IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

DOI:10.1587/transfun.2024VLP0008

Publicized:2024/10/08

This advance publication article will be replaced by the finalized version after proofreading.

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Hybrid Iterative Annealing Method Using a Quantum Annealer and a Classical Computer and its Evaluation

Keisuke FUKADA[†a)], **Tatsuhiko SHIRAI**[†], *Nonmembers*, *and* **Nozomu TOGAWA**[†b)], *Senior Member*

SUMMARY A combinatorial optimization problem is the problem of minimizing the energy function among many combinations of variables, which is often difficult to solve with conventional classical computers. Recently, Ising machines, including quantum annealers, have gained attention as a promising architecture for efficiently solving combinatorial optimization problems. Among various methods for solving such problems using Ising machines, one prominent approach is the three-stage annealing method. The approach effectively solves a combinatorial optimization problem, utilizing an initial solution, but it performs the annealing process only once. Repeating the annealing process several times may enhance the solution more efficiently. In this paper, we propose a novel hybrid iterative annealing method that consists of an initial process using a classical computer, an annealing process using a quantum annealer, and a correction process/selection process using a classical computer. The proposed method repeats the annealing process and the correction process/selection process until the solution is sufficiently converged. In the experimental evaluations through the three types of typical combinatorial optimization problems, the proposed method shows improvements by up to 54.0% compared to the three-stage annealing method.

key words: *Quantum annealer, Ising machine, combinatorial optimization problem, hybrid iterative annealing*

1. Introduction

A combinatorial optimization problem is the problem of finding an optimal solution from a finite set of solutions that optimizes an energy function while satisfying the problem's constraints. Among these problems, NP-hard problems [1]–[5] are difficult to solve using classical computers, which are often seen in consumer applications. Over the last few years, several Ising machines, including quantum annealers such as D-Wave 2000Q and D-Wave Advantage [6], [7] have been developed as novel machines for efficiently solving combinatorial optimization problems [8]–[12]. Ising machines are non-von Neuman-type hardware machines to search for the Ising model's minimum energy state, called *ground state*. To solve a combinatorial optimization problem using an Ising machine, we map its optimal solution to the ground state of the Ising model [13], [14] or its equivalent Quadratic Unconstrained Binary Optimization (QUBO) model [15]. Nowadays, various combinatorial optimization problems including consumer applications have been solved by Ising machines [16]–[20].

One of the efficient methods for solving combinatorial optimization problems using Ising machines is the three-stage annealing method proposed in [21]. In [21], by giving a good initial solution to an Ising machine, a quasi-optimal solution is obtained using an Ising machine. However, this approach performs an annealing process only once. If we can develop a framework to perform the annealing process iteratively, we can obtain a further improved solution to a combinatorial optimization problem.

In this paper, we propose a hybrid iterative annealing method as the more effective method hybridly using a quantum annealer and a classical computer.[1] Hereafter, we adopt a quantum annealer as an Ising machine although our method is generally applicable to others. The hybrid iterative annealing method consists of the following four processes: (a) an initial process, (b) an annealing process, (c) a correction process, and (d) a selection process. The processes (a), (c), and (d) are performed using a classical computer, whereas the process (b) is performed by a quantum annealer. By repeating the processes (b)–(d), we can finally obtain a solution to the combinatorial optimization problem.

The proposed method works as follows: In the initial process, we map a combinatorial optimization problem to the Ising model or QUBO model, solve it by a classical algorithm, and input the obtained solution to a quantum annealer as the initial solution. In the annealing process, a quantum annealer performs annealing and searches for improved solutions from the input initial solution. The correction process corrects the solutions to satisfy the constraints of the combinatorial optimization problem. In the selection process, we select one solution that minimizes the energy function mapping the combinatorial optimization problem to an Ising model or QUBO model using a classical computer, and this solution is input as the initial solution for the annealing process of the next iteration.

The contributions of this paper are summarized as follows.

1. We propose a novel hybrid iterative annealing method that hybridly utilizes a quantum annealer and a classical computer for efficiently solving a combinatorial optimization problem.

[†]The author is with the Department of Computer Science and Communications Engineering, Waseda University, Tokyo, 169-8555 Japan.

a) E-mail: keisuke.fukada@togawa.cs.waseda.ac.jp

b) E-mail: ntogawa@waseda.jp

[1]The preliminary version of this paper appeared in [22]. In this paper, we have described the backgrounds and related works in detail in Section 3. We have deepened the discussions and added examples in Section 3.3. We have added the experimental results on MaxCut problems and confirmed the effectiveness of the proposed method in Section 4.4.

2. We evaluate the proposed method for three typical combinatorial optimization problems. The experimental evaluations show that the proposed method improves the cost of the quadratic assignment problem by up to 28.7%, the traveling salesman problem by up to 54.0%, and the MaxCut problems by up to 2.52%. Also, the proposed method reaches an optimal solution in several traveling salesman problem instances.

The rest of this paper is organized as follows: Section 2 describes quantum annealing algorithms on a quantum annealer; Section 3 proposes the proposed method hybridly utilizing a quantum annealer and a classical computer to efficiently solve a combinatorial optimization problem; Section 4 applies the proposed method to three combinatorial optimization problems and demonstrates the effectiveness of the method; Section 5 summarizes this paper.

2. Quantum annealing

There are various types of Ising machines, and various annealing algorithms have been developed. In this section, we will mainly focus on quantum annealing [23] performed on a quantum annealer that adjusts the strength of the transverse magnetic field acting on the qubits to find the ground state of the Ising model through quantum effects [24].

Quantum annealing is a typical algorithm performed on a quantum annealer that can solve combinatorial optimization problems. D-Wave has been developing several quantum annealers, such as D-Wave 2000Q and D-Wave Advantage [6], [7] (hereinafter, they are called D-Wave machines). The Hamiltonian of the D-Wave machines is given as

$$\hat{\mathcal{H}} = -\frac{A(s)}{2}\left(\sum_{i\in V}\hat{\sigma}_i^x\right) + \frac{B(s)}{2}\left(\sum_{i\in V}h_i\hat{\sigma}_i^z + \sum_{(i,j)\in E}J_{i,j}\hat{\sigma}_i^z\hat{\sigma}_j^z\right), \quad (1)$$

where $\hat{\sigma}_i^{x,z}$ are the Pauli operators acting on the i-th qubit, respectively, $A(s)$ and $B(s)$ are the weights of the term, V and E are the sets of vertices and edges of the chimera or pegasus graph of the D-Wave machine, h_i is the external magnetic field acting on the i-th qubit, and $J_{i,j}$ is the interaction coefficient between the i-th and j-th qubits. $A(s)$ $(0 \le s \le 1)$ is a monotonically decreasing function and $B(s)$ is a monotonically increasing function. The first term on the right-hand side of Eq. (1) is the transverse magnetic field term for disturbing the qubits, and the second term is the optimization term.

The expectation value of the second term in Eq. (1) in the computational basis (i.e., z-basis) gives the Ising model form. As in [16], a combinatorial optimization problem can be mapped onto this form, represented by:

$$\mathcal{H}_{\text{comb}} = \sum_{i\in V}h_i\sigma_i + \sum_{(i,j)\in E}J_{i,j}\sigma_i\sigma_j, \quad (2)$$

where σ_i is a spin variable taking either (-1) or $(+1)$. A combination of σ_i minimizing $\mathcal{H}_{\text{comb}}$ gives an optimal solution to the combinatorial optimization problem, and the optimal solution corresponds to the ground state of the second term in (1). $\mathcal{H}_{\text{comb}}$ is also called an energy function.

The Ising model can be transformed into the QUBO model as follows:

$$\mathcal{H}'_{\text{comb}} = \sum_{i\in V}a_ix_i + \sum_{(i,j)\in E}b_{i,j}x_ix_j, \quad (3)$$

where x_i is a binary variable taking either 0 or $(+1)$, a_i is the external magnetic field coefficient acting on the binary variable x_i, and $b_{i,j}$ is the interaction coefficient between the two binary variables x_i and x_j. The binary variable x_i can be transformed from the spin variable σ_i by $x_i = (\sigma_i + 1)/2$.

When performing quantum annealing using a quantum annealer, we have two options: forward annealing and reverse annealing [25].

Forward annealing is a standard form of quantum annealing that starts with $s = 0$ at $t = 0$. Initially, the qubits are in the ground state of the first term in Eq. (1). The problem's solution space is globally searched by linearly increasing s until $s = 1$ at the end of time. If the annealing time is sufficiently long, the quantum state converges to the ground state of the second term in Eq. (1) [26]. The spin configuration is obtained by measuring the quantum state in the z-basis.

