SCIENTIFIC REPORTS
natureresearch

OPEN

Efficient partition of integer optimization problems with one-hot encoding

Received: 12 June 2019
Accepted: 27 August 2019
Published online: 10 September 2019

Shuntaro Okada[1,2], Masayuki Ohzeki[2,3,4] & Shinichiro Taguchi[1]

Quantum annealing is a heuristic algorithm for solving combinatorial optimization problems, and hardware for implementing this algorithm has been developed by D-Wave Systems Inc. The current version of the D-Wave quantum annealer can solve unconstrained binary optimization problems with a limited number of binary variables. However, the cost functions of several practical problems are defined by a large number of integer variables. To solve these problems using the quantum annealer, integer variables are generally binarized with one-hot encoding, and the binarized problem is partitioned into small subproblems. However, the entire search space of the binarized problem is considerably larger than that of the original integer problem and is dominated by infeasible solutions. Therefore, to efficiently solve large optimization problems with one-hot encoding, partitioning methods that extract subproblems with as many feasible solutions as possible are required. In this study, we propose two partitioning methods and demonstrate that they result in improved solutions.

The combinatorial optimization problems aim to minimize cost functions defined by discrete variables, and these problems often have significant real-world applications. In general, the cost function of a combinatorial optimization problem can be expressed as the Hamiltonian of a classical Ising model[1]. Therefore, many algorithms for solving combinatorial optimization problems have been inspired by physics. Simulated annealing (SA)[2] is one of the most famous algorithms, employing thermal fluctuations to escape local minima. In contrast to SA, quantum annealing (QA)[3] is a method that exploits quantum fluctuations and the resulting tunneling effect. A popular research topic involves evaluating whether it is more advantageous to employ quantum effects than thermal fluctuations, and numerous studies have been conducted on this topic[4–9]. In addition, the recent development of a commercial quantum annealer by D-Wave Systems Inc[10]. has attracted many companies and researchers. The performance of QA has been experimentally studied using the quantum annealer and compared with that of SA[11–13], and several companies have demonstrated the applicability of the annealer to practical problems[14–30].

The time-dependent Hamiltonian of QA is given as follows

$$\hat{H}(t) = A(t)\hat{H}_{\mathrm{q}} + B(t)\hat{H}_0, \tag{1}$$

where $\hat{H}_0$ is the target Hamiltonian representing the cost function, and $\hat{H}_{\mathrm{q}}$ denotes the quantum fluctuation term for which the ground state is trivial. The initial values of the coefficients are set to $A(0) = 1$ and $B(0) = 0$, and the system is prepared in the trivial ground state determined by $\hat{H}_{\mathrm{q}}$. Then, the strength of the quantum fluctuation is reduced toward zero, and the coefficients are set to $A(\tau) = 0$ and $B(\tau) = 1$ at the end of QA, where τ is the annealing time. The dynamics of the system is described by the Schrödinger equation:

$$i\frac{d}{dt}\psi(t) = \hat{H}(t)\psi(t), \tag{2}$$

where $\psi(t)$ is the state vector of the system, and $\hbar$ is set to 1 for simplicity. Given that the coefficients change sufficiently slowly, the adiabatic theorem[31] ensures that the system remains close to the instantaneous ground state of the time-dependent Hamiltonian. Thus, by setting the annealing time τ to be sufficiently large, the ground states of the target Hamiltonian $\hat{H}_0$ can be obtained with high probability.

[1]Electronics R & I Division, DENSO CORPORATION, Tokyo, 108-0075, Japan. [2]Graduate School of Information Sciences, Tohoku University, Sendai, 980-8579, Japan. [3]Institute of Innovative Research, Tokyo Institute of Technology, Yokohama, 226-8503, Japan. [4]Sigma-i Co. Ltd., Tokyo, 108-0075, Japan. Correspondence and requests for materials should be addressed to S.O. (email: shuntaro.okada.j3z@jp.denso.com)

The current version of the D-Wave quantum annealer (D-Wave 2000Q) implements transverse-magnetic-field QA, for which the quantum fluctuation is given as follows:

$$\hat{H}_{\mathrm{q}} = -\sum_{i=1}^{N_{\mathrm{q}}} \hat{\sigma}_i^{(x)}, \tag{3}$$

where N_{q} denotes the number of qubits. The quantum annealer can handle a cost function as follows:

$$\hat{H}_0 = \sum_{(i,j)\in \mathrm{chimera}} J_{ij}\hat{\sigma}_i^{(z)}\hat{\sigma}_j^{(z)} + \sum_{i=1}^{N_{\mathrm{q}}} h_i \hat{\sigma}_i^{(z)}, \tag{4}$$

where the interactions between qubits are restricted to the Chimera graph[32], which is a 16 × 16 grid of complete bipartite graphs $K_{4,4}$ in D-Wave 2000Q. It should be noted that the number of operable qubits is less than $N_{\mathrm{q}} = 2{,}048$ due to defects in the qubits and connectivities.

Due to the limited number of available qubits, large optimization problems cannot be solved directly using the D-Wave quantum annealer. In real settings, large problems are partitioned into subproblems that can be handled by the quantum annealer. The subproblems are iteratively optimized by the quantum annealer, and the optimization result is used to improve the current solution[33–35]. A cluster of spins in the subproblem is simultaneously updated in this scheme; this iterative method is a type of large-neighborhood local search algorithm[36]. Although these algorithms can be performed using classical computers, subproblems are fundamentally restricted to tree structures that are solvable in polynomial time by belief propagation or dynamic programming[37–40]. Therefore, using the quantum annealer is advantageous if it can solve subproblems with many closed loops more efficiently than classical algorithms. Furthermore, solving subproblems that are as large as possible is essential for improving solution accuracy[41]. The size of subproblems that can be embedded into the quantum annealer strongly depends on the quality of the minor embedding, in particular for problems with few interactions. Because subproblems must be iteratively embedded, fast algorithms for embedding larger subproblems are required for exploiting the potential of the quantum annealer. Although complete-graph embedding[42–44] can be used for problems with dense interactions, a subproblem-embedding algorithm, that was developed in a previous study[41], may be effective in improving the solution accuracy of sparse problems.

In addition, the quantum annealer requires the cost function to be represented in the form of a quadratic unconstrained binary optimization (QUBO) problem or Ising model; however, many cost functions in practical problems are defined by integer variables. The binarization of integer variables is generally achieved using one-hot encoding[1]. For example, suppose that we wish to solve the following integer optimization problem with N integer variables $\{S_i\}_{i=1,2,\ldots,N}$:

$$\underset{\{S_i\}}{\arg\min} \sum_{i=1}^{N-1} J_{i,i+1}\delta(S_i,\, S_{i+1}), \tag{5}$$

where $S_i \in (1, 2, \ldots, Q)$, Q is the number of components, $J_{i,i+1}$ is an interaction between S_i and S_{i+1}, and δ is the Kronecker delta function. The integer variables $\{S_i\}_{i=1,2,\ldots,N}$ can be binarized by one-hot encoding as follows:

$$\underset{\{x_i^{(q)}\}}{\arg\min} \sum_{i=1}^{N-1} J_{i,i+1} \sum_{q=1}^{Q} x_i^{(q)} x_{i+1}^{(q)} \ \mathrm{s.t.} \quad \sum_{q=1}^{Q} x_i^{(q)} = 1, \tag{6}$$

where $x_i^{(q)} \in (0,\ 1)$ is a binary variable that is assigned to component q of S_i, and $x_i^{(q)} = 1$ indicates that component q is selected for S_i. In addition, feasible solutions are constrained to configurations in which exactly one component is selected for each S_i. Subsequently, a penalty term is introduced to obtain the following unconstrained form:

