SCIENTIFIC REPORTS

OPEN

Improving solutions by embedding larger subproblems in a D-Wave quantum annealer

Received: 7 November 2018
Accepted: 27 December 2018
Published online: 14 February 2019

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

Quantum annealing is a heuristic algorithm that solves combinatorial optimization problems, and D-Wave Systems Inc. has developed hardware implementation of this algorithm. However, in general, we cannot embed all the logical variables of a large-scale problem, since the number of available qubits is limited. In order to handle a large problem, qbsolv has been proposed as a method for partitioning the original large problem into subproblems that are embeddable in the D-Wave quantum annealer, and it then iteratively optimizes the subproblems using the quantum annealer. Multiple logical variables in the subproblem are simultaneously updated in this iterative solver, and using this approach we expect to obtain better solutions than can be obtained by conventional local search algorithms. Although embedding of large subproblems is essential for improving the accuracy of solutions in this scheme, the size of the subproblems are small in qbsolv since the subproblems are basically embedded by using an embedding of a complete graph even for sparse problem graphs. This means that the resource of the D-Wave quantum annealer is not exploited efficiently. In this paper, we propose a fast algorithm for embedding larger subproblems, and we show that better solutions are obtained efficiently by embedding larger subproblems.

Combinatorial optimization problems, the minimization of cost functions with discrete variables, have significant real-world applications. The cost function of a combinatorial optimization problem can generally be mapped to the Hamiltonian of a classical Ising model[1]. Inspired by simulated annealing[2], quantum annealing (QA)[3] was proposed as a method for searching the ground state of a Hamiltonian with a complicated energy landscape. While SA employs thermal fluctuations to escape local minima, QA utilizes quantum fluctuations. Numerous studies have investigated whether QA outperforms SA in terms of the computational time required to obtain a high-accuracy solution. Most of the studies have shown that QA is superior to SA[4–6], while a few have also suggested that it is inferior[7]. Recently, a commercial quantum annealer based on superconducting flux qubits[8] has been developed by D-Wave Systems Inc. Experimental studies using the D-Wave quantum annealer have been performed to compare the performance of QA and SA[6,9] and to demonstrate the applicability of the D-Wave quantum annealer to practical problems[10–12].

The generic form of a time-dependent Hamiltonian in QA is

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

where $\hat{H}_0$ is the classical Hamiltonian which represents the cost function to be minimized, and $\hat{H}_{\mathrm{q}}$ is the quantum fluctuation term for which the ground state is trivial. At the beginning of QA, the coefficients of the time-dependent Hamiltonian are set to $A(0) = 1$, $B(0) = 0$, and the system is in trivial ground state determined by $\hat{H}_{\mathrm{q}}$. At the end of QA, the coefficients are set to $A(\tau) = 0$ and $B(\tau) = 1$ where τ is the annealing time. The system evolves according to the Schrödinger equation:

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

[1]Electronics R & I Division, DENSO Corporation, Tokyo, 103-6015, 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. Correspondence and requests for materials should be addressed to S.O. (email: okada@denso.co.jp)