On the other hand, reverse annealing starts with $s = 1$ at $t = 0$, setting the initial state of the qubit as an eigenstate of the second term in Eq. (1) in advance. After linearly decreasing s up to s_{min}, which means applying a certain amount of transverse magnetic field, the process linearly increases s until $s = 1$ at the end of time (See the discussion in Section IV and Fig. 3 in the detailed setting of s in the reverse annealing). Reverse annealing controls the size of the search space around the initial state by the value of s_{min}. When we input an initial solution to a quantum annealer, the reverse annealing can be effectively used.

3. Hybrid iterative annealing

Ising machines have been designed to search for the solution that minimizes the energy function expressed in the Ising model or QUBO model. However, performing annealing only once with an Ising machine is not practically sufficient to find the optimal solution to complex combinatorial optimization problems. Hence, various methods using Ising machines to solve a combinatorial optimization problem have been studied.

Ref. [19] proposed that after a solution is obtained by annealing with an Ising machine, the solution is input into the Ising machine as an initial solution, and annealing is performed again. Ref. [19] evaluated the method with nurse scheduling problems and reported that the results are better than those obtained when annealing is performed only once.

Ref. [27] proposed a hybrid method of an Ising machine and a classical computer that after a solution is obtained by greedy search, which is one of the typical heuristic methods,

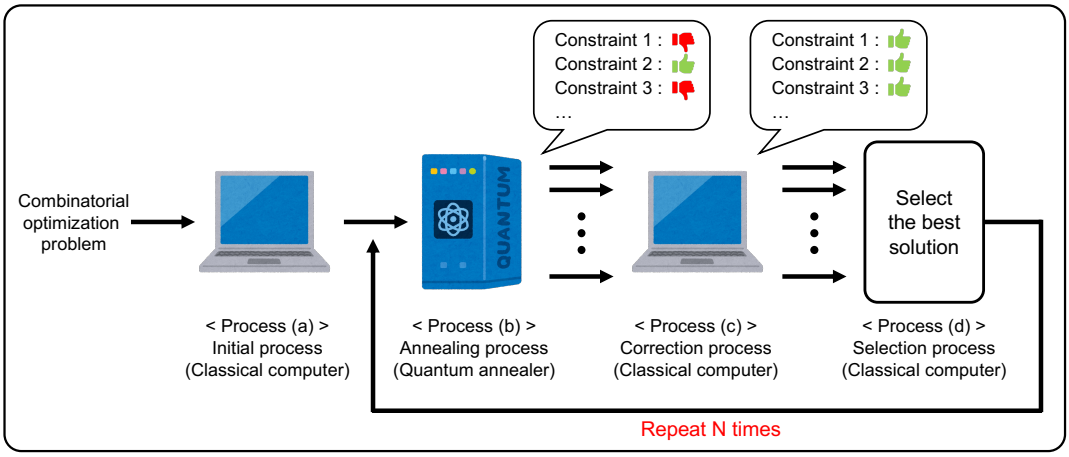


Fig. 1: The flow of the proposed hybrid iterative annealing method.

the solution is input into an Ising machine as an initial solution and annealing is performed. Ref. [27] evaluated the method with job shop scheduling problems and reported that the results are better than those obtained when annealing is performed only once.

Ref. [28] studied two methods: one is to input a randomly selected solution as an initial solution into an Ising machine and perform annealing, and the other is to input a solution obtained by greedy search into an Ising machine and perform annealing. Ref. [28] evaluated two methods with combinatorial optimization problems of wireless networks and reported that the method of inputting a randomly selected solution as an initial solution into the Ising machine is less accurate than the method of performing annealing only once, while the method of inputting a solution obtained by greedy search into the Ising machine show more improvement than the method of performing annealing only once.

Ref. [21] proposed the three-stage annealing method as one of the efficient methods for solving combinatorial optimization problems using Ising machines. The three-stage annealing method consists of three processes: an initial process using a classical computer, an annealing process using an Ising machine, and a correction process using a classical computer. In Ref. [21], an initial solution is generated using a classical computer, and it is input to the annealing process. Then, the annealing process improves the solution. Finally, the correction process corrects the obtained solution so that it satisfies the constraint of the original combinatorial optimization problem. Experimental evaluations using a slot-placement problem show that the solution is significantly improved compared to the approach without giving an initial solution. These results indicate that if we input a good initial solution to an Ising machine and perform an annealing process repeatedly, the solution is expected to be further improved.

On the basis of the above discussion, we propose a hybrid iterative annealing method as the more effective method hybridly using a quantum annealer and a classical computer. Fig. 1 shows the flow of the proposed hybrid iterative annealing method. The hybrid iterative annealing method consists of the following four processes: (a) an initial process, (b) an annealing process, (c) a correction process, and (d) a selection process. The processes (a), (c), and (d) are performed using a classical computer, whereas the process (b) is performed by a quantum annealer. By repeating the processes (b)–(d), we can finally obtain a solution to the combinatorial optimization problem. In the following, we propose the four processes (a)–(d).

Note that, the main difference between the proposed method and [21] in the processes (a)–(d) below is summarized as follows: First, the target of [21] is the slot-placement problem and the process (a) and the process (c), i.e., the initial process and the correction process, are designed for the slot-placement problem. On the other hand, in the proposed method, the initial process is simply designed for a general combinatorial optimization problem, and the correction process is designed for a general two-way 1-hot constraint. In that sense, the proposed method can be applied to more various application problems. Second, a semiconductor-based Ising machine is used in [21], but the proposed method employs a quantum annealer. A quantum annealer cannot directly accept an initial solution but requires a reverse annealing for an initial solution. Thus, the process (b), i.e., the annealing process, is different from [21], and we carefully set up an annealing schedule as discussed in Section 4.1. The process (d) is trivial and almost the same as [21].

3.1 Initial process

The initial process aims to solve a combinatorial optimization problem using a classical computer and provide a good solution to be input as the initial solution for a quantum annealer. The initial process proposed in this paper employs a random method, which is a simple but fast classical algorithm for solving a combinatorial optimization problem.

In the random method, we first generate a solution x_{min} randomly that satisfies the combinatorial optimization problem's constraints. Next, we randomly select a solution x_{new}

Algorithm 1 Random method

```
x_min ← a random feasible solution;
for 1 ≤ i ≤ R do
    x_new ← the feasible one from the neighborhood of x_min;
    if Energy(x_new) < Energy(x_min) then
        x_min ← x_new;
return x_min
```

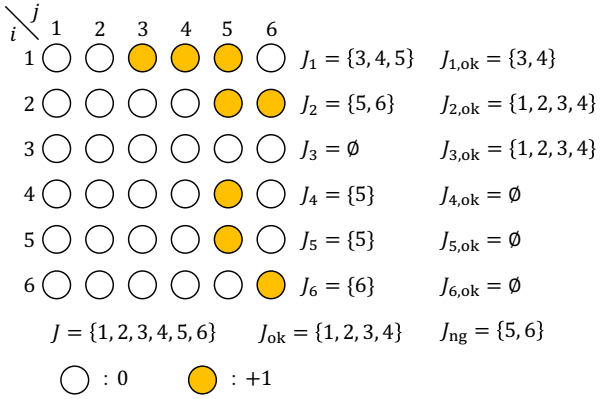


Fig. 2: An example of mathematical expression in the correction process.

from the neighborhood of $x_{\min}$, satisfying the problem's constraints. If the energy value of x_{new} is smaller than that of $x_{\min}$, we update $x_{\min}$ to x_{new}. We repeat this process for R times. After that, we obtain $x_{\min}$ as the final solution of the initial process and input it to a quantum annealer.

The algorithm for the random method is presented in Algorithm 1.

3.2 Annealing process

In the annealing process, we input the energy function of the combinatorial optimization problem and the initial solution provided through the initial process or the selection process to a quantum annealer and perform the reverse annealing, as described in Section 2.

Note that reverse annealing has various scheduling schemes. In the proposed method, the initial solution is kept in the quantum annealer as long as possible (see Section 4.1 in detail).

3.3 Correction process

In the annealing process, we may obtain a solution that does not satisfy the given problem's constraints. Consequently, post-processing with a classical computer often becomes necessary to correct these solutions to satisfy the constraints. This process is called the *correction process*.