$$H_0 = \sum_{i=1}^{N-1} J_{i,i+1} \sum_{q=1}^{Q} x_i^{(q)} x_{i+1}^{(q)} + \lambda \sum_{i=1}^{N} \left(\sum_{q=1}^{Q} x_i^{(q)} - 1 \right)^2, \tag{7}$$

where the second term formulates the penalty term which is introduced to extract feasible solutions satisfying the constraint $\sum_{q=1}^{Q} x_i^{(q)} = 1$, which we call the *one-hot constraint*, and parameter λ controls the strength of the penalty term. By setting parameter λ to a sufficiently large value, the ground states of the original integer optimization problem (Eq. (5)) are correctly encoded. However, the performance of the D-Wave quantum annealer is significantly affected by noise and intrinsic control errors when λ is larger than necessary. Therefore, to obtain highly accurate solutions, we must explore an appropriate value of λ, which is a tedious task for optimization under the one-hot constraint. In addition, the entire search space of the binarized optimization problem (Eq. (7)) is dominated by infeasible solutions. Figure 1(a) presents the problem graph of Eq. (7). In this figure, vertices and edges represent binary variables and the interactions between them, respectively. Q binary variables $\{x_i^{(q)}\}_{q=1,2,\ldots,Q}$ are assigned to each S_i, and the total number of binary variables is NQ. Although the number of configurations of the binary variables is 2^{NQ}, the number of feasible solutions is only Q^N. Therefore, to efficiently solve large optimization problems under the one-hot constraint using the quantum annealer, partitioning methods are required for extracting subproblems with as many feasible solutions as possible. A simple example of an undesirable partition is depicted in Fig. 1(b). Here, suppose that we wish to improve the current solution presented in Fig. 1(b) and that the three binary variables enclosed by the green rectangle are extracted as the subproblem. In this case, superior

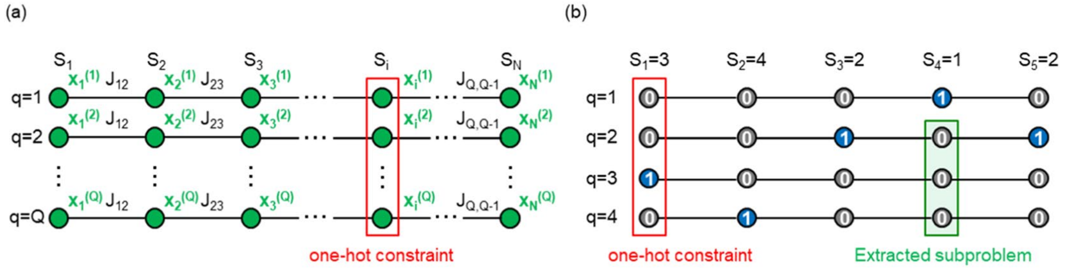

Figure 1. (**a**) Problem graph of Eq. (7). Vertices and edges represent binary variables $x_i^{(q)}$ and the interactions between them, respectively. Although the penalty term generates fully connected vertical interactions between $x_i^{(q)}$ and $x_i^{(q')}$, these are not shown for simplicity. Q binary variables $\{x_i^{(q)}\}_{q=1,2,\ldots,Q}$ are assigned to each S_i, and the total number of binary variables is NQ. Although there exist 2^{NQ} configurations of the binary variables, only Q^N configurations satisfy the one-hot constraint. (**b**) Example of an undesirable partition, where binary variables enclosed by the green rectangle are extracted as a subproblem. The binary variables colored blue represent the tentatively selected components in the current solution. Superior feasible solutions cannot be explored by optimizing the extracted subproblem.

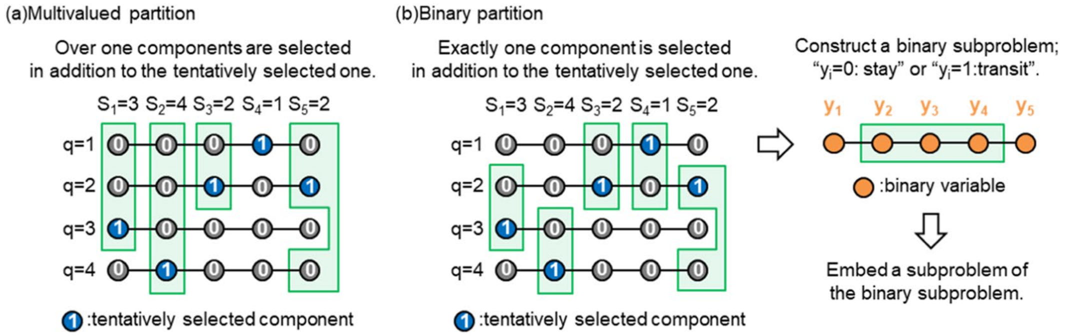
(a)Multivalued partition
Over one components are selected in addition to the tentatively selected one.
S1=3 S2=4 S3=2 S4=1 S5=2
q=1
q=2
q=3
q=4
1 :tentatively selected component
(b)Binary partition
Exactly one component is selected in addition to the tentatively selected one.
S1=3 S2=4 S3=2 S4=1 S5=2
q=1
q=2
q=3
q=4
1 :tentatively selected component
Construct a binary subproblem; "yi=0: stay" or "yi=1:transit".
y1 y2 y3 y4 y5
:binary variable
Embed a subproblem of the binary subproblem.

Figure 2. Proposed methods for finding superior solutions to large optimization problems under the one-hot constraint.

feasible solutions cannot be explored by optimizing the subproblem because only the current solution in the subproblem satisfies the one-hot constraint. To the best of our knowledge, the partitioning method proposed by Nishimura *et al.*[30] is the first to focus on the one-hot constraint. This method is applicable to the double-constrained problems $\sum_q x_i^{(q)} = 1$ and $\sum_i x_i^{(q)} = 1$, such as the assignment problem and the traveling salesman problem. However, the extracted subproblems still contain infeasible solutions for which parameter λ must be adjusted. In this study, we propose two partitioning methods applicable to problems whose cost function involves a single one-hot constraint, as illustrated in Eq. (7). The first method is similar to the partition proposed by Nishimura *et al.*, while the second method extracts subproblems comprising only feasible solutions and does not require adjusting parameter λ. The performance of the proposed methods is assessed for several Potts models, which are generalized Ising models whose cost function is defined by integer variables[45]. We demonstrate that the proposed methods efficiently obtain superior solutions.

Results

In this section, we propose efficient partitioning methods for solving large integer optimization problems under the one-hot constraint. In addition, we assess the performance of the proposed methods for several Potts models.

Proposed methods. We propose two partitioning methods: a multivalued partition and a binary partition. These methods are summarized in Fig. 2. Both methods extract a subproblem that involves binary variables assigned to the tentatively selected components for each S_i. The resulting subproblems include feasible solutions other than the current solution.

The multivalued partition extracts a subproblem with two or more components for each S_i, as illustrated in Fig. 2(a). In addition to the tentatively selected component, the multivalued partition randomly selects one or more components for each S_i, and then extracts a subproblem that comprises the binary variables assigned to the selected components. The extracted subproblem involves feasible solutions other than the current solution, and the randomly selected components are explored for each S_i by optimizing the subproblem. However, the extracted subproblem still contains infeasible solutions, and the penalty term remains in the cost function of the subproblem. This partitioning method is similar to the partition proposed by Nishimura *et al.* Although subproblems are embedded using complete-graph embedding in the study by Nishimura *et al.*[30], we employed

Model	J_{ij}	Δ_{ij}	λ
Ferromagnetic Potts model	−1	0	3.3
Anti-ferromagnetic Potts model	+1	0	1.0
Potts glass model	+1 (50%) or −1 (50%)	0	3.3
Potts gauge glass model	−1	0 (50%) or +1 (25%) or −1 (25%)	3.3

Table 1. Parameter settings of cost function Eq. (10).

the subproblem-embedding algorithm developed in our previous paper[41]. Details of achieving a multivalued partition using the subproblem-embedding algorithm are provided in the Methods section.

