is considered that the initial temperature and cooling rate must be appropriately set so that the final cost of energy function is minimum when getting to a sufficiently low temperature. Now we investigate how much the initial temperature of the SA method of Sect. 5.1.2 affects the final solutions. We fix the final temperature at 0.1 and change the initial temperature and the cooling rate. Then we record the final state of energy function. Figures 7 shows the average $TWWL_{\text{final}}$ when the initial temperature $T_{\text{start}}$ is changed and Fig. 8 shows the enlarged one when the problem size is $n = 3 \times 3$.

In these figures, “Average $TWWL_{\text{final}}$” is the average final state of TWWL of 100,000 iterations for 100 SA trials. “Average $TWWL_{\text{final}}$” includes the error ranges. In these results, the parameters used is summarized in Table 6. As in Figs. 7 and 8, we can obtain sufficiently converged solutions even if the initial temperature is 10. We do not consider that setting the initial temperature to 10 and the other parameters...
to those shown in Table 6 is the fully optimum but we believe that, as in Figs. 7 and 8, they give a reasonably good parameter setting. Particularly, starting the SA process from the low temperature speeds up the entire SA process and hence we can do a fairer comparison in execution time between the SA method and an actual Ising machine.

Based on these results, we set the initial temperature to be 10 and other parameters are set according to Table 6 in the SA method.

5.4.2 Execution Time Comparison

Now we compare the execution time of the Ising machine based on our proposed QUBO model and the SA method using the parameters of Sect. 5.4.1. The parameters in the Ising machine used in the proposed method are the same as Sect. 5.1.1. In the same way as Sect. 5.3, the execution times of “Ours w/ interpretation”, “SA(avg.)”, and “SA(best)” are shown in Table 7 and Fig. 9, where the execution time of “Ours w/ interpretation” is the sum of “Annealing time” and “Interpretation time”.

As can be seen in Table 7 and Fig. 9, “Ours w/ interpretation” can still solve the constrained slot-placement problem 8 times faster compared to “SA(avg.)” and 8.5 times faster compared to “SA(best)”.

5.5 Discussions

As in Sect. 5.1.2, the SA method used takes the cost function of Eq. (2) and Eq. (18). Equation (2) is equivalent to Eq. (7), which is re-expressed by using binary variables. Also, Eq. (18) is equivalent to Eq. (13), which is re-expressed by using binary variables.

The difference in the cost function between the SA method and our method is that our method has Eq. (8) and Eq. (11) in the QUBO model. This is because of the following reason:

**QUBO model**: Since the QUBO model has to represent the slot-placement constraint as an energy function, we introduce Eq. (8) and Eq. (11). These two terms are necessary for an actual Ising machine to solve the constrained slot-placement problem.

**SA method**: As in Sect. 5.1.2, the SA method always satisfies the slot-placement constraint. This is because we start the SA process satisfying the slot-placement

---

**Fig. 7** Problem size dependence of average $\text{TWWL}_{\text{final}}$.

**Fig. 8** Problem size dependence of average $\text{TWWL}_{\text{final}} (n = 3 \times 3)$.

**Table 6** Set up in the simulated annealing for a fairer comparison.

| Parameter          | Value          |
|--------------------|----------------|
| Initial solution   | Random         |
| Initial temperature| 10, 100, 1,000 |
| Final temperature  | 0.1            |
| Inner loop         | 1,000          |
| Outer loop         | 100            |
| Cooling rate       | Final temperature | Inner loop:Outer loop |

**Table 7** Execution time of the proposed method and the optimized SA method.

| $m$ | $n$ | $\text{SA(avg)}$ | $\text{SA(best)}$ | $\text{Ours w/ interpretation}$ |
|-----|-----|------------------|-------------------|---------------------------------|
|     |     | Total time | Annealing time | Interpretation time | Interpretation time | Interpretation time |
| 5   | $3 \times 3$ | $2.1 \times 10^{-2}$ | $2.2 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $1.7 \times 10^{-3}$ |
| 7   | $3 \times 3$ | $2.6 \times 10^{-2}$ | $3.0 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $1.7 \times 10^{-3}$ |
| 9   | $3 \times 3$ | $3.1 \times 10^{-2}$ | $3.2 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $1.1 \times 10^{-3}$ |
| 8   | $4 \times 4$ | $3.1 \times 10^{-2}$ | $3.5 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $4.0 \times 10^{-3}$ |
| 12  | $4 \times 4$ | $3.3 \times 10^{-2}$ | $3.6 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $3.8 \times 10^{-3}$ |
| 16  | $4 \times 4$ | $1.2 \times 10^{-2}$ | $1.8 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $2.6 \times 10^{-3}$ |
| 12  | $5 \times 5$ | $1.7 \times 10^{-2}$ | $2.8 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $2.6 \times 10^{-3}$ |
| 18  | $5 \times 5$ | $1.5 \times 10^{-2}$ | $1.8 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $5.8 \times 10^{-3}$ |
| 25  | $5 \times 5$ | $1.4 \times 10^{-2}$ | $1.9 \times 10^{-2}$ | $4.2 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $3.7 \times 10^{-3}$ |
| 18  | $6 \times 6$ | $1.7 \times 10^{-2}$ | $2.1 \times 10^{-2}$ | $4.8 \times 10^{-3}$ | $4.8 \times 10^{-3}$ | $4.5 \times 10^{-3}$ |
| 27  | $6 \times 6$ | $1.5 \times 10^{-2}$ | $1.8 \times 10^{-2}$ | $5.3 \times 10^{-3}$ | $5.1 \times 10^{-3}$ | $2.1 \times 10^{-4}$ |
constraint and exchange the items so that the slot-placement constraint is satisfied.