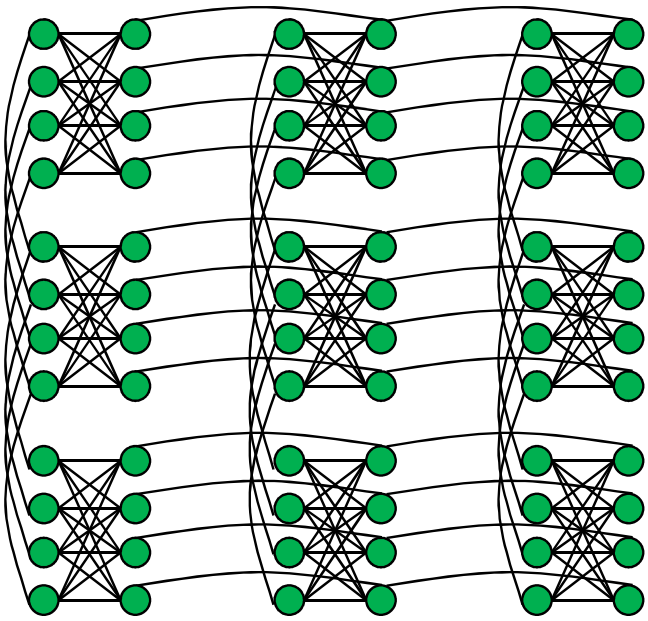

Figure 1. A Chimera graph for $(M, N, L) = (3, 3, 4)$. The nine complete bipartite graphs $K_{4,4}$ are arranged in a grid pattern.

where $|\psi(t)\rangle$ is a state vector of the system and $\hbar$ is set to 1 for simplicity. The system will remain close to the instantaneous ground state of the time-dependent Hamiltonian if the system changes sufficiently slowly and if the adiabatic condition[13],

$$\frac{1}{[\varepsilon_1(t) - \varepsilon_0(t)]^2}\left|\langle 1(t)|\frac{d\hat{H}(t)}{dt}|0(t)\rangle\right| \ll 1, \tag{3}$$

is always satisfied during QA. Here $|0(t)\rangle$, $|1(t)\rangle$, $\varepsilon_0(t)$ and $\varepsilon_1(t)$ are eigenvectors and eigenvalues of the instantaneous ground state and first excited state, respectively. Thus, by setting the annealing time τ large enough, we ultimately obtain the ground state of the classical Hamiltonian $\hat{H}_0$, which represents the optimal solution.

The current version of D-Wave quantum annealer (D-Wave 2000Q) implements QA with a transverse magnetic field:

$$\hat{H}_q = -\sum_{i=1}^{N_q}\hat{\sigma}_i^{(x)}, \tag{4}$$

where N_q represents the total number of qubits. A cost function that the D-Wave quantum annealer can handle is:

$$\hat{H}_0 = \sum_{(i,j)\in \text{chimera}}^{N_q} J_{ij}\hat{\sigma}_i^{(z)}\hat{\sigma}_j^{(z)} + \sum_{i=1}^{N_q} h_i\hat{\sigma}_i^{(z)}, \tag{5}$$

where the interactions between qubits are restricted to Chimera graph, which is constructed as an $M \times N$ grid of complete bipartite graphs $K_{L,L}$[14]. Chimera graph for $(M, N, L) = (3, 3, 4)$ is shown in Fig. 1, where the vertices and edges represent qubits and the interactions between them, respectively. Although the Chimera graph for D-Wave 2000Q is $(M, N, L) = (16, 16, 4)$, the number of operable qubits is less than $N_q = 2MNL = 2048$, since there are defects in the qubits and connectivities.

Limited connectivity between the qubits is a restriction to employing the D-Wave quantum annealer for real-world applications. Before solving an optimization problem, it is necessary to map a problem graph onto a subgraph of the hardware graph. This process is called minor embedding. The problem graph is defined as a graph in which the vertices and edges represent the logical variables and interactions between them, respectively. The hardware graph is defined as a graph for which the vertices and edges represent the qubits and interactions between them, respectively. It is known to be NP-hard to find minor embeddings if both of the problem graph and hardware graph are the part of input, and polynomial time if the problem graph or hardware graph is fixed[15]. There exist various algorithms to find the minor embeddings, and a heuristic algorithm proposed by Cai *et al.*[16] is the most versatile option so far. While this general algorithm searches for a minor embedding of an arbitrary problem graph into an arbitrary hardware graph, the computational time increases drastically with the number of qubits, especially for sparse problem graphs. To reduce the computational time for the minor embedding, some algorithms that exploit features of the hardware graphs and specific problem graphs have been developed. Although the number of logical variables embeddable into hardware graphs is small, utilizing complete graph