The binary partition is summarized in Fig. 2(b). In addition to a tentatively selected component, the binary partition randomly selects exactly one component for each S_i. Subsequently, new binary variables $\{y_i\}_{i=1,2,\ldots,N}$ that represent "stay in the tentatively selected component ($y_i = 0$)" or "transit to the randomly selected component ($y_i = 1$)" are introduced for each S_i, and a binary subproblem is constructed whose cost function is defined by $\{y_i\}_{i=1,2,\ldots,N}$. The cost function of the binary subproblem is derived in the Methods section. Thereafter, a subproblem of the binary subproblem is embedded into the D-Wave quantum annealer by the subproblem-embedding algorithm[41]. Here, the cost function of the binary subproblem does not involve the penalty term because all solutions in the binary subproblem are feasible. Therefore, the binary partition does not require adjusting parameter λ. In addition, a larger number of binary variables can be embedded into the D-Wave quantum annealer because the penalty term, which generates fully connected interactions between $x_i^{(q)}$ and $x_i^{(q\prime)}$, is not involved. Consequently, the number of feasible solutions involved in the embedded subproblem is significantly increased using the binary partition. The binary subproblem can be regarded as one of the simplest cases of optimization under the half-hot constraint[46]. The penalty term of the half-hot constraint is given by

$$\lambda\sum_{i=1}^{N}\left(\sum_{q=1}^{Q}x_i^{(q)} - \frac{Q}{2}\right)^2, \tag{8}$$

and $Q/2$ components are extracted. The half-hot constraint is proposed to avoid the difficulty caused by the longitudinal magnetic field in the penalty term of the one-hot constraint. This difficulty is avoidable using the binary partition, which may contribute to improving solution accuracy. A disadvantage of the binary partition is that only two components are considered for each integer variable. As demonstrated in the following subsection, this leads to poor performance for the ferromagnetic Potts model.

Performance assessment. The performance of the proposed methods is evaluated for the following four types of Potts models on a cubic lattice with 10 × 10 × 10 integer variables: the ferromagnetic, anti-ferromagnetic, Potts glass[47] and Potts gauge glass[48,49] models. While the ground states of the ferromagnetic and anti-ferromagnetic Potts models are trivial, it is generally difficult to obtain the ground states of the Potts glass and Potts gauge glass models due to competing interactions.

The cost function is given by

$$H_0 = \sum_{<i,j>} J_{ij}\delta(S_i, S_j + \Delta_{ij}), \tag{9}$$

where Q is set to 4, $S_i \in (1, 2, 3, 4)$, $\Delta_{ij} \in (0, \pm 1)$, δ is the Kronecker delta function, and J_{ij} represents the interaction between the nearest neighbors on the cubic lattice with the periodic boundary condition. The cost function is represented in QUBO form using the one-hot constraint as follows:

$$H_0 = \sum_{<i,j>} J_{ij}\sum_{q=1}^{4}x_i^{(q)}x_j^{(q-\Delta_{ij})} + \lambda\sum_{i=1}^{1000}\left(\sum_{q=1}^{4}x_i^{(q)} - 1\right)^2. \tag{10}$$

Parameters J_{ij}, Δ_{ij}, and λ in each model are presented in Table 1. $\Delta_{ij} \neq 0$ generates interactions between different components in the Potts gauge glass model, and Fig. 3 illustrates the local interactions generated by the first term of Eq. (10). Although it is generally difficult to determine an appropriate value of λ a priori, we can derive the lower bound of λ to correctly encode the original optimal solutions for the ferromagnetic and anti-ferromagnetic Potts models. The lower bound strongly depends on whether there exist infeasible solutions that set the first term of Eq. (10) to a smaller value than that of the original optimal solutions. $\lambda > 0$ is sufficient if such infeasible solutions do not exist; however, a sufficiently large value of λ is necessary if such infeasible solutions exist. For the ferromagnetic Potts model, the first term of Eq. (10) for an infeasible solution with $x_i^{(1)} = x_i^{(2)} = 1$, $x_i^{(q\geq 3)} = 0$ is lower than that of the original optimal solution (e.g., $x_i^{(1)} = 1$, $x_i^{(q\geq 2)} = 0$) by $3N$. Because the second term in Eq. (10) increases by $N\lambda$ in this infeasible solution, $\lambda > 3$ is required. In this study, we set $\lambda = 3.3$ because an unnecessarily large value is not preferable, as mentioned in the Introduction section. While for the anti-ferromagnetic Potts model, the original optimal solutions minimize the first term of Eq. (10). Therefore, $\lambda > 0$ is sufficient, and we set $\lambda = 1.0$, which is the same value as J_{ij} in the first term. However, for the Potts glass and Potts gauge glass models, the lower bound cannot be derived because the original optimal solutions are not trivial. At least, by setting $\lambda > 3$, we can restrict energy changes caused by a single-spin flip from the optimal solutions to be a positive value, and λ is set to 3.3 in this study.

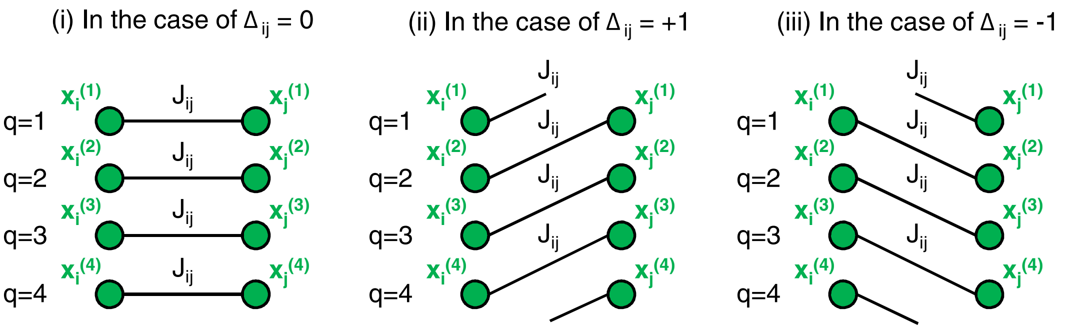
(i) In the case of Δ ij = 0
(ii) In the case of Δ ij = +1
(iii) In the case of Δ ij = -1
q=1
q=2
q=3
q=4
xi(1)
xi(2)
xi(3)
xi(4)
xj(1)
xj(2)
xj(3)
xj(4)
Jij

Figure 3. Local interactions generated by the first term of Eq. (10). $\Delta_{ij} \neq 0$ causes interactions between different components.