A combinatorial optimization problem generally has various types of constraints. One of the typical ones is the two-way 1-hot constraint, which is seen in the quadratic assignment problem (QAP) and the traveling salesman problem (TSP).[1] The two-way 1-hot constraint is defined as follows: Assume that a solution to a combinatorial optimization problem is given by an $n \times n$ matrix, each element of which takes either (+1) or 0.[2] At that time, only one variable in each row must have a value of (+1) and the others in the same row must have a value of 0. Further, only one variable in each column must have a value of (+1), and the others in the column must have a value of 0.

3.3.1 Definition

In this section, we define the notations for the random correction process. Assume that we have an $n \times n$ matrix $x_M = \{x_{i,j}\}$, each element $x_{i,j}$ of which takes either (+1) or 0. Let i $(1 \le i \le n)$ be the indices in the row direction, and j $(1 \le j \le n)$ be the indices in the column direction. Let $J = \{1, \ldots, j\}$ be a set of indices in the column direction and $J_i \subseteq J$ be a set of indices whose variable is (+1) in the i-th row, i.e.,

$$J \stackrel{\text{def}}{=} \{j\}_{j=1}^{n}, \tag{4}$$

$$J_i \stackrel{\text{def}}{=} \{\, j \mid x_{i,j} = 1\} \quad (1 \le i \le n). \tag{5}$$

Also, we define J_{ng} and J_{ok} as follows:

$$J_{\text{ng}} \stackrel{\text{def}}{=} \left\{ j \mid \exists i \sum_{1 \le k \le n} x_{i,k} = 1 \wedge x_{i,j} = 1 \right\}, \tag{6}$$

$$J_{\text{ok}} \stackrel{\text{def}}{=} J \setminus J_{\text{ng}}. \tag{7}$$

J_{ng} shows the set of column indices where $x_{i,j} = 1$ for the row direction satisfying the 1-hot constraint. The column indices included in J_{ng} should be unselected as much as possible to keep the annealing result.

Example 1: Fig. 2 shows an example of 36 QUBO binary variables obtained by the annealing process, with white circles indicating 0 and orange circles indicating (+1). Since J is defined as the set of whole column numbers j, we obtain $J = \{1, 2, 3, 4, 5, 6\}$. J_1 is the set of column number j whose binary variable is (+1) in the 1st row, and we obtain $J_1 = \{3, 4, 5\}$. Similarly, $J_2 = \{5, 6\}, J_3 = \emptyset, J_4 = \{5\}, J_5 = \{5\}$, and $J_6 = \{6\}$. J_{ng} is the set of j whose binary variable is

[1] The two-way 1-hot constraints are often seen in assignment problems [29], [30], which is considered to be a most useful application in quantum annealers. Thus, we particularly pick up the two-way 1-hot constraints in this paper. Extending the proposed method to deal with other constraints is one of the important future works.

[2] When we assume the QUBO form in Eq. (3), the solution is given by 0-1 binary variables. Further, if we consider the QAP or TSP, its solution is given by an $n \times n$ matrix as discussed here.

Algorithm 2 Random correction process

1: $x \leftarrow$ solve(QUBO);
2: $x_M \leftarrow$ transform x to n-by-n matrix;
3: **for** $1 \le k \le 2$ **do**
4: Set $J_{\rm ok}$ and $J_{\rm ng}$ from x_M;
5: **while** (Until every row satisfies the 1-hot constraint) **do**
6: $i \leftarrow$ randomly select i-th row which does not satisfy the 1-hot constraint from x_M;
7: Set $J_{i,{\rm ok}}$;
8: $j \leftarrow$ randomly select one from $J_{i,{\rm ok}}$;
9: Only the variable in i-th row and j-th column is set to (+1) and all the other variables in the i-th row are set to 0;
10: Update $J_{\rm ok}$ and $J_{\rm ng}$;
11: $x_M \leftarrow x_M^T$;
return (return x)

(+1) in the rows satisfying the 1-hot constraint for the row direction, and it can be defined as $J_{\rm ng} = \{5, 6\}$ since the 4th row, 5th row, and 6th row satisfy the 1-hot constraint where the 5th column ($j = 5$) and 6th column ($j = 6$) give (+1). $J_{\rm ok}$ is the complement of $J_{\rm ng}$ for J, and thus it is given as $J_{\rm ok} = \{1, 2, 3, 4\}$.

Furthermore, we define $J_{i,{\rm ok}}$ for i-th row, which does not satisfy the 1-hot constraint, as follows.

$$J_{i,{\rm ok}} \stackrel{\rm def}{=} \begin{cases} J_i \cap J_{\rm ok} & \text{if } J_i \cap J_{\rm ok} \neq \emptyset \\ J_{\rm ok} & \text{Otherwise} \end{cases} \quad (1 \le i \le n) \qquad (8)$$

$J_{i,{\rm ok}}$ shows the set of columns which less affects the 1-hot constraint.

Example 2: When $i = 1$, the intersection of $J_1 = \{3, 4, 5\}$ and $J_{\rm ok} = \{1, 2, 3, 4\}$ is not an empty set but $\{3, 4\}$, and thus $J_{1,{\rm ok}}$ is given as $\{3, 4\}$. On the other hand, when $i = 2$, the intersection of $J_2 = \{5, 6\}$ and $J_{\rm ok} = \{1, 2, 3, 4\}$ is an empty set, and hence $J_{2,{\rm ok}}$ is given as $J_{\rm ok} = \{1, 2, 3, 4\}$. Similarly, for $i = 3$, we obtain $J_{3,{\rm ok}} = \{1, 2, 3, 4\}$. When $4 \le i \le 6$, the i-th row satisfies the 1-hot constraint, and thus $J_{i,{\rm ok}}$ is an empty set.

3.3.2 Random correction process algorithm

The algorithm for the random correction process is shown in Algorithm 2. First, the solution obtained by the annealing process is converted to an $n \times n$ matrix, x_M (lines 1–2). The following process is repeated until all the 1-hot constraints for the row direction are satisfied, and then the same process is repeated after the $n \times n$ matrix x_M is transposed (lines 3–11). First, $J_{\rm ok}$ and $J_{\rm ng}$ are derived from x_M (line 4). Here, i-th row that does not satisfy the 1-hot constraint for the row direction is randomly selected, and $J_{i,{\rm ok}}$ is obtained from J_i and $J_{\rm ok}$ (lines 6–7). We randomly select j from $J_{i,{\rm ok}}$, and set only the binary variable in i-th row and j-th column to (+1) and set all other binary variables in i-th row to 0 (lines 8–9). Since the above operation increases the number of rows that satisfy the 1-hot constraint for the row direction, we update $J_{\rm ok}$ and $J_{\rm ng}$ (line 10).

In the algorithm, the annealing result is utilized as much as possible, even if it does not satisfy the two-way 1-hot constraint. Focusing on every i-th row, if the i-th row satisfies the 1-hot constraint, we do not flip any variable in the i-th row. If the i-th row has two or more (+1) variables, we select one of them as much as possible to satisfy the 1-hot constraint. Thus, the annealing result generated by the annealing process is well utilized in the correction process. We perform this process in the row direction and in the column direction.

Example 3: Assume that the binary variables in Fig. 2 are obtained from the annealing process. The annealing process outputs a set of 36 binary variables, which are converted to a 6×6 binary variable matrix, denoted as x_M (lines 1–2). We can set $J_{\rm ok} = \{1, 2, 3, 4\}$ and $J_{\rm ng} = \{5, 6\}$ as described in 3.3.1 (line 4). Next, the rows that do not satisfy the 1-hot constraint for the row direction are the 1st, 2nd, and 3rd rows, from which we randomly select one (line 6). Here, we assume $i = 1$. Since $J_1 = \{3, 4, 5\}$ and $J_{\rm ok} = \{1, 2, 3, 4\}$, we can set $J_{1,{\rm ok}} = \{3, 4\}$ (line 7). Then, one element is randomly selected from $J_{1,{\rm ok}} = \{3, 4\}$, and here, we assume $j = 3$ (line 8). Next, only the binary variable in the 1st row and 3rd column, i.e., $i = 1$ and $j = 3$, is set to (+1), and all other binary variables are set to 0 in the 1st row (line 9). From the above operation, $J_{\rm ok}$ and $J_{\rm ng}$ are updated to $J_{\rm ok} = \{1, 2, 4\}$ and $J_{\rm ng} = \{3, 5, 6\}$ (line 10). We continue to perform the same process for the 2nd to 6th rows (lines 5–10). At the end of line 10, all the 1-hot constraints for row directions must be satisfied.