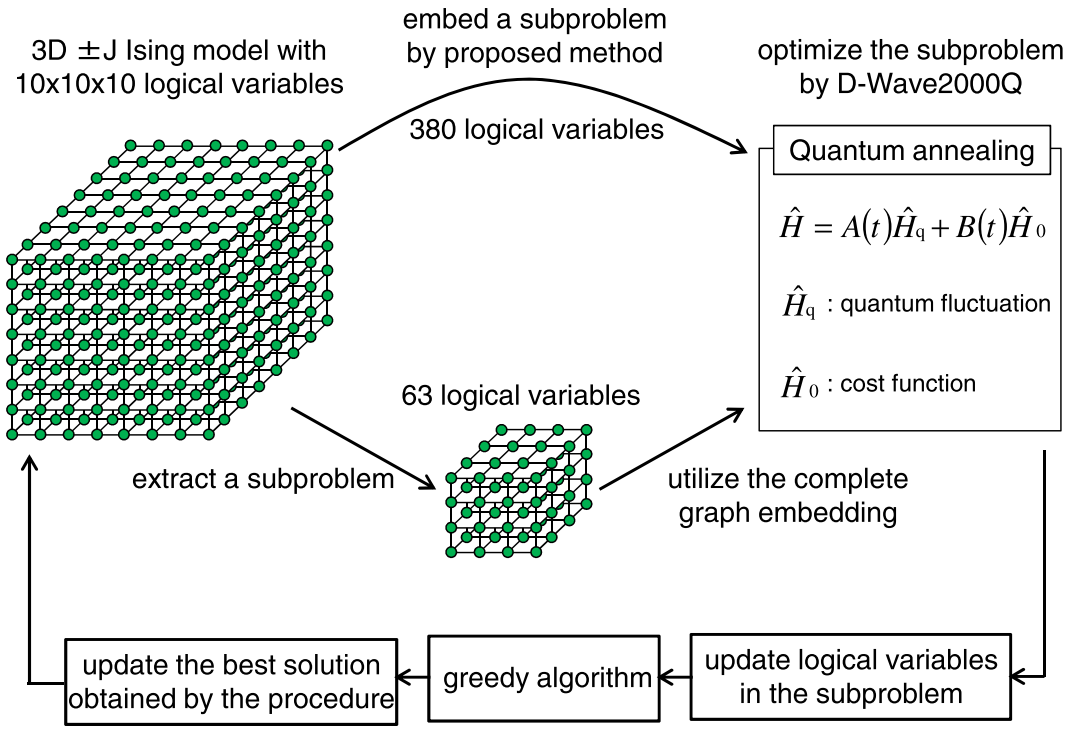

Figure 2. The optimization process implemented in this study. The solutions obtained by utilizing the proposed algorithm and by using complete-graph embedding are compared.

embedding[17,18] is the simplest way to reduce the computation time. Complete graph embedding can be applied to arbitrary problem graphs with up to 64 logical variables for a Chimera graph with $(M, N, L) = (16, 16, 4)$ without defects. This is basically the embedding used in qbsolv[19]. Other embedding algorithms[20] can find the minor embedding efficiently in reasonably dense problem graphs by exploiting the bipartite structure of the Chimera graph[21], and it is possible to embed a larger number of logical variables than for a complete graph embedding[17,18]. A minor embedding of the Cartesian product of two complete graphs, which often appears in real-world optimization problems, has been proposed in the literature[22]. In addition, efficient minor embeddings for the cubic lattice[23] and two-dimensional square-octagonal and triangular lattices[24] are also proposed. However, efficient embedding algorithms for arbitrary sparse problem graphs do not exist, despite the fact that sophisticated minor embeddings for sparse problem graphs are more important than for dense problem graphs in order to exploit the potential of the D-Wave quantum annealer. More logical variables can be embedded with shorter-length chains for sparse problem graphs.

In the methods section below, we propose a fast algorithm for embedding larger subproblems based on Cai's algorithm[16]. We do not need to embed all of the logical variables of a problem graph in embedding of subproblems, and the logical variables that can be embedded easily are selected as a part of the subproblem in our proposed algorithm. As a result, the proposed algorithm can embed larger subproblems than complete graph embedding[17,18], with shorter computational time than the Cai's algorithm[16], not only for dense problem graphs but also for sparse problem graphs.