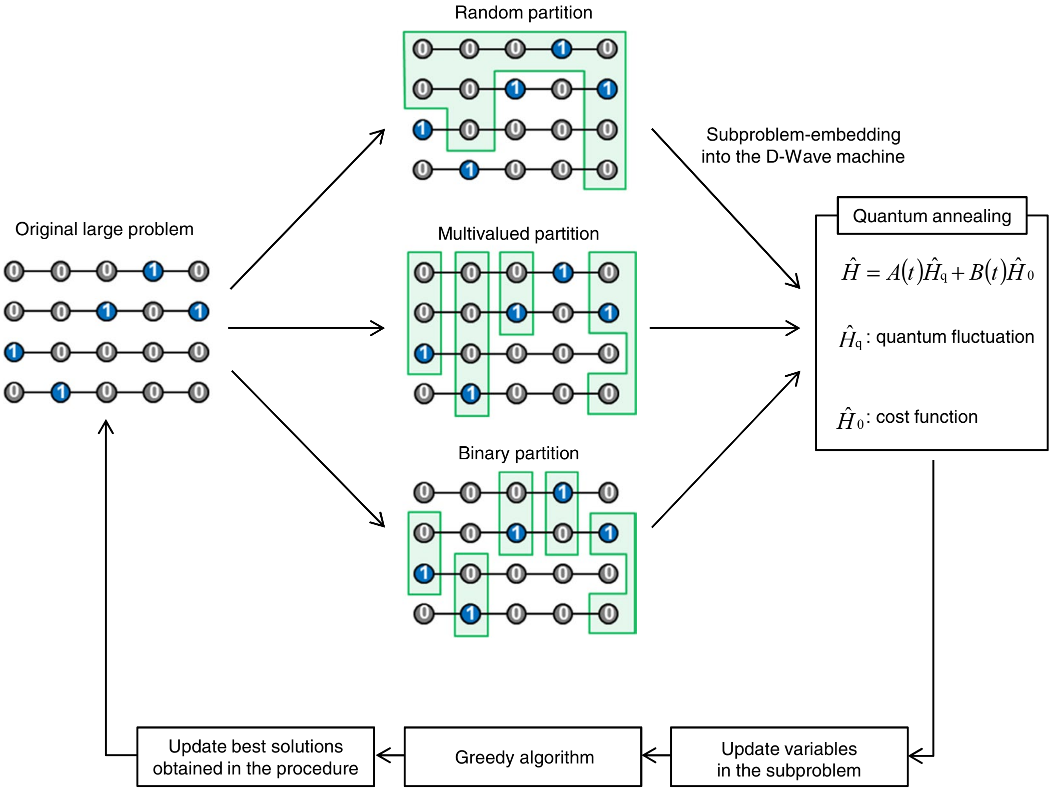

Figure 4. Optimization process demonstrated in this study. The solution accuracy is evaluated for the three partitioning methods.

The optimization process demonstrated in this study is illustrated in Fig. 4. The original large problem is partitioned using three partitioning methods: random, multivalued, and binary partitions. The random partition does not address whether an extracted subproblem contains feasible solutions for each S_i. A subproblem-embedding algorithm proposed in the literature[41] is used for embedding a subproblem into the D-Wave quantum annealer with defects in qubits and the interactions between them (details on embeddings are provided in the Methods section). After optimizing the embedded subproblem by the D-Wave quantum annealer under the parameter settings provided in Table 2, the variables in the subproblem are replaced by the best solution among the 1,000 solutions obtained using the quantum annealer. Subsequently, a greedy algorithm is executed by a conventional digital computer to recover the one-hot constraint and obtain an exact (local) minimum. In this algorithm, if there exist integer variables violating the one-hot constraint, the constraint is first recovered by extracting the integer variables and selecting exactly one component that minimizes the local energy for each integer variable. Then, an integer variable is randomly selected, and the tentatively selected component is replaced with one that minimizes the local energy. Refining the current solution is completed when all local energies are minimized. Finally, the best solution obtained in the procedure is updated, and the above processes are iterated. We then compare the solution accuracy for the three partitioning methods.

Parameter	Value
solver	D-Wave 2000Q_2
annealing time	20[μs]
Num_reads	1,000
auto_scale	True
postprocess	none

Table 2. Parameter settings of D-Wave quantum annealer.

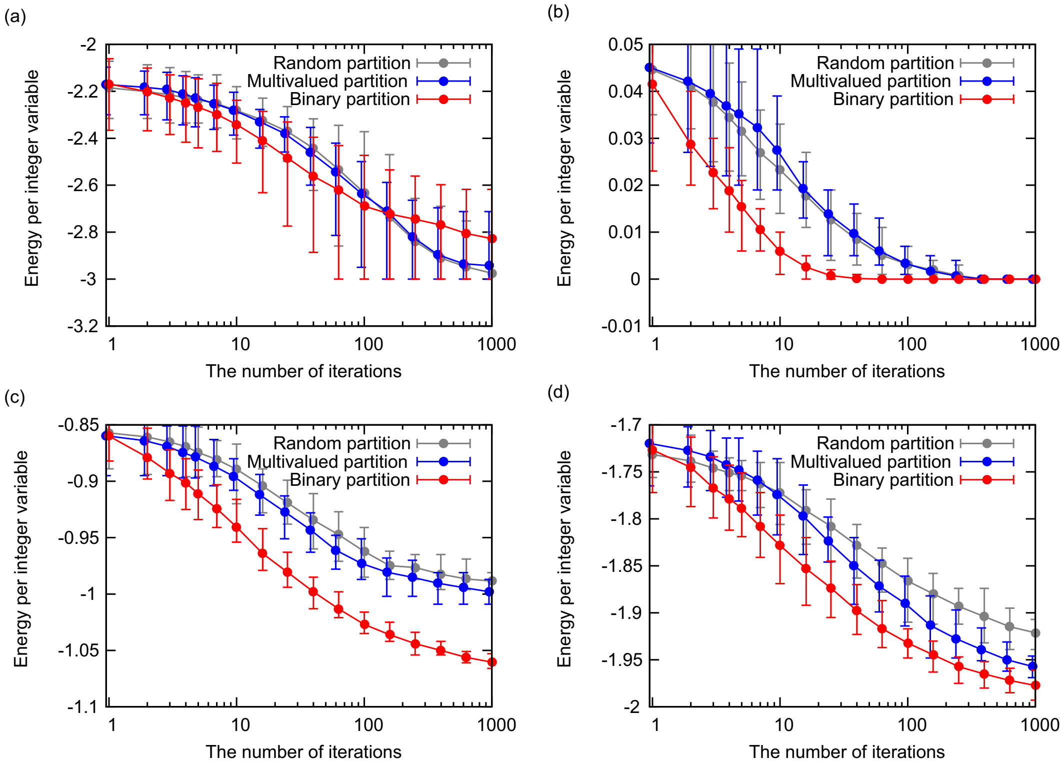

Figure 5. Energies obtained using the three partitioning methods. The average, maximum, and minimum energies for 16 trials are plotted. (**a**) Ferromagnetic Potts model. (**b**) Anti-ferromagnetic Potts model. (**c**) Potts glass model. (**d**) Potts gauge glass model.

Figure 5 represents the energies obtained by the three partitioning methods. The average, maximum, and minimum energies for 16 trials are plotted, and the same 16 initial states are used for each partitioning method. The horizontal axis represents the number of iterations, which is the number of subproblem optimizations performed by the D-Wave quantum annealer. The plot for the multivalued partition is shifted slightly to the left to avoid overlap with other plots. Figure 5(a,b) illustrate the energies obtained for the ferromagnetic and anti-ferromagnetic Potts models, respectively. The ground states of these models are trivial, and the minimum energy is -3 and 0 for the ferromagnetic and anti-ferromagnetic Potts models, respectively. Although the multivalued partition is expected to solve large optimization problems more efficiently than the random partition, the performances of the random and multivalued partitions are almost identical. The performance of the binary partition differs from the other methods, however; it is the lowest for the ferromagnetic Potts model, and the highest for the anti-ferromagnetic Potts model. Figure 5(c,d) present the energies obtained for the Potts glass and Potts gauge glass models, respectively. As expected, superior solutions are obtained with a smaller number of iterations using the multivalued partition rather than the random partition, in particular for the Potts gauge glass model. Of the three partitioning methods, the binary partition shows the highest performance for both the Potts glass and Potts gauge glass models.

Discussion

In this section, we discuss the differences between the three partitioning methods. The following three questions arise from the results presented in the previous section.

- Why is the multivalued partition not superior to the random partition for the ferromagnetic and anti-ferromagnetic Potts models?
- Why is the performance of the binary partition the lowest for the ferromagnetic Potts model?

- Why does the binary partition exhibit the highest performance for all models except for the ferromagnetic Potts model?