Next, we correct the 1-hot constraint for column direction. First, we transpose x_M (line 11). By transposing x_M, every column becomes a row, and we can newly define $J_{\rm ok}$ and $J_{\rm ng}$ for the rows in the transposed x_M (line 4). Then, we perform the same process of correcting the 1-hot constraints for each row (lines 5–10). Finally, by transposing x_M again, all 1-hot constraints in both row and column directions must be satisfied.

3.4 Selection process

In the selection process, a classical computer selects one solution that gives the smallest energy function mapping a combinatorial optimization problem to an Ising model or QUBO model among a set of multiple solutions that satisfy the constraints obtained by the correction process. The selected solution is input as the initial solution for the annealing process to be performed in the next iteration.

4. Experimental evaluation

In this section, we have done three experiments comparing our proposed method with a baseline method. The baseline method refers to a method in which the processes (a)–(d) are executed once without iteration in the proposed method.

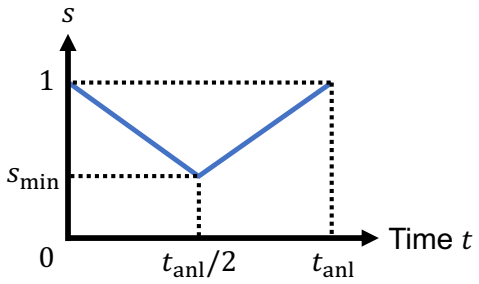


Fig. 3: The schedule of reverse annealing.

The baseline method is similar to [21] but its target of the process (a) is TSP, QAP, and MaxCut problem. The target of the process (c) is TSP and QAP. The process (b) uses a quantum annealer instead of using a semiconductor-based Ising machine.

In this evaluation, we have picked up the quadratic assignment problem (QAP) and the traveling salesman problem (TSP), which have constraints and are difficult to solve with a quantum annealer. In this paper, we focus on the problems with two-way 1-hot constraints of QAP and TSP. We also picked up the MaxCut problem to investigate the impact of the proposed method on problems without constraints.

4.1 Common setup

We employed the D-Wave Advantage quantum annealer (D-Wave machine) in this experiment. In the annealing process, we fixed the execution time of quantum annealing, denoted as t_{anl}, at 2 ms, the maximum allowable time for the D-Wave Advantage [31]. Furthermore, we obtained a set of 100 solutions at every annealing process. Additionally, we defined the annealing fraction s at time t during annealing as specified in Eq. (9) below:

$$s = \begin{cases} -\frac{2(1-s_{\min})}{t_{\text{anl}}}t + 1 & (0 \le t \le t_{\text{anl}}/2) \\ \frac{2(1-s_{\min})}{t_{\text{anl}}}t - 1 + 2s_{\min} & (t_{\text{anl}}/2 \le t \le t_{\text{anl}}). \end{cases} \quad (9)$$

In Eq. (9), s linearly decreases from 1 to $s_{\min}$, from time 0 to time $t_{\text{anl}}/2$. After that, s linearly increases from $s_{\min}$ to 1, from time $t_{\text{anl}}/2$ to t_{anl}. Fig. 3 shows a graphical representation of Eq. (9). Reverse annealing uses the Hamiltonian in Eq. (1), and s is time-evolved according to Eq. (9). We repeated the annealing process, correction process, and selection process, 10 times ($N = 10$).[1]

We also compare the costs obtained by forward annealing using a quantum annealer. For forward annealing, we employed the D-Wave machine, fixed the execution time of quantum annealing at 2 ms, and obtained a set of 100 solutions, as in [22].

We ran all experimental conditions 20 times for each QAP, TSP, or MaxCut instance and obtained the average cost. Note that the number of binary variables that can be input to D-Wave Advantage is 180 when the QUBO matrix is fully connected [32], [33], and hence, it accepts a problem with 180 or less variables.

4.2 Quadratic Assignment Problem

4.2.1 Setup

The setup for QAP is described as follows: In the initial process, we configured the number of iterations for the random method as $R = 10$, since we can relatively obtain good initial solutions in QAP. In the annealing process, we set the minimum annealing fraction, denoted as $s_{\min}$, as 0.3, 0.4, and 0.5.[2]

We randomly generated six QAP instances[3] that can be input to the D-Wave machine in this experiment. These instances are denoted as n8_range[0-99], n8_range[0-299], n10_range[0-99], n10_range[0-299], n12_range[0-99], and n12_range[0-299], where the number following n represents the number of districts and the numbers following *range* specify the range of distances between every two districts.

We used the energy function of the QAP described in [16]. The weight of the energy function is defined as follows: When dealing with N facilities and N districts, the logistics value between facilities f_i and f_j is represented as $w(f_i, f_j)$, and the distance between the districts $l(f_i)$ and $l(f_j)$ is denoted as $d(l(f_i), l(f_j))$ where $l(f_i)$ denotes the assigned district of facility i. The weight of the constraint term in the QAP energy function is defined as $N \times \max(w(f_i, f_j)) \times \max(d(l(f_i), l(f_j)))$.

4.2.2 Result

Table 1 shows the experimental results. *FA* represents the cost obtained through forward annealing. *Opt* represents the optimal solution for the instance and is obtained through an exhaustive search. *diff* represents the percentage improvement of our proposed method compared to the baseline method. *N/A* indicates no feasible solution was obtained at all, and bold numbers indicate an improvement over the baseline method. Note that, when we perform forward annealing using a D-Wave machine, we could not obtain a solution satisfying the two-way 1-hot constraints in every instance.

Table 1 demonstrates that in the case of $s_{\min} = 0.3$, our proposed method improved the cost by 5.86% on average compared to the baseline method. However, for the n10_range[0-99] instance, the cost obtained was higher than

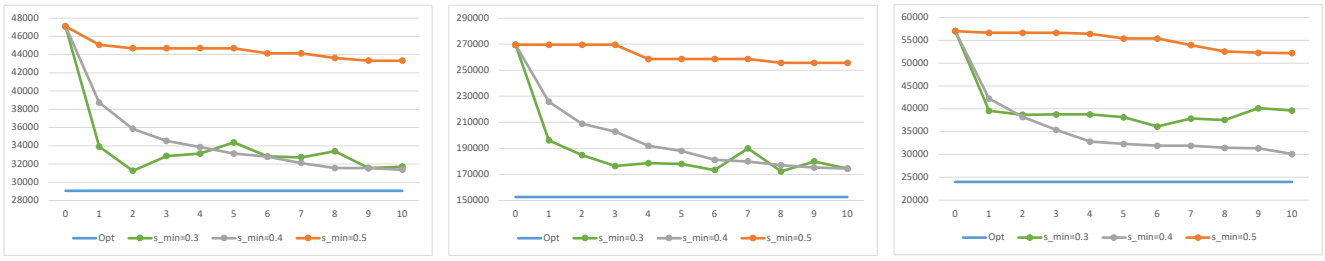
[1] As Fig. 4 and Fig. 5 (QAP and TSP) show, the solution is well converged in most of the cases if we set up an appropriate $s_{\min}$ value. In Fig. 6 (MaxCut problem), the solution is converged in some cases but it is still decreased at $N = 10$. However, the energy value is much decreased compared with the initial solution. Overall, $N = 10$ can be sufficient in the proposed iterative annealing.

[2] We set several $s_{\min}$ values among various $s_{\min}$ candidates, which lead to relatively good solutions, in all the experiments in Section 4.

[3] In QAPLIB [34], a small dataset that can be input to the D-Wave machine can be available. We have also conducted an experiment on these datasets. See the supplements at the end of this paper.

Table 1: Comparison results on QAP.

instance	#variables	FA	Opt	$s_{\min} = 0.3$			$s_{\min} = 0.4$			$s_{\min} = 0.5$		
				Baseline	Ours	diff [%]	Baseline	Ours	diff [%]	Baseline	Ours	diff [%]
n8_range[0-99]	64	N/A	29080	33923	**31710**	-6.52	38746	**31372**	-19.0	45072	**43324**	-3.88
n8_range[0-299]	64	N/A	152764	196106	**174471**	-11.0	196106	**174477**	-22.7	269563	**255622**	-5.17
n10_range[0-99]	100	N/A	24004	39578	39627	0.122	42252	**30114**	-28.7	56655	**52209**	-7.85
n10_range[0-299]	100	N/A	275884	464253	**438028**	-5.65	490521	**366389**	-25.3	627926	**557738**	-11.2
n12_range[0-99]	144	N/A	34412	70135	**68789**	-1.92	74178	**56103**	-24.4	94306	**85792**	-9.03
n12_range[0-299]	144	N/A	321138	735453	**660510**	-10.2	703244	**520792**	-25.9	911153	**861856**	-5.41



(a) n8_range[0-99]. (b) n8_range[0-299]. (c) n10_range[0-99].