In the following section, we show the improvement in solutions achieved by embedding larger subproblems for a ferromagnetic model and a spin-glass model on a three-dimensional $\pm J$ Ising model with 1,000 logical variables. Since the cubic lattice with 1,000 variables is not embeddable into D-Wave 2000Q, we extract embeddable subproblems into D-Wave 2000Q and iteratively optimize them using a quantum annealer like {\ttqbsolv}. In this study, we utilized two algorithms to embed subproblems into the Chimera graph of D-Wave 2000Q, with few defects in the qubits and connectivities. One is the proposed algorithm, which can embed 380 logical variables, and the other is complete graph embedding[18], which can embed only 63 logical variables. We have confirmed that better solutions can be achieved with less number of iterations for both the ferromagnetic and spin-glass models by embedding large subproblems.

Results

In this section, we demonstrate that better solutions are obtained efficiently by embedding larger subproblems for the ferromagnetic and spin-glass models on the three-dimensional $\pm J$ Ising model with 1,000 logical variables

The optimization process implemented in this study is shown in Fig. 2. The problem graph is the cubic lattice with $10 \times 10 \times 10$ logical variables. We partition the original large problem into subproblems and then iteratively optimize the subproblems using a quantum annealer. Two algorithms are utilized to embed subproblems in this study. One is the proposed algorithm, which can embed 380 logical variables, and the other is a complete-graph embedding[18], which can embed only 63 logical variables into the Chimera graph of the D-Wave 2000Q with few

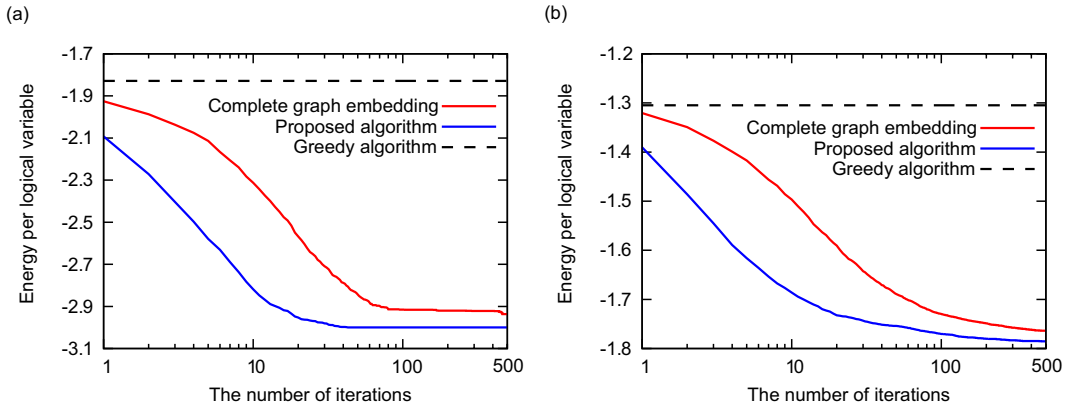

Figure 3. Comparisons of the average energy from 32 trials. The dashed lines represent the average energies of local minima attained by the greedy algorithm. (**a**) The ferromagnetic model with $p_F = 0.0$. (**b**) The spin-glass model with $p_F = 0.5$.

defects in the qubits and connectivities. After embedding and optimizing the subproblem, the logical variables of the subproblem are updated in the output of D-Wave 2000Q. Then, a greedy algorithm is executed by a conventional digital computer to get to an exact (local) minimum. In this greedy algorithm, one logical variable is randomly selected, and it is flipped if the energy decreases. We finish refining the solution using the greedy algorithm if the energy change caused by flipping each logical variable is completely non-negative. Finally, the best solution obtained by this procedure is updated. These processes are iterated, and we confirm that the solutions are improved by embedding larger subproblems.