A possible answer to the first question is that, for the ferromagnetic and anti-ferromagnetic Potts models, improved feasible solutions can be obtained through infeasible solutions because there likely exist infeasible solutions whose energy is lower than that of a current feasible solution in a neighbor. Figure 6(a) presents a simple example for the one-dimensional ferromagnetic Potts model. We assume that the binary variable enclosed by the green rectangle is extracted as a one-variable subproblem, which is one of the simplest cases of the random partition. The energy change caused by flipping the extracted binary variable is $-2J+\lambda$ because two interactions are simultaneously recovered ($-2J$) and the one-hot constraint is violated ($+\lambda$). If $\lambda < 2J$, flipping the binary variable decreases the energy despite violating the constraint. It should be noted that $\lambda > J$ is sufficient to correctly encode the ground states of the one-dimensional ferromagnetic Potts model. This is because the energy of the lowest-energy infeasible states, in which two components are commonly selected for each S_i, is $-2NJ+N\lambda$ and must be larger than that of the ground states ($-NJ$). Consequently, if λ is appropriately tuned ($J < \lambda < 2J$), the current solution is updated to the infeasible solution by optimizing the subproblem, and superior feasible solutions are obtained via the infeasible solution. Whether energy changes become negative or not in spite of violating the one-hot constraint strongly depends on the number of simultaneously recovered interactions. For the ferromagnetic and anti-ferromagnetic Potts models without competing interactions, many interactions can be simultaneously recovered by violating the one-hot constraint. As a result, the multivalued partition is not effective in improving the solution accuracy for these models. In contrast, for the Potts glass and Potts gauge glass models with competing interactions, the performance of the multivalued partition is superior to that of the random partition.

In answer to the second question, subproblems that can eliminate domain walls are rarely extracted by the binary partition. Figure 6(b) presents one of first excited states, which is commonly observed in the optimization of the ferromagnetic Potts model. The 10 variables in Fig. 6(b) are divided into two domains: five variables S_1, ..., S_5 are aligned to $q=1$ in one domain, while the remaining five variables S_6, ..., S_{10} are aligned to $q=2$ in the other domain. The boundary between the domains is referred to as the domain wall. To improve the current solution, an extracted subproblem must contain one of the ground states because the current solution is the first excited state. For example, to align all integer variables $\{S_i\}_{i=1,2,\ldots,10}$ to $q=1$, component $q=1$ must be selected for integer variables S_6, ..., S_{10}. The probability of component $q=1$ being selected for S_6, ..., S_{10} is $(1/3)^5 = 1/243$ because, in addition to the tentatively selected component, the binary partition randomly selects one component for each S_i. This probability exponentially decreases with respect to the number of variables, and the extraction of only two components is not suitable for the ferromagnetic Potts model. We further conjecture that the binary partition exhibits poor performance for optimization problems containing ferromagnetically ordered domains, and that the concomitant use of the binary and multivalued partitions may be preferable for such problems.

The answer to the third question is that there exist several binary subproblems that can improve the current solution. Figure 6(c) illustrates the local interactions in the anti-ferromagnetic Potts model. The current solution is one of the first excited states in which the local energy with respect to S_1 and S_4 is not minimized, which we say "the interaction between S_1 and S_4 is *broken*" in this paper. Assuming that the integer variable S_4 is updated to improve the current solution, there are two binary subproblems that can improve the current solution. Therefore, the disadvantage of the binary partition, which is that only two components are considered for each integer variable, is mitigated for the optimization of the anti-ferromagnetic Potts model. We can thus exploit the advantages of the binary partition, which are that the extracted subproblems contain a larger number of feasible solutions and that the adjustment of parameter λ is not required. This is also the case for the Potts glass and Potts gauge glass models, in which competing interactions generate several binary subproblems that improve the current solution. Figure 6(d) presents a simple example for the Potts gauge glass model. One of the ground states and first excited states are illustrated at the top of Fig. 6(d), where the interaction depicted by a dashed line represents the broken interaction. There exist no configurations that minimize all of the local energies due to the competing interaction between different components, and one interaction is broken even in the ground state. Suppose that we update integer variable S_4 to improve the current solution in the first excited state, then there are two binary subproblems that can improve the current solution, as illustrated at the bottom of Fig. 6(d). One subproblem recovers the interaction between S_1 and S_4, while the other subproblem recovers the interaction between S_3 and S_4. The competing interactions generate two binary subproblems that improve the current solution, and each subproblem recovers different interactions. Thus, the disadvantage of the binary partition is mitigated as long as Q is not very large. It should be noted that although the number of binary subproblems that improve the current solution increases as Q is increased for the anti-ferromagnetic Potts model, this number does not increase for the Potts gauge glass model.

Conclusion

In this study, we proposed two partitioning methods to efficiently solve large optimization problems under the one-hot constraint using the D-Wave quantum annealer. The performance of the proposed methods was assessed for the ferromagnetic, anti-ferromagnetic, Potts glass, and Potts gauge glass models. Of the three partitioning methods, the binary partition showed the highest performance for all models except for the ferromagnetic Potts model. The advantages of the binary partition are that it enables embedding a larger number of binary variables and does not require adjusting parameter λ. However, its disadvantage is that only two components are considered for each integer variable. Although this disadvantage leads to poor performance for the ferromagnetic Potts model, the effect is mitigated for optimization problems that have many binary subproblems improving the current solution, such as the anti-ferromagnetic Potts model, and for optimization problems with competing interactions. Although the multivalued partition exhibits a better performance than the random partition for

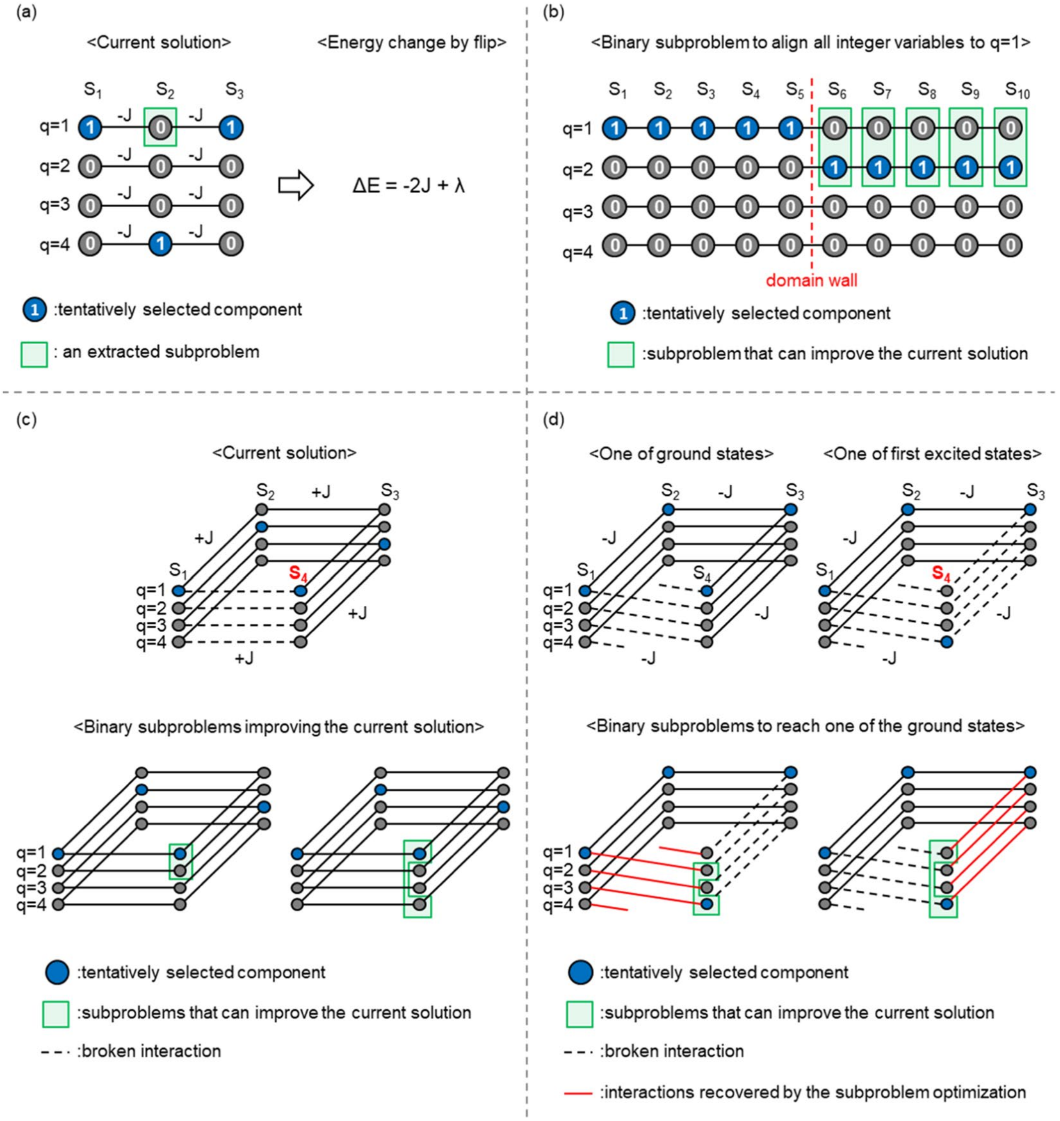
(a)
<Current solution>
<Energy change by flip>
S1 S2 S3
q=1 q=2 q=3 q=4
-J
ΔE = -2J + λ
:tentatively selected component
: an extracted subproblem
(b)
<Binary subproblem to align all integer variables to q=1>
S1 S2 S3 S4 S5 S6 S7 S8 S9 S10
q=1 q=2 q=3 q=4
domain wall
:tentatively selected component
:subproblem that can improve the current solution
(c)
<Current solution>
S1 S2 S3 S4
+J
q=1 q=2 q=3 q=4
<Binary subproblems improving the current solution>
:tentatively selected component
:subproblems that can improve the current solution
:broken interaction
(d)
<One of ground states>
<One of first excited states>
S1 S2 S3 S4
-J
q=1 q=2 q=3 q=4
<Binary subproblems to reach one of the ground states>
:tentatively selected component
:subproblems that can improve the current solution
:broken interaction
:interactions recovered by the subproblem optimization