(d) n10_range[0-299]. (e) n12_range[0-99]. (f) n12_range[0-299].

Fig. 4: Cost transition of QAP.

that of the baseline method. This is because the transverse magnetic field applied to the initial solution in reverse annealing was too strong, and hence the initial solution does not affect the final solution effectively.

In the case of $s_{\min} = 0.4$, our proposed method showcased a significant cost improvement of 24.3% on average compared to the baseline method. Our proposed method exhibited considerable influence in all instances, improving the cost by up to 28.7%. This is because the strength of the transverse magnetic field applied to the initial solution was weaker than the case of $s_{\min} = 0.3$, allowing the search for a better solution by keeping the initial solution.

In the case of $s_{\min} = 0.5$, our proposed method improved the cost by 7.09% on average compared to the baseline method, and the cost improvement is seen in all the instances. However, the improvement was smaller than $s_{\min} = 0.4$. This is because the strength of the transverse magnetic field applied to the initial solution was weaker than the case of $s_{\min} = 0.4$, and the initial solution was kept too much.

Fig. 4 shows the cost transition of QAP for each iteration. The vertical axis indicates the cost of QAP, and the horizontal axis indicates the number of iterations N, where $N = 0$ is the cost obtained by the initial process of the process (a) and $N = 1$ is the cost obtained by the baseline method. The blue line indicates the optimal solution for the instance, the line with the green marker indicates the case of $s_{\min} = 0.3$, the line with the gray marker indicates the case of $s_{\min} = 0.4$, and the line with the orange marker indicates the case of $s_{\min} = 0.5$.

Fig. 4 demonstrates that in the case of $s_{\min} = 0.3$, costs increase or decrease during iterations, and it is not expected that they converge to a specific value since the strength of the transverse magnetic field applied to the initial solution was too strong.

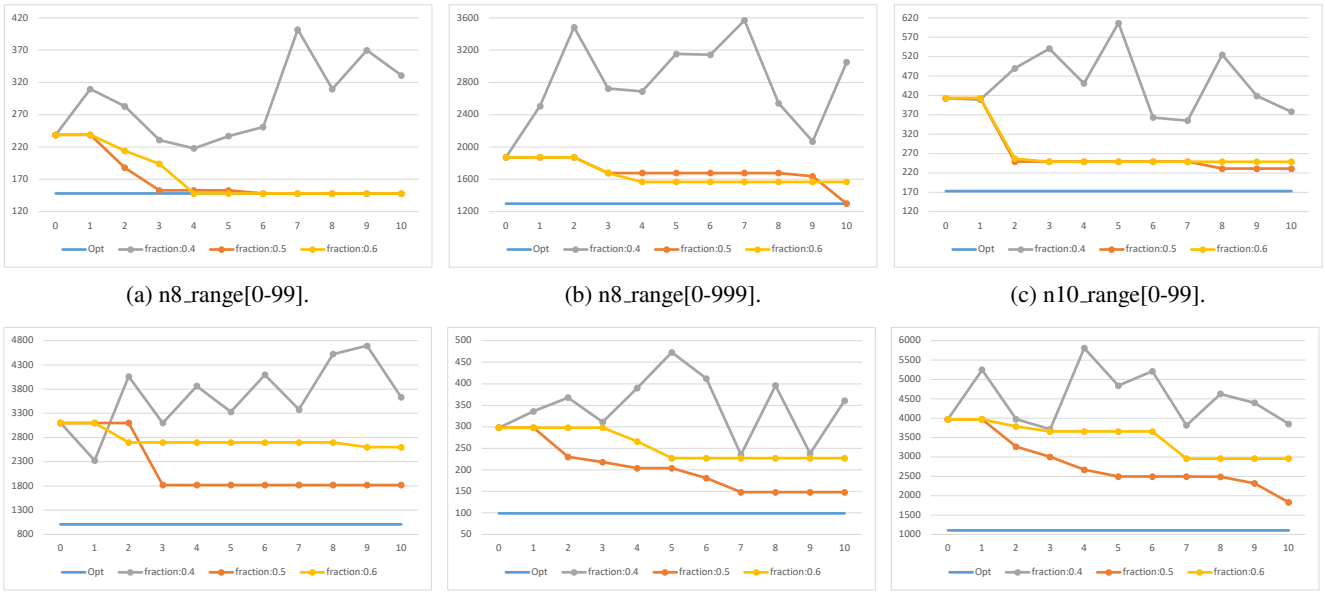
In the case of $s_{\min} = 0.4$, costs are improved in all iterations. The ratio of cost improvement decreases as the number of iterations N increases and is close to zero from $N = 9$ to $N = 10$, indicating that the cost converges at around $N = 10$ in this experiment.

In the case of $s_{\min} = 0.5$, the ratio of cost improvement is notably small but intermittent. Thus, by increasing N larger than 10, we can expect the cost to converge to a specific value, as in the case of $s_{\min} = 0.4$.

Based on these discussions above, our proposed method is most effective when the minimum annealing fraction $s_{\min}$ is 0.4 in this experiment.

Table 2: Comparison results on TSP.

instance	#variables	FA	Opt	$s_{\min} = 0.4$			$s_{\min} = 0.5$			$s_{\min} = 0.6$		
				Baseline	Ours	diff [%]	Baseline	Ours	diff [%]	Baseline	Ours	diff [%]
n8_range[0-99]	64	N/A	148	310	331	6.77	239	**148***	-38.1	239	**148***	-38.1
n8_range[0-999]	64	N/A	1300	2509	3052	21.6	1874	**1300***	-30.6	1874	**1569**	-16.3
n10_range[0-99]	100	N/A	173	409	**378**	-7.58	413	**231**	-44.1	413	**249**	-39.7
n10_range[0-999]	100	N/A	1007	2326	3632	56.1	3103	**1819**	-41.4	3103	**2601**	-16.2
n12_range[0-99]	144	N/A	99	336	361	7.44	298	**148**	-50.3	298	**227**	-23.8
n12_range[0-999]	144	N/A	1104	5249	**3852**	-26.6	3974	**1829**	-54.0	3974	**2959**	-25.5



(a) n8_range[0-99]. (b) n8_range[0-999]. (c) n10_range[0-99].

(d) n10_range[0-999]. (e) n12_range[0-99]. (f) n12_range[0-999].

Fig. 5: Cost transition in TSP.

4.3 Traveling Salesman Problem

4.3.1 Setup

The setup for TSP is described as follows: In the initial process, we configured the number of iterations for the random method as $R = 10$, since we can relatively obtain good initial solutions in TSP. In the annealing process, we set the minimum annealing fraction, denoted as $s_{\min}$, as 0.4, 0.5, and 0.6.

We randomly generated six TSP instances[1] that can be input to the D-Wave machine in this experiment. These instances are denoted as n8_range[0-99], n8_range[0-999], n10_range[0-99], n10_range[0-999], n12_range[0-99], and n12_range[0-999], where the number following *n* represents the number of cities and the numbers following *range* specify the range of distances between every two cities. We used the energy function of the TSP described in [16]. When the distance between cities c_i and c_j is expressed as $d(c_i, c_j)$, the weight of the constraint term in the TSP energy function is defined as $\max(d(c_i, c_j))$.

[1]Typical TSP instances are included in TSPLIB [35], but the minimum number of variables required by a TSPLIB instance is 196. Since this exceeds the number of variables that can be input to the D-Wave machine, we generate random TSP instances here.

4.3.2 Result

Table 2 shows the experimental results. *FA* represents the cost obtained through forward annealing. *Opt* represents the optimal solution for the instance and is obtained through an exhaustive search. *diff* represents the percentage improvement of our proposed method compared to the baseline method. *N/A* indicates no feasible solution was obtained at all, bold numbers indicate an improvement over the baseline method, and numbers with an asterisk indicate that the optimal solution has been reached. Note that, when we perform forward annealing using a D-Wave machine, we could not obtain a solution satisfying the two-way 1-hot constraints in every instance.

Table 2 demonstrates that in the case of $s_{\min} = 0.4$, the cost of our proposed method was, on average, 9.62% higher than that of the baseline method. This is because

the transverse magnetic field applied to the initial solution in reverse annealing was too strong, and hence the initial solution does not affect the final solution effectively.