The Hamiltonian optimized in this study is shown below:

$$H_0(\{x\}) = \sum_{<i,j>} J_{ij} x_i x_j, \tag{6}$$

$$p(J_{ij}) = p_F \delta(J_{ij} - J) + (1 - p_F)\delta(J_{ij} + J), \tag{7}$$

where $x_i \in (-1, +1)$ represents a logical variable, J_{ij} is the interaction between nearest neighbors in the cubic lattice, and p_F is the probability that $J_{ij} = +J$, the anti-ferromagnetic interaction. We evaluated solutions for a ferromagnetic model with $p_F = 0.0$ and a spin-glass model with $p_F = 0.5$[23]. The ferromagnetic model has no frustration, so that $x_1 = x_2 = \cdots = x_{1000} = -1$ and $+1$ are the trivial ground states. However, it is often the case that logical variables are divided into two kinds of domains, with the logical variables equal to $+1$ in one domain and -1 in the other domain. The boundaries of the domains are called domain walls, and it is essential to eliminate domain walls to obtain the ground state of ferromagnetic model. While domain walls cannot be eliminated efficiently by single-spin-flip algorithms such as simulated annealing, cluster Monte Carlo algorithms[25] address domain walls well by flipping logical variables in the same domain simultaneously. Although the structures of clusters in these algorithms[25] are different from those of the subproblems extracted by the proposed algorithm, we expect that domain walls can be eliminated efficiently by embedding larger subproblems. In the spin-glass model with $p_F = 0.5$, there are many frustrations, and the energy landscape is rugged with many local minima. In order to obtain better solutions, it is essential to search for as many local minima as possible. By embedding larger subproblems, the phase space searched by optimizing the subproblem grows exponentially, and it is possible to search for better local minima that could be distant from the current solution in the phase space. As a result, we expect that better solutions can be obtained efficiently by embedding larger subproblems for both the ferromagnetic and the spin-glass models.

The energies obtained for $p_F = 0.0$ and $p_F = 0.5$ are shown in Fig. 3(a and b), respectively. The average energies for 32 trials are plotted, and the same initial states are used for each run [$p_F = 0.0$ and 0.5, with the two embedding algorithms illustrated in Fig. 2]. For both $p_F = 0.0$ and $p_F = 0.5$, lower energies are obtained with a smaller number of iterations by embedding larger subproblems. The ground state energy for the ferromagnetic model with $p_F = 0.0$ is -3 and the ground state is obtained for all the trials after 45 iterations by using the proposed algorithm. For the spin-glass model with $p_F = 0.5$, the average energy obtained by 500 iterations of the complete graph embedding can be achieved by 75 iteragions of the proposed algorithm. The dashed lines represent the average energies of local minima attained by the greedy algorithm which is explained in this section, from the same 32 initial states. These results imply that the D-Wave quantum annealer is useful to search for better local minima of large optimization problems and embedding larger subproblems is effective in achieving high-accurate solutions.

Discussion

In the present paper, we showed that better solutions are obtained efficiently by embedding larger subproblems for the spin-glass and ferromagnetic models on the cubic lattice with $10 \times 10 \times 10$ logical variables. The energy landscape of the spin-glass model is rugged with many local minima. It is essential to search for as many local minima as possible, and this can be achieved by embedding larger subproblems, for which the phase space is

exponentially larger than that of small subproblems. For the ferromagnetic model, although there are no frustrations and a trivial ground state exists, eliminating the domain walls from which single-spin-flip algorithms suffer is essential to obtain the ground state. The logical variables in small domains can be flipped simultaneously by embedding larger subproblems, and as a result the ground state can be obtained efficiently with a smaller number of iterations. Although we demonstrated the improvements in the solutions specifically for the spin-glass and ferromagnetic models on a cubic lattice, we expect that better solutions can be obtained efficiently for a wide range of optimization problems by embedding larger subproblems.