Figure 6. Discussion on results. Vertices and edges represent binary variables $\{x_i^{(q)}\}$ and the interactions between them, respectively. Binary variables colored blue are tentatively selected components, and binary variables enclosed by green rectangles are the extracted subproblem. (**a**) Simple example of the random partition that can reduce the energy despite violating the one-hot constraint. (**b**) One of the first excited states commonly observed in the optimization of the ferromagnetic Potts model. To align all integer variables $\{S_i\}_{i=1,2,\ldots,10}$ to $q=1$, component $q=1$ must be selected for variables S_6, ..., S_{10}. (**c**) Local interactions in the anti-ferromagnetic Potts model. The two binary subproblems can improve the current solution. (**d**) Simple example of the Potts gauge glass model. One interaction is broken in the ground state due to competing interactions between different components. Two binary subproblems can improve the first excited state. Broken interactions are represented by dashed lines, while interactions recovered by optimizing the binary subproblem are represented by red lines.

Potts glass and Potts gauge glass models, we did not identify problems for which the multivalued partition is most suitable. Future studies should focus on constructing algorithms that can efficiently solve the ferromagnetic Potts model using the binary partition. In addition, the performance of the proposed methods should be assessed for various optimization problems, such as the graph coloring problem whose cost function is represented as the Hamiltonian of the anti-ferromagnetic Potts model.

Methods

Details on partitioning and embedding are provided in this section.

Subproblem-embedding algorithm. In this subsection, we briefly explain the subproblem-embedding algorithm developed in a previous study[41]. This algorithm aims to quickly find minor embeddings of subproblems to efficiently implement large-neighborhood local searches using the D-Wave quantum annealer.

Given a large optimization problem whose cost function is represented in QUBO form, this algorithm embeds binary variables one by one into the Chimera graph. After a randomly selected binary variable is embedded into the Chimera graph first, this algorithm embeds a binary variable interacting with the already embedded binary variables into the graph. The latter procedure is iterated until all qubits in the Chimera graph are used. It is generally necessary to assign several qubits to one binary variable and extend a chain to correctly represent interactions between binary variables on the Chimera graph. This algorithm implements Dijkstra's algorithm to greedily determine how to extend chains. Although chains often cannot be extended by only unused qubits, this difficulty can be avoided in the embedding of subproblems because it is not necessary to embed all binary variables. In this case, the algorithm stops attempting to embed the binary variable and attempts to embed other binary variables that can be easily embedded. Thus, extracting and embedding a subproblem is simultaneously implemented in this algorithm, and the resulting subproblem comprises only binary variables that can be easily embedded. Therefore, the computational time is significantly lower than in Cai's algorithm[50]. In addition, this algorithm can easily deal with hardware defects by implementing Dijkstra's algorithm on a Chimera graph with defects.

It is the random partition to directly apply this algorithm to extract and embed a subproblem of integer optimization problems because this algorithm does not address whether an extracted subproblem contains feasible solutions for each S_i or not. To combine the multivalued partition with this algorithm, the order of the binary variables embedded into the Chimera graph must be appropriately specified, as described in the following subsection.

Multivalued partition. The multivalued partition requires that the binary variable assigned to the tentatively selected component must be embedded into the Chimera graph. In addition, more than two binary variables should be embedded for each integer variable to distinguish the multivalued and binary partitions. On the other hand, the subproblem-embedding algorithm extracts and embeds a subproblem comprising binary variables that can be easily embedded. Therefore, in order to combine the multivalued partition and the subproblem-embedding algorithm, it is needed to appropriately specify the order of the binary variables embedded into the Chimera graph.

First, an integer variable is randomly selected, and binary variables assigned to the selected integer variable are embedded into the Chimera graph. Then, to determine binary variables to be additionally embedded into the Chimera graph, we select an integer variable as follows:

1. An already embedded binary variable x_{embedded} is selected in the order of being embedded into the Chimera graph.
2. An integer variable S_{ctr} to which x_{embedded} is assigned is selected.
3. Integer variables $\{S_i\}$ that interact with S_{ctr} in the problem graph are extracted.
4. An integer variable S_i is selected in a random order from $\{S_i\}$.
5. The binary variables $\{x_i^{(q)}\}_{q=1,2,\ldots,Q}$ assigned to S_i are attempted to be embedded.

Then, the order of the binary variables $\{x_i^{(q)}\}_{q=1,2,\ldots,Q}$ embedded into the Chimera graph is determined using the following two criteria:

1. The binary variable adjacent to the binary variables that are already embedded.
2. The binary variable assigned to the tentatively selected component.

For the binary variable that is embedded first among $\{x_i^{(q)}\}_{q=1,2,\ldots,Q}$, criterion 1 is important than criterion 2 to avoid embedding independent integer variables. For the reminder of the binary variables assigned to S_i, we prioritize criterion 2 to achieve the multivalued partition. It should be noted that the number of components embedded into the Chimera graph is not uniform for each integer variable because the subproblem-embedding algorithm embeds only binary variables that can be easily embedded. If only one component can be embedded, the integer variable is excluded from the subproblem.

Table 3 represents the average number $N_S(Q_{\text{embed}})$ of embedded integer variables with Q_{embed} components per subproblem. The performance is assessed for embedding the Potts gauge glass model on the cubic lattice into D-Wave 2000Q_2 with defects and is averaged over 1,000 trials. $65.7(=14.5+8.0+43.2)$ integer variables are embedded into the Chimera graph on average, and the average number of embedded binary variables is as follows:

$$\sum_{Q_{\text{embed}}=1}^{4} Q_{\text{embed}} N_S(Q_{\text{embed}}) = 2\times 14.5 + 3\times 8.0 + 4\times 43.2 = 225.8. \tag{11}$$

It should be noted that, to distinguish the multivalued and binary partitions, $Q_{\text{embed}} > 2$ is required for most integer variables. All four components are embedded for 65.8% of the integer variables in the subproblem, indicating that we can embed the multivalued subproblem that is distinct from the binary subproblem.