In this sense, we believe that the cost function of the SA method and our method is essentially the same and that the comparison must be fair from the viewpoint of the cost function.

On the other hand, the scheduling can be further optimized, particularly for the SA method. As in Sect. 5.5 above, we optimize the scheduling scheme in the SA method and demonstrate that still our proposed method is superior in execution time.

Overall, we believe that we have done a fair comparison from the viewpoints of cost function and scheduling.

Note that, we can use completely the same cost function (energy function) of Eq. (14) even in the SA method. In this case, the SA method searches for the ground state of the QUBO model by software. This approach definitely requires much execution time compared to using the actual Ising machine. For example, in one application program, it is reported that the software simulation requires two orders of magnitude more execution time than the actual Ising machine [1].

6. Conclusions

In this paper, we have proposed a QUBO model mapping to solve the constrained slot-placement problem and an interpretation method for the obtained Ising solutions so as to satisfy the slot-placement constraint. We found that Fujitsu Digital Annealer is faster than the simulated annealing that we implemented.

In the future, we will propose an efficient method to reduce the number of spins by dividing the constraint slot-placement problem and solve larger scale problems.

References

[1] M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura, and H.G. Katzgraber, “Physics-Inspired Optimization for Quadratic Unconstrained Problems Using a Digital Annealer,” frontiers in Physics, April 2019.

[2] H. Goto, K. Tatsumura, and A.R. Dixon, “Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems,” Science Advances, vol.5, no.4, April 2019.

[3] M. Maezawa, G. Fuji, 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,” Journal of the Physical Society of Japan, vol.88, no.6, Feb. 2019.

[4] S. Matsubara, H. Tamura, M. Takatsu, D. Yoo, B. Vatankhahghadin, H. Yamasaki, T. Miyazawa, S. Tsukamoto, Y. Watanabe, K. Takemoto, and A. Sheikhholeslami, “Ising-model optimizer with parallel-trial bit-sieve engine,” vol.772, pp.432–438, 2018.

[5] M. Maezawa, K. Imafuku, M. Hidaka, H. Koike, and S. Kawabata, “Design of quantum annealing machine for prime factoring,” Proc. 2017 16th International Superconductive Electronics Conference (ISEC), pp.1–3, June 2017.

[6] S. Tsukamoto, M. Takatsu, S. Matsubara, and H. Tamura, “An accelerator architecture for combinatorial optimization problems,” vol.53, no.5, pp.8–13, Sept. 2017.

[7] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuo, “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.

[8] 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. Kawai, I. Inoue, S. Usunoki, and H. Takesue, “A coherent Ising machine for 2000-node optimization problems,” Science, vol.354, no.6312, pp.603–606, Oct. 2016.

[9] P.L. McMahon, A. Marandi, Y. Haribara, R. Hamerly, C. Langrock, S. Tamate, T. Inagaki, H. Takesue, S. Usunoki, 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, Oct. 2016.

[10] C. Yoshimura, M. Hayashi, T. Okuyama, and M. Yamaoka, “FPGA-based Annealing Processor for Ising Model,” Proc. Fourth International Symposium on Computing and Networking (CANDAR 2016), pp.436–442, Nov. 2016.

[11] P.I. Bunyk, E.M. Hoskinson, M.W. Johnson, E. Tolkacheva, F. Altomare, A.J. Berkley, R. Harris, J.P. Hilton, T. Lanting, A.J. Przybysz, and J. Whittaker, “Architectural Considerations in the Design of a Superconducting Quantum Annealing Processor,” IEEE Trans. Appl. Supercond., vol.24, no.4, pp.1–10, Aug. 2014.