For practical applications, it is essential to utilize the D-Wave quantum annealer as a part of an iterative solver like qbsolv, as long as the problem size embeddable in the D-Wave quantum annealer remains limited. A hybrid use of classical algorithms and the D-Wave quantum annealer is inevitable for this scheme. Although we simply adopted a greedy algorithm as a classical optimization algorithm, a myriad of classical algorithms can be combined with the D-Wave quantum annealer[19,26,27]. One guideline for selecting a classical solver is to exploit the complementary advantages of QA and classical algorithms[19]. For example, a more versatile optimization algorithm may be constructed by combining QA and SA[28], since QA performs well for the energy landscape with many high and thin barriers, while SA efficiently explores the phase space with low and wide barriers[29].

Although QA was initially proposed as an optimization method, the D-Wave quantum annealer has recently been considered as a sampling machine. It has been assumed that the output of D-Wave quantum annealer is close to a Boltzmann distribution of the Hamiltonian at a freeze-out point during annealing[30], and applications that utilize the quantum annealer as a sampling machine have been reported[31–34]. In addition, a local search around a specific initial state using the D-Wave quantum annealer has been proposed in the literature[28] and it is implemented in D-Wave 2000Q. This is called reverse annealing[35]. By combining reverse annealing and the embedding algorithm proposed in this paper, it may be possible to implement Markov chain Monte Carlo methods efficiently for large problems.

In future work, we will compare solutions obtained by the proposed algorithm with those obtained using the best-known embedding for the cubic lattice[23], and evaluate the utility of embedding larger subproblems for various optimization problems. Furthermore, we will construct high-performance optimization algorithms that exploit the proposed embedding algorithm.

Methods

In this section, we describe a fast algorithm for embedding larger subproblems into a hardware graph.

Definition of minor embedding. In general, a problem graph is not a subgraph of a Chimera graph, and the problem graph must be mapped onto a subgraph of a Chimera graph in order to solve optimization problems using the D-Wave quantum annealer. This process is called minor embedding of the problem graph into the hardware graph, and this is achieved by representing one logical variable with multiple qubits. For example, more than two qubits must be assigned to represent a logical variable that interacts with ten logical variables, since the maximum degree of a Chimera graph is six. If the two qubits $\hat{\sigma}_1^{(z)}$ and $\hat{\sigma}_2^{(z)}$ are used to represent the same logical variable, $\hat{\sigma}_1^{(z)}$ and $\hat{\sigma}_2^{(z)}$ must be connected on Chimera graph. The local energy related to $\hat{\sigma}_1^{(z)}$ and $\hat{\sigma}_2^{(z)}$ is denoted as $J_{12}\hat{\sigma}_1^{(z)}\hat{\sigma}_2^{(z)}$, and we can set the local energy of $\hat{\sigma}_1^{(z)} = \hat{\sigma}_2^{(z)}$ lower than that of $\hat{\sigma}_1^{(z)} = -\hat{\sigma}_2^{(z)}$ by setting $J_{12} < 0$. If $|J_{12}|$ is large enough, the optimal solutions of the embedded problem will be identical to that of the original optimization problem. Note that the assignment of multiple qubits to one logical variable is allowed, while the assignment of multiple logical variables to one qubit is forbidden. As shown in the literature[17], a minor embedding of a problem graph G_p into a hardware graph G_q is defined as follows:

1. Each vertex v in V_p is mapped to the vertex set of a connected subtree T_v of G_q.
2. If $(u, v) \in E_p$, then there exist $i_u, i_v \in V_q$ such that $i_u \in T_u$, $i_v \in T_v$, and $(i_u, i_v) \in E_q$.

A connected subtree T_v is often called a chain.