Binary partition. To solve large optimization problems using the binary partition, the cost function of the binary subproblem must be derived from the cost function of the original large problem. The general form of the local energy between S_i and S_j is given by

Q_{embed}	N_S (Q_{embed})
2	14.5
3	8.0
4	43.2

Table 3. Average number of embedded integer variables with Q_{embed} components.

	Complete graph embedding	Subproblem-embedding algorithm
Multivalued partition	9.6	33.9
Binary partition	19.3	122.8

Table 4. Average number of feasible solutions in an embedded subproblem: $\log_{10}N_{\text{feasible}}$.

$$\begin{aligned} H_{ij} \; = \; & \sum_{q=1}^{Q}\sum_{q'=1}^{Q} Q_{ij}^{(qq')} x_i^{(q)} x_j^{(q')} + \sum_{q=1}^{Q}\left(Q_{ii}^{(qq)} x_i^{(q)} + Q_{jj}^{(qq)} x_j^{(q)}\right) \\ & + \lambda\left(\sum_{q=1}^{Q} x_i^{(q)} - 1\right)^2 + \lambda\left(\sum_{q=1}^{Q} x_j^{(q)} - 1\right)^2, \end{aligned} \tag{12}$$

where $Q_{ij}^{(qq')}$ represents the interaction between $x_i^{(q)}$ and $x_j^{(q')}$. The binary partition extracts a binary subproblem by randomly selecting one component in addition to the tentatively selected component for each integer variable. The local energy of the binary subproblem in QUBO form is given as follows:

$$H_{ij}^{(\text{Binary})} = R_{ij} y_i y_j + R_{ii} y_i + R_{jj} y_j, \tag{13}$$

$$R_{ij} = Q_{ij}^{(\alpha_i\alpha_j)} - Q_{ij}^{(\alpha_i\beta_j)} - Q_{ij}^{(\beta_i\alpha_j)} + Q_{ij}^{(\beta_i\beta_j)}, \tag{14}$$

$$R_{ii} = \sum_{k\neq i}(Q_{ik}^{(\beta_i\alpha_k)} - Q_{ik}^{(\alpha_i\alpha_k)}) - Q_{ii}^{(\alpha_i\alpha_i)} + Q_{ii}^{(\beta_i\beta_i)}, \tag{15}$$

$$R_{jj} = \sum_{k\neq j}(Q_{kj}^{(\alpha_k\beta_j)} - Q_{kj}^{(\alpha_k\alpha_j)}) - Q_{jj}^{(\alpha_j\alpha_j)} + Q_{jj}^{(\beta_j\beta_j)}, \tag{16}$$

where $y_i \in \{0, 1\}$, α_i and β_i denote the tentatively selected component and randomly selected component for S_i, respectively, and $y_i = 0 (y_i = 1)$ indicates "stay in the tentatively selected component α_i" ("transit to the other component β_i")" It should be noted that the cost function of the binary subproblem does not contain the penalty term because all solutions in the binary subproblem satisfy the one-hot constraint.

The problem graph of the binary subproblem extracted from the three-dimensional Potts model is a cubic lattice with bond dilutions. The density of the interactions in the binary subproblem is lower than that in the multivalued subproblem because the cost function of the binary subproblem does not contain the penalty term, which generates partially fully connected interactions between $x_i^{(q)}$ and $x_i^{(q')}$. The average number of embedded binary variables is 408 when the binary partition is used, and only 225 when the multivalued partition is used. Furthermore, all configurations in the binary subproblem satisfy the one-hot constraint, while the configurations in the multivalued subproblem do not. Therefore, the average number N_{feasible} of feasible solutions involved in the embedded subproblem is considerably increased using the binary partition. Table 4 illustrates $\log_{10}N_{\text{feasible}}$ in a subproblem embedded by the multivalued and binary partitions combined with the complete graph embedding[44] and subproblem-embedding algorithm[41] for the Potts gauge glass model on the cubic lattice.

References

1. Lucas, A. Ising formulations of many np problems. *Front. Phys.* **2**, 5, https://doi.org/10.3389/fphy.2014.00005 (2014).
2. Karkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Sci.* **220**, 671–680, https://doi.org/10.1126/science.220.4598.671 (1983).
3. Kadowaki, T. & Nishimori, H. Quantum annealing in the transverse ising model. *Phys. Rev. E* **58**, 5355–5363, https://doi.org/10.1103/PhysRevE.58.5355 (1998).
4. Santoro, G. E., Martoňák, R., Tosatti, E. & Car, R. Theory of quantum annealing of an ising spin glass. *Sci.* **295**, 2427–2430, https://doi.org/10.1126/science.1068774 (2002).
5. Martoňák, R., Santoro, G. E. & Tosatti, E. Quantum annealing of traveling-salesman problem. *Phys. Rev. E* **70**, 057701, https://doi.org/10.1103/PhysRevE.70.057701 (2004).
6. Stella, L., Santoro, G. E. & Tosatti, E. Optimization by quantum annealing: Lessons from simple cases. *Phys. Rev. B* **72**, 014303, https://doi.org/10.1103/PhysRevB.72.014303 (2005).