[12] M.W. Johnson, M.H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A.J. Berkley, J. Johansson, P. Bunyk, E.M. Chapple, C. Enderud, J.P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M.C. Thom, E. Tolkacheva, C.J. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose, “Quantum annealing with manufactured spins,” Nature, vol.473, no.7346, pp.194–198, May 2011.

[13] S. Tanaka, R. Tamura, and B.K. Chakrabarti, Quantum Spin Glasses, Annealing and Computation, 1st ed., Cambridge University Press, New York, NY, USA, 2017.

[14] A. Lucas, “Ising formulations of many NP problems,” Frontiers in Physics, vol.2, pp.1–15, 2014.

[15] K. Tanahashi, S. Takeyamagi, T. Motoshashi, and S. Tanaka, “Application of Ising Machines and a Software Development for Ising Machines,” Journal of the Physical Society of Japan, vol.88, no.6, June 2019.

[16] A. Hemani and A. Postula, “Cell placement by self-organisation,” Neural Networks, vol.3, no.4, pp.377–383, 1990.

[17] A. Lucas, “Ising formulations of many NP problems,” Frontiers in Physics, vol.2, pp.1–15, 2014.

[18] K. Tanahashi, S. Takeyamagi, T. Motoshashi, and S. Tanaka, “Application of Ising Machines and a Software Development for Ising Machines,” Journal of the Physical Society of Japan, vol.88, no.6, June 2019.

[19] A. Hemani and A. Postula, “Cell placement by self-organisation,” Neural Networks, vol.3, no.4, pp.377–383, 1990.
different approaches to solve the job assignment problem,” Proc. International Conference on Trends in Electronics and Informatics (ICETI), pp.129–134, May 2017.

[20] F. Xiongfeng and S. Qiang, “An applied case of quadratic assignment problem in hospital department layout,” Proc. International Conference on Service Systems and Service Management (ICSSSM), June 2015.

[21] S. Srinivasan, V. Kamakoti, and A. Bhattacharya, “Towards quick solutions for generalized placement problem,” Proc. 2011 International Symposium on Electronic System Design (ISED), pp.19–21, Dec. 2011.

[22] M. Rajeswari, S. Jaiganesh, P. Sujatha, T. Vengattaraman, and P. Dhavachelvan, “A study and scrutiny of diverse optimization algorithm to solve multi-objective quadratic assignment problem,” IEICE Technical Report, COMP2016-11, pp.21–22, Oct. 2016.

[23] Qela1998, “The Quadratic Assignment Problem: Theory and Algorithms,” Kluwer Academic Publishers, 1998.

[24] P. Merz and B. Freisleben, “A comparison of memetic algorithms, tabu search, and ant colonies for the quadratic assignment problem,” Proc. Congress of Evolutionary Computation, pp.2063–2070, 1999.

[25] N. Nishimura, K. Tanahashi, K. Suganuma, M.J. Miyama, and M. Ohzeki, “Item Listing Optimization for E-Commerce Websites Based on Diversity,” frontiers in Computer Science, July 2019.

[26] K. Terada, “Floorplan-aware high-level synthesis algorithms and their acceleration by Ising computations,” Waseda University Ph. D. thesis, pp.94–108, 2018.

Sho Kanamaru received the B. Eng. and M. Eng degrees from Waseda University in 2018 and 2020, respectively, all in Communications and Computer Engineering. His research interests are combinatorial optimization, and Ising computer.

Kazushi Kawamura received the B. Eng., M. Eng. and Dr. Eng. degrees from Waseda University in 2012, 2013 and 2016, respectively, all in computer science. He is presently an adjunct researcher, in Green Computing Systems Research Organization, Waseda University and an assistant professor in the Institute of Innovative Research, Tokyo Institute of Technology. His research interests include Ising computing, FPGA design, high-level synthesis and secure computing. He is a member of IEEE and the Institute of Electronics, Information and Communication Engineers.

Shu Tanaka received the B. Sci. degree from Tokyo Institute of Technology in 2003 and the M. Sci. and Dr. Sci. degrees from The University of Tokyo in 2005 and 2008, respectively. He is presently a guest associate profesor in Green Computing Systems Research Organization, Waseda University and an associate professor in the Department of Applied Physics and Physico-Informatics, Keio University. His research interests are quantum annealing, Ising machine, statistical mechanics, and materials science. He is a member of JPS.

Yoshinori Tomita received the B. Eng. degree, M. Eng. degree and finished Dr. program rom Tokyo Institute of Technology in 1993, 1995 and 1999. He is presently working on application of Fujitsu Digital annealer to real world combinatorial optimization problems at Fujitsu Laboratories LTD. His research interests are optimization problems and simulation.

Nozomu Togawa 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 presently a Professor in the Department of Computer Science and Communications Engineering, Waseda University. His research interests are VLSI design, graph theory, and computational geometry. He is a member of IEEE and IPSJ.