In contrast, in the case of $s_{\min} = 0.5$, our proposed method showcased a significant cost improvement of 43.1% on average compared to the baseline method and reached the optimal solution for n8_range[0-99] and n8_range[0-999]. This is because the strength of the transverse magnetic field applied to the initial solution was weaker than the case of $s_{\min} = 0.4$, allowing the search for a better solution by keeping the initial solution.

In the case of $s_{\min} = 0.6$, our proposed method improved the cost by 26.6% on average compared to the baseline method. Although the percentage improvement was smaller than the case of $s_{\min} = 0.5$, we reached the optimal solution for the n8_range [0-99]. This is because the strength of the transverse magnetic field applied to the initial solution was weaker than the case of $s_{\min} = 0.5$, and the initial solution was kept too much.

Fig. 5 shows the cost transition of TSP for each iteration. The vertical axis indicates the cost of TSP, and the horizontal axis indicates the number of iterations N, where $N = 0$ is the cost obtained by the initial process of the process (a) and $N = 1$ is the cost obtained by the baseline method. The blue line indicates the optimal solution for the instance, the line with the gray marker indicates the case of $s_{\min} = 0.4$, the line with the orange marker indicates the case of $s_{\min} = 0.5$, and the line with the yellow marker indicates the case of $s_{\min} = 0.6$.

Fig. 5 demonstrates that in the case of $s_{\min} = 0.4$, the cost does not converge to a specific value as N increased because some iterations decrease the cost significantly while others increase it significantly since the transverse magnetic field applied to the initial solution in reverse annealing was too strong.

In the case of $s_{\min} = 0.5$, the five instances of n8_range[0-99], n8_range[0-999], n10_range[0-99], n10_range[0-999], and n12_range[0-99] showed the cost improvement in several iterations while it remains unchanged in many iterations. It indicates that by the time N reaches 10, the cost has converged to an optimal or near-optimal solution. In the instances of n12_range[0-999], the cost is repeatedly improved until N reaches 10, indicating that increasing N over 10 is expected to converge to an optimal or near-optimal solution.

In the case of $s_{\min} = 0.6$, the cost remains unchanged in many iterations but is improved in a few iterations.

Based on these discussions above, our proposed method is most effective when the minimum annealing fraction $s_{\min}$ is 0.5 in this experiment.

It is noted that we used randomly generated TSP instances in this experimental evaluation. As one of our future works, we will verify whether optimal solutions can be found when benchmark problems such as TSPLIB are used, as some instances were able to reach optimal solutions in this experiment, when a large-sized quantum annealer is available.

4.4 MaxCut problem

4.4.1 Setup

The setup for the MaxCut problem is described as follows: In the initial process, we configured the number of iterations for the random method as $R = 100$, since we require many iterations to obtain good initial solutions in the MaxCut problem, compared to QAP and TSP. In the annealing process, we set the minimum annealing fraction, denoted as $s_{\min}$, as 0.4, 0.5, and 0.6.

Typical MaxCut instances are included in Gset [36], but the minimum number of variables in the Gset instance is 800. Since this exceeds the number of variables that can be input to the D-Wave machine, we generated six MaxCut instances randomly that can be input to the D-Wave machine in this experiment. These instances are denoted as n50_w[1-100]_d75, n50_w[1-100]_d100, n100_w[1-100]_d75, n100_w[1-100]_d100, n150_w[1-100]_d75, and n150_w[1-100]_d100, where the number following n represents the number of vertices of a graph, the numbers following w specify the range of weights between every two vertices, and the number following d specifies the edge density of the graph. We used the energy function of the MaxCut problem described in [16].

4.4.2 Result

Table 3 shows the experimental results. In Table 3, the energy values finally obtained by every method are shown.[1] *FA* represents the energy value obtained through forward annealing. *diff* represents the percentage improvement of our proposed method compared to the baseline method. Bold numbers indicate an improvement over the baseline method. Note that the optimal solution was not obtained through an exhaustive search.

Table 3 demonstrates that in the case of $s_{\min} = 0.4$, our proposed method improved the energy value by 1.31% on average compared to the baseline method. This is because the transverse magnetic field applied to the initial solution in reverse annealing was too strong, and hence the initial solution does not affect the final solution effectively.

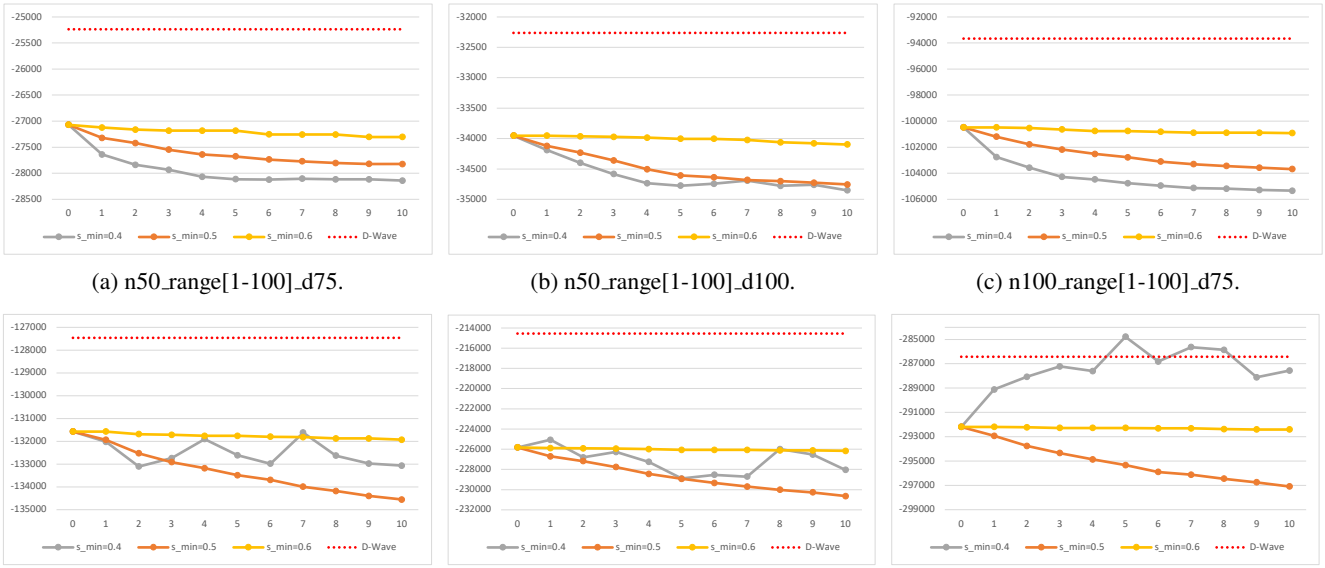
In the case of $s_{\min} = 0.5$, our proposed method showcased a significant energy value improvement of 1.88% on average compared to the baseline method. This is because the strength of the transverse magnetic field applied to the initial solution was weaker than the case of $s_{\min} = 0.4$, allowing the search for a better solution by keeping the initial solution.

In the case of $s_{\min} = 0.6$, our proposed method improved the energy value by 0.33% on average compared to the baseline method. This is because the strength of the

[1] Since the energy value is negative in the MaxCut problem [16], it tends to decrease as the iteration increases, as shown in Fig. 6, which means that the number of *cuts* in every MaxCut instance gradually increases.

Table 3: Comparison results on the MaxCut problem.

instance	#variables	FA	$s_{\min} = 0.4$			$s_{\min} = 0.5$			$s_{\min} = 0.6$		
			Baseline	Ours	diff [%]	Baseline	Ours	diff [%]	Baseline	Ours	diff [%]
n50_w[1-100]_d75	50	-25235	-27640	**-28140**	-1.81	-27321	**-27823**	-1.84	-27121	**-27303**	-0.670
n50_w[1-100]_d100	50	-32262	-34190	**-34851**	-1.93	-34122	**-34755**	-1.85	-33953	**-34097**	-0.423
n100_w[1-100]_d75	100	-93659.0	-102751	**-105336**	-2.52	-101196	**-103677**	-2.45	-100479	**-100910**	-0.429
n100_w[1-100]_d100	100	-127466	-132019	**-133066**	-0.793	-131935	**-134555**	-1.99	-131569	**-131928**	-0.273
n150_w[1-100]_d75	150	-214560	-225068	**-228041**	-1.32	-226703	**-230632**	-1.73	-225892	**-226161**	-0.119
n150_w[1-100]_d100	150	-286429	-289114	-287558	0.538	-292928	**-297082**	-1.42	-292188	**-292406**	-0.0743



(a) n50_range[1-100]_d75. (b) n50_range[1-100]_d100. (c) n100_range[1-100]_d75.