7. Battaglia, D. A., Santoro, G. E. & Tosatti, E. Optimization by quantum annealing: Lessons from hard satisfiability problems. *Phys. Rev. E* **71**, 066707, https://doi.org/10.1103/PhysRevE.71.066707 (2005).
8. Zanca, T. & Santoro, G. E. Quantum annealing speedup over simulated annealing on random ising chains. *Phys. Rev. B* **93**, 224431, https://doi.org/10.1103/PhysRevB.93.224431 (2016).
9. Wauters, M. M., Fazio, R., Nishimori, H. & Santoro, G. E. Direct comparison of quantum and simulated annealing on a fully connected ising ferromagnet. *Phys. Rev. A* **96**, 022326, https://doi.org/10.1103/PhysRevA.96.022326 (2017).
10. Johnson, M. W. *et al.* Quantum annealing with manufactured spins. *Nat.* **473**, 194–198, https://doi.org/10.1038/nature10012 (2011).
11. Rønnow, T. F. *et al.* Defining and detecting quantum speedup. *science* **345**, 420–424, https://doi.org/10.1126/science.1252319 (2014).
12. Katzgraber, H. G., Hamze, F., Zhu, Z., Ochoa, A. J. & Munoz-Bauza, H. Seeking quantum speedup through spin glasses: The good, the bad, and the ugly. *Phys. Rev. X* **5**, 031026, https://doi.org/10.1103/PhysRevX.5.031026 (2015).
13. Denchev, V. S. *et al.* What is the computational value of finite range tunneling? *Phys. Rev. X* **6**, 031015, https://doi.org/10.1103/PhysRevX.6.031015 (2016).
14. Wang, C., Chen, H. & Jonckheere, E. Quantum versus simulated annealing in wireless interference network optimization. *Sci. Rep.* **6**, 25797, https://doi.org/10.1038/srep25797 (2016).
15. Rosenberg, G. *et al.* Solving the optimal trading trajectory problem using a quantum annealer. *IEEE J. Sel. Top. Signal Process.* **10**, 1053–1060, https://doi.org/10.1109/JSTSP.2016.2574703 (2016).
16. Boyda, E. *et al.* Deploying a quantum annealing processor to detect tree cover in aerial imagery of california. *PLoS ONE* **12**((2)), e0172505, https://doi.org/10.1371/journal.pone.0172505 (2017).
17. O'Malley, D., Vesselinov, V. V., Alexandrov, B. S. & Alexandrov, L. B. Nonnegative/binary matrix factorization with a d-wave quantum annealer. Preprint at https://arxiv.org/abs/1704.01605 (2017).
18. Neukart, F. *et al.* Traffic flow optimization using a quantum annealer. *Front. ICT* **4**, 29, https://doi.org/10.3389/fict.2017.00029 (2017).
19. Baldassi, C. & Zecchina, R. Efficiency of quantum vs. classical annealing in nonconvex learning problems. *Proc. Natl. Acad. Sci.* **115**, 1457–1462, https://doi.org/10.1073/pnas.1711456115 (2018).
20. Yarkoni, S., Plaat, A. & Back, T. First results solving arbitrarily structured maximum independent set problems using quantum annealing. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, 1–6, https://doi.org/10.1109/CEC.2018.8477865 (2018).
21. Adachi, S. H. & Henderson, M. P. Application of quantum annealing to training of deep neural networks. Preprint at https://arxiv.org/abs/1510.06356 (2015).
22. Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B. & Melko, R. Quantum boltzmann machine. *Phys. Rev. X* **8**, 021050, https://doi.org/10.1103/PhysRevX.8.021050 (2018).
23. Benedetti, M., Realpe-Gómez, J., Biswas, R. & Perdomo-Ortiz, A. Quantum-assisted learning of hardware-embedded probabilistic graphical models. *Phys. Rev. X* **7**, 041052, https://doi.org/10.1103/PhysRevX.7.041052 (2017).
24. Harris, R. *et al.* Phase transitions in a programmable quantum spin glass simulator. *Sci.* **361**, 162–165, https://doi.org/10.1126/science.aat2025 (2018).
25. King, A. D. *et al.* Observation of topological phenomena in a programmable lattice of 1,800 qubits. *Nat.* **560**, 456–460, https://doi.org/10.1038/s41586-018-0410-x (2018).
26. Streif, M., Neukart, F. & Leib, M. Solving quantum chemistry problems with a d-wave quantum annealer. Preprint at https://arxiv.org/abs/1811.05256 (2018).
27. Ohzeki, M., Miki, A., Miyama, M. J. & Terabe, M. *Control of automated guided vehicles without collision by quantum annealer and digital devices*. Preprint at https://arxiv.org/abs/1812.01532 (2018).
28. Kitai, K. *et al.* Expanding the horizon of automated metamaterials discovery via quantum annealing. Preprint at https://arxiv.org/abs/1902.06573 (2019).
29. Irie, H., Wongpaisarnsin, G., Terabe, M., Miki, A. & Taguchi, S. Quantum annealing of vehicle routing problem with time, state and capacity. In *Quantum Technology and Optimization Problems* (2019).
30. Nishimura, N., Tanahashi, K., Suganuma, K., Miyama, M. J. & Ohzeki, M. Item listing optimization for e-commerce websites based on diversity. Preprint at https://arxiv.org/abs/1903.12478 (2019).
31. Morita, S. & Nishimori, H. Mathematical foundation of quantum annealing. *J. Math. Phys.* **49**, 125210, https://doi.org/10.1063/1.2995837 (2008).
32. Bunyk, P. I. *et al.* Architectural considerations in the design of a superconducting quantum annealing processor. IEEE Transactions. *Appl. Supercond.* **24**, 1700110, https://doi.org/10.1109/TASC.2014.2318294 (2014).
33. Booth, M., Reinhardt, S. P. & Roy, A. Partitioning optimization problems for hybrid classcal/quantum execution. http://www.dwavesys.com/sites/default/files/partitioning_QUBOs_for_quantum_acceleration-2.pdf (2017).
34. Rosenberg, G. *et al.* Building an iterative heuristic solver for a quantum annealer. *Comput. Optim Appl* **65**, 845, https://doi.org/10.1007/s10589-016-9844-y (2016).
35. Narimani, A., Saeed, S. S., Changiz, R & Zaribafiyan, A. Combinatorial optimization by decomposition on hybrid cpu–non-cpu solver architectures. Preprint at https://arxiv.org/abs/1708.03439 (2017).
36. Ahuja, R. K., Ergun, O., Orlin, J. B. & Punnen, A. P. A survey of very large-scale neighborhood search techniques. *Discret. Appl. Math.* **123**, 75–102, https://doi.org/10.1016/S0166-218X(01)00338-9 (2002).
37. Hamze, F. & Freitas, N. D. From fields to trees. In *The 20th conference on Uncertainty in Artificial Intelligence* (AUAI Press, Arlington, Virginia, 2004), 243–250 (2004).
38. Fix, A., Chen, J., Boros, E. & Zabih, R. Approximate mrf inference using bounded treewidth subgraphs. *In Computer Vision – ECCV* **2012**, 385–398, https://doi.org/10.1007/978-3-642-33718-5_28 (2012).
39. Decelle, A. & Krzakala, F. Belief-propagation-guided monte-carlo sampling. *Phys. Rev. B* **89**, 214421, https://doi.org/10.1103/PhysRevB.89.214421 (2014).
40. Selby, A. Efficient subgraph-based sampling of ising-type models with frustration. Preprint at https://arxiv.org/abs/1409.3934 (2014).
41. Okada, S., Ohzeki, M., Terabe, M. & Taguchi, S. Improving solutions by embedding larger subproblems in a d-wave quantum annealer. *Sci. Rep.* **9**, 2098, https://doi.org/10.1038/s41598-018-38388-4 (2019).
42. Choi, V. Minor-embedding in adiabatic quantum computation: Ii. minor-univerdal graph design. *Quantum Inf. Process.* **10**, 343–353, https://doi.org/10.1007/s11128-010-0200-3 (2011).
43. Klymko, C., Sullivan, B. D. & Humble, T. S. Adiabatic quantum programming: Minor embedding with hard faults. *Quantum Inf Process.* **13**, 709, https://doi.org/10.1007/s11128-013-0683-9 (2014).
44. Boothby, T., King, A. D. & Roy, A. Fast clique minor generation in chimera qubit connectivity graphs. *Quantum Inf Process.* **15**, 495, https://doi.org/10.1007/s11128-015-1150-6 (2016).
45. Wu, F. Y. The potts model. *Rev. Mod. Phys.* **54**, 235–268, https://doi.org/10.1103/RevModPhys.54.235 (1982).
46. Okada, S., Ohzeki, M. & Tanaka, K. The efficient quantum and simulated annealing of potts models using a half-hot constraint. Preprint at https://arxiv.org/abs/1904.01522 (2019).
47. Gross, D. J., Kanter, I. & Sompolinsky, H. Mean-field theory of the potts glass. *Phys. Rev. Lett.* **55**, 304, https://doi.org/10.1103/PhysRevLett.55.304 (1985).
48. Nishimori, H. & Stephen, M. J. Gauge-invariant frustrated potts spin-glass. *Phys. Rev. B* **27**, 5644–5652, https://doi.org/10.1103/PhysRevB.27.5644 (1983).

49. Çağlar, T. & Berker, A. N. Chiral potts spin glass in d = 2 and 3 dimensions. *Phys. Rev. E* **94**, 032121, https://doi.org/10.1103/PhysRevE.94.032121 (2016).
50. Cai, J., Macready, B. & Roy, A. A practical heuristic for finding graph minors. Preprint at https://arxiv.org/abs/1406.2741 (2014).

Acknowledgements

The authors are deeply grateful to Shu Tanaka, Masamichi J. Miyama and Tadashi Kadowaki for fruitful discussions. The author M. O. is grateful for the financial support provided by JSPS KAKENHI 19H01095 and 16H04382, Next Generation High-Performance Computing Infrastructures and Applications R&D Program by MEXT.

Author Contributions

S.O. conceived and developed the concept and conducted all the experiments. M.O. proposed the plan to evaluate the validity of the concept, discussed the details of the results, and reviewed the manuscript. S.T. directed the project in our study.

Additional Information

Competing Interests: The authors declare no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.