(d) n100_range[1-100]_d100. (e) n150_range[1-100]_d75. (f) n150_range[1-100]_d100.

Fig. 6: Energy value transition in the MaxCut problem.

transverse magnetic field applied to the initial solution was weaker than the case of $s_{\min} = 0.5$, and the initial solution was kept too much.

Fig. 6 shows the energy value transition of the MaxCut problem for each iteration. The vertical axis indicates the energy values of the MaxCut problem, and the horizontal axis indicates the number of iterations N, where $N = 0$ is the energy value obtained by the initial process of the process (a) and $N = 1$ is the energy value obtained by the baseline method. The line with the gray marker indicates the case of $s_{\min} = 0.4$, the line with the orange marker indicates the case of $s_{\min} = 0.5$, the line with the yellow marker indicates the case of $s_{\min} = 0.6$, and the red dotted line indicates the energy value obtained by forward annealing.

Fig. 6 demonstrates that in the case of $s_{\min} = 0.4$, the energy value tends to improve with each iteration because the search for the solution space by the quantum annealer that performs the annealing process of the process (b) is relatively easy for instances with a small number of variables or a small graph density such as n50_w[1-100]_d75, n50_w[1-100]_d100, and n100_w[1-100]_d75. The ratio of energy value improvement for each iteration decreases as N approaches 10, indicating convergence at around $N = 10$. On the other hand, for instances with a large number of variables or large graph density such as n100_w[1-100]_d100, n150_w[1-100]_d75, and n150_w[1-100]_d100, the search for the solution space becomes relatively more difficult. Thus, some iterations show a large energy value improvement, while others show a large increase.

In the case of $s_{\min} = 0.5$, the energy value improves with each iteration until N reaches 10, so further improvement and convergence can be expected when N is increased over 10.

In the case of $s_{\min} = 0.6$, the energy value is improved slightly with each iteration. Therefore, increasing N to 10 or more is expected to further improve and converge the energy values.

Based on these discussions above, our proposed method is most effective when the minimum annealing fraction $s_{\min}$ is 0.5 in this experiment, as well.

5. Conclusion

In this paper, we proposed a hybrid iterative annealing method for efficiently solving combinatorial optimization problems, utilizing a quantum annealer and a classical computer. The experimental evaluations showed that the pro-

posed method obtained at most 28.7% better solutions for QAP, 54.0% better solutions for TSP, and 2.52% better solutions for MaxCut problems. Also, the proposed method reaches an optimal solution in several TSP instances.

In the future, we will apply the proposed method to other Ising machines and evaluate the method to validate the proposed method further. We also compare our proposed method to novel classical methods and investigate the impact of each process in our method in the future.

Acknowledgments

This paper is based on the results obtained from a project, JPNP16007, subsidized by the New Energy and Industrial Technology Development Organization (NEDO), Japan.

References

[1] E. Filip and M. Otakar, "The travelling salesman problem and its application in logistic practice," WSEAS Transactions on Business and Economics, vol.8, no.4, pp.163–173, 2011.

[2] T.C. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," Econometrica: journal of the Econometric Society, pp.53–76, 1957.

[3] R.M. Karp, "Reducibility among combinatorial problems," Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department, pp.85–103, Springer, 1972.

[4] L. Pusey-Nazzaro *et al.*, "Adiabatic quantum optimization fails to solve the knapsack problem," arXiv preprint arXiv:2008.07456, 2020.

[5] M.R. Garey and D.S. Johnson, "The complexity of near-optimal graph coloring," Journal of the ACM (JACM), vol.23, no.1, pp.43–49, 1976.

[6] D. Willsch, M. Willsch, C.D. Gonzalez Calaza, F. Jin, H. De Raedt, M. Svensson, and K. Michielsen, "Benchmarking Advantage and D-Wave 2000Q quantum annealers with exact cover problems," Quantum Information Processing, vol.21, no.4, p.141, 2022.

[7] C.R. McLeod and M. Sasdelli, "Benchmarking D-Wave Quantum Annealers: Spectral Gap Scaling of Maximum Cardinality Matching Problems," International Conference on Computational Science, pp.150–163, Springer, 2022.

[8] M.W. Johnson, M.H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A.J. Berkley, J. Johansson, P. Bunyk, *et al.*, "Quantum annealing with manufactured spins," Nature, vol.473, no.7346, pp.194–198, 2011.

[9] "Third generation digital annealer white paper." https://www.fujitsu.com/jp/documents/about/research/techintro/3rd-g-da.pdf.

[10] H. Goto, K. Tatsumura, and A.R. Dixon, "Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems," Science advances, vol.5, no.4, p.eaav2372, 2019.

[11] P.L. McMahon, A. Marandi, Y. Haribara, R. Hamerly, C. Langrock, S. Tamate, T. Inagaki, H. Takesue, S. Utsunomiya, K. Aihara, *et al.*, "A fully programmable 100-spin coherent Ising machine with all-to-all connections," Science, vol.354, no.6312, pp.614–617, 2016.

[12] 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 Journal of Solid-State Circuits, vol.51, no.1, pp.303–309, 2015.

[13] E. Ising, "Beitrag zur theorie des ferromagnetismus," Zeitschrift fur Physik, vol.31, no.1, pp.253–258, Feb. 1925.

[14] H. Nishimori, Statistical physics of spin glasses and information processing: an introduction, Clarendon Press, 2001.

[15] E. Boros, P.L. Hammer, and G. Tavares, "Local search heuristics for quadratic unconstrained binary optimization (QUBO)," Journal of Heuristics, vol.13, pp.99–132, 2007.

[16] A. Lucas, "Ising formulations of many NP problems," Frontiers in physics, vol.2, p.5, 2014.

[17] S. Tanaka, R. Tamura, and B.K. Chakrabarti, Quantum spin glasses, annealing and computation, Cambridge University Press, 2017.

[18] K. Tanahashi, S. Takayanagi, T. Motohashi, 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, p.061010, 2019.

[19] K. Ikeda, Y. Nakamura, and T.S. Humble, "Application of quantum annealing to nurse scheduling problem," Scientific reports, vol.9, no.1, p.12837, 2019.

[20] T. Shirai and N. Togawa, "Spin-variable reduction method for handling linear equality constraints in Ising machines," IEEE Transactions on Computers, 2023.

[21] K. Fukada, M. Parizy, Y. Tomita, and N. Togawa, "A three-stage annealing method solving slot-placement problems using an Ising machine," IEEE Access, vol.9, pp.134413–134426, 2021.

[22] K. Fukada, T. Shirai, and N. Togawa, "Hybrid Iterative Annealing Method Using a Quantum Annealer and a Classical Computer," 2024 IEEE International Conference on Consumer Electronics (ICCE), pp.1–6, IEEE, 2024.

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

[24] F. Phillipson and H.S. Bhatia, "Portfolio optimisation using the d-wave quantum annealer," International Conference on Computational Science, pp.45–59, Springer, 2021.

[25] A. Callison and N. Chancellor, "Hybrid quantum-classical algorithms in the noisy intermediate-scale quantum era and beyond," Phys. Rev. A, vol.106, p.010101, Jul 2022.

[26] S. Morita and H. Nishimori, "Mathematical foundation of quantum annealing," Journal of Mathematical Physics, vol.49, no.12, p.125210, 12 2008.

[27] C. Carugno, M. Ferrari Dacrema, and P. Cremonesi, "Evaluating the job shop scheduling problem on a d-wave quantum annealer," Scientific Reports, vol.12, no.1, p.6539, 2022.

[28] M. Kim, D. Venturelli, and K. Jamieson, "Towards hybrid classical-quantum computation structures in wirelessly-networked systems," Proceedings of the 19th ACM Workshop on Hot Topics in Networks, pp.110–116, 2020.

[29] P. Codognet, "Comparing QUBO Models of the Magic Square Problem for Quantum Annealing," Metaheuristics International Conference, pp.470–477, Springer, 2022.

[30] P. Codognet, "Modeling the costas array problem in qubo for quantum annealing," European conference on evolutionary computation in combinatorial optimization (part of evostar), pp.143–158, Springer, 2022.

[31] E. Pelofske, A. Bärtschi, and S. Eidenbenz, "Quantum Annealing vs. QAOA: 127 Qubit Higher-Order Ising Problems on NISQ Computers," International Conference on High Performance Computing, pp.240–258, Springer, 2023.

[32] "The D-Wave Advantage2 Prototype." https://www.dwavesys.com/media/eixhdtpa/14-1063a-a_the_d-wave_advantage2_prototype-4.pdf.

[33] "Next-Generation Topology of D-Wave Quantum Processors." https://www.dwavesys.com/media/jwwj5z3z/14-1026a-c_next-generation-topology-of-dw-quantum-processors.pdf.

[34] R.E. Burkard, S.E. Karisch, and F. Rendl, "QAPLIB–a quadratic assignment problem library," Journal of Global optimization, vol.10, no.4, pp.391–403, 1997.

[35] G. Reinelt, "TSPLIB-A traveling salesman problem library," ORSA

journal on computing, vol.3, no.4, pp.376–384, 1991.
[36] "Gset." https://web.stanford.edu/ yyye/yyye/Gset/.

Keisuke Fukada received the B. Eng. and M. Eng. degrees in computer science from Waseda University, in 2021 and 2022, respectively. He is currently pursuing a Dr. Eng. degree in the Department of Computer Science and Communications Engineering, Waseda University. His research interests are mathematical optimization and quantum computation. He is a member of IEEE.

Tatsuhiko Shirai received B. Sci., M. Sci., and Dr. Sci. degrees from The University of Tokyo in 2011, 2013, and 2016, respectively. He is presently an assistant professor in the Department of Computer Science and Communications Engineering, Waseda University. His research interests are quantum dynamics, statistical mechanics, and computational science. He is a member of JPS.

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 quantum computation and integrated system design. He is a member of ACM, IEICE, and IPSJ.

Suppelements

We conducted the experiments using QAPLIB [34]. Note that we set $s_{\min}$ to 0.4, 0.5, and 0.6 while other experimental conditions, such as the annealing schedule, the energy function, and the weight of the energy function, are the same as described in Section 4.1 and Section 4.2.1.

Table 4 shows the experimental results. *FA* represents the cost through forward annealing. *Opt* represents the optimal solution for the instance obtained from [34], and *diff* represents the percentage improvement of our proposed method compared to the baseline method. *N/A* indicates no feasible solution was obtained at all, and bold numbers indicate an improvement over the baseline method. Note that, when we perform forward annealing using a D-Wave machine, we could not obtain a solution satisfying the two-way 1-hot constraints in every instance.

Table 4 demonstrates that in the case of $s_{\min} = 0.4$, our proposed method improved the cost by 3.46% on average compared to the baseline method. However, for the nug12 and tai12a instances, the cost obtained was higher than that of the baseline method. This is because the transverse magnetic field applied to the initial solution in reverse annealing was too strong, and hence the initial solution does not affect the final solution effectively.

In the case of $s_{\min} = 0.5$, our proposed method showcased a significant cost improvement of 7.22% on average compared to the baseline method. Similar to $s_{\min} = 0.4$, the cost obtained by our proposed method was slightly higher than that of the baseline method for the tai12a instance. However, our proposed method exhibited considerable influence in other instances, decreasing the cost by up to 12.1%. This is because the strength of the transverse magnetic field applied to the initial solution was weaker than the case of $s_{\min} = 0.4$, allowing the search for a better solution by keeping the initial solution.

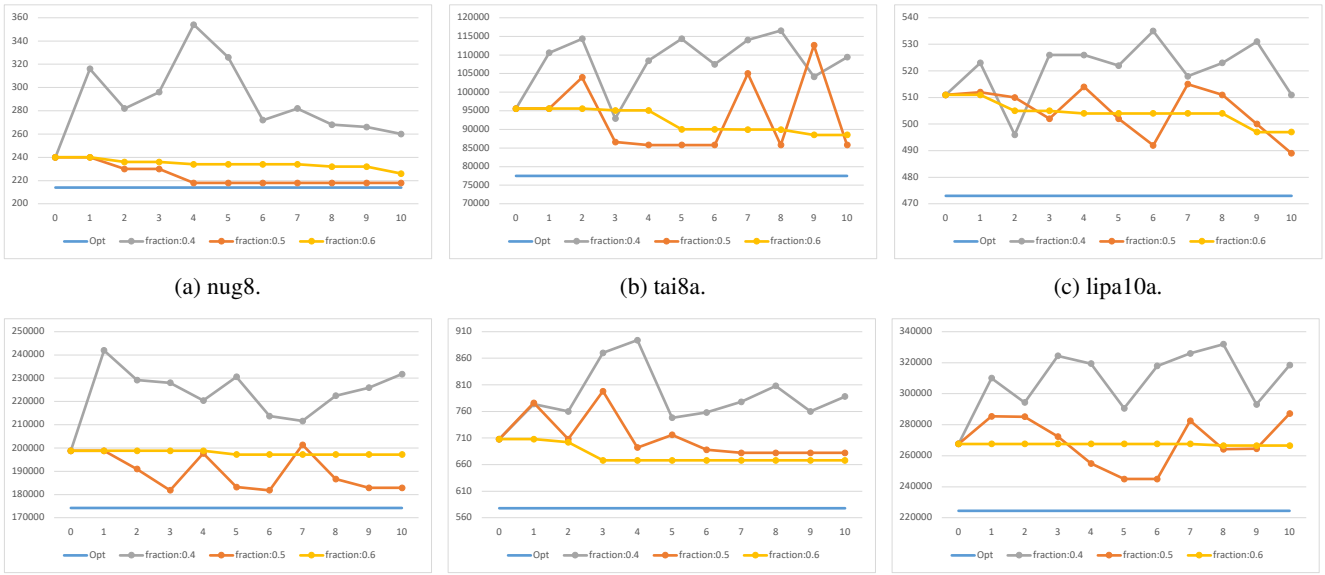
In the case of $s_{\min} = 0.6$, our proposed method improved the cost by 3.81% on average compared to the baseline method, and the cost improvement is seen in all the instances. However, the improvement was smaller than $s_{\min} = 0.5$. This is because the strength of the transverse magnetic field applied to the initial solution was weaker than the case of $s_{\min} = 0.5$, and the initial solution was kept too much.

Fig. 7 shows the cost transition of QAP for each iteration. The vertical axis indicates the cost of QAP, and the horizontal axis indicates the number of iterations N, where $N = 0$ is the cost obtained by the initial process of the process (a) and $N = 1$ is the cost obtained by the baseline method. The blue line indicates the optimal solution for the instance obtained from [34], the line with the gray marker indicates the case of $s_{\min} = 0.4$, the line with the orange marker indicates the case of $s_{\min} = 0.5$, and the line with the yellow marker indicates the case of $s_{\min} = 0.6$.

Fig. 7 demonstrates that in the case of $s_{\min} = 0.4$, the cost does not converge to a specific value as N increased because some iterations improve the cost significantly while

Table 4: Comparison results on QAP (QAP instances in QAPLIB are used).

instance	#variables	FA	Opt	$s_{\min} = 0.4$			$s_{\min} = 0.5$			$s_{\min} = 0.6$		
				Baseline	Ours	diff [%]	Baseline	Ours	diff [%]	Baseline	Ours	diff [%]
nug8	64	N/A	214	316	**260**	-17.7	240	**218**	-9.17	240	**226**	-5.83
tai8a	64	N/A	77502	110588	**109380**	-1.09	95566.0	**85790.0**	-10.2	95566.0	**88504.0**	-7.39
lipa10a	100	N/A	473	523	**511**	-2.29	512	**489**	-4.49	511	**497**	-2.74
rou10	100	N/A	174220	241936	**231762**	-4.20	198802	**182868**	-8.01	198802	**197158**	-0.827
nug12	144	N/A	578	774	788	1.80	776	**682**	-12.1	708	**668**	-5.65
tai12a	144	N/A	224416	310054	318416	2.70	285372	287182	0.634	267618	**266548**	-0.400



(a) nug8. (b) tai8a. (c) lipa10a.

(d) rou10. (e) nug12. (f) tai12a.

Fig. 7: Cost transition in QAP (QAP instances in QAPLIB are used).

others increase it significantly since the transverse magnetic field applied to the initial solution in reverse annealing was too strong.

In the case of $s_{\min} = 0.5$, the four instances of nug8, lipa10a, rou10, and nug12 showed the cost improvement in several iterations. It indicates that by the time N reaches 10, the cost has converged to a near-optimal solution. In the remaining two instances of tai8a and tai12a, the cost does not converge to a specific value as N increased because some iterations improve the cost significantly while others increase it significantly.

In the case of $s_{\min} = 0.6$, the cost remains unchanged in many iterations but is improved in a few iterations.

Based on these discussions above, our proposed method is most effective when the minimum annealing fraction $s_{\min}$ is 0.5 in this experiment, as in the discussion in the main text of Section 4.