A conventional heuristic algorithm. A conventional heuristic algorithm for finding a minor embedding of an arbitrary problem graph into an arbitrary hardware graph has been proposed by Cai *et al.* in the literature[16]. The embedding process of this algorithm is divided into two stages. In the initial stage, logical variables are embedded one by one into the hardware graph, and all of the logical variables are embedded by allowing multiple assignments of the logical variables to one qubit. For example, suppose that the logical variables x_1, ..., x_k are already embedded in the hardware graph, and a logical variable x_{add} that is adjacent to x_1,..., x_k in the problem graph is selected to be additionally embedded. In this case, as shown in Fig. 4(a), an unused qubit to which no logical variables are assigned is selected as the root of x_{add}, and the shortest paths from the root of x_{add} to T_{x_1},..., T_{x_k} are calculated on the hardware graph using Dijkstra's algorithm. Then, by assigning x_{add} or x_i ($i = 1, 2, \ldots, k$) to qubits in the shortest paths, the adjacency between x_{add} and $x_1, \ldots, x_k$ will be represented on the hardware graph. However, it will often be the case that a path with only unused qubits does not exist. For example, as shown in Fig. 4(b), if a logical variable z is assigned to all of the qubits adjacent to T_{x_1}, a path from the root of x_{add} to T_{x_1} with only unused qubits does not exist. In such a case, by assigning multiple logical variables [x_{add} or x_1 and z in Fig. 4(b)] to one qubit, x_{add} will be embedded once. After embedding all of the logical variables by allowing multiple assignments in the initial stage, the minor embedding obtained in the initial stage is refined so that only one logical variable is assigned to one qubit in the last stage.

The computational time for this algorithm is dominated by Dijkstra's algorithm. The computational time $T_{conv}^{(1)}$ and the number $N_{Dijkstra}^{(1)}$ of shortest paths searched by Dijkstra's algorithm in the initial stage are given by

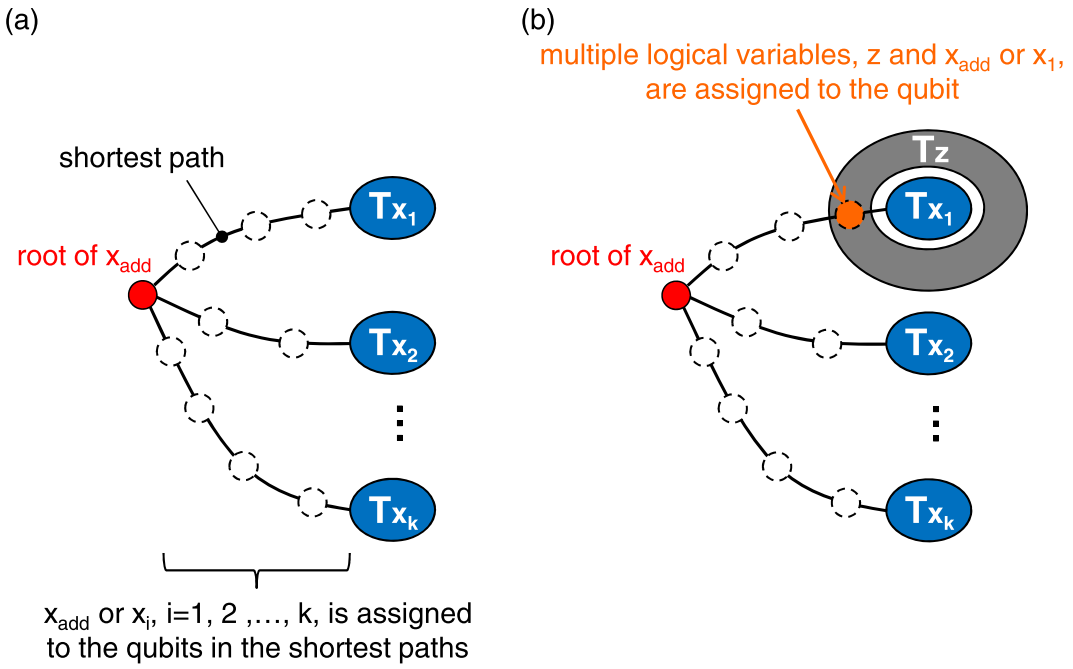

Figure 4. An embedding of a logical variable x_{add}. (**a**) A case for which paths to the adjacent logical variables exist. (**b**) A case for which multiple assignments of logical variables is necessary.

$$T_{conv}^{(1)} \sim O\left(N_{Dijkstra}^{(1)} T_{Dijkstra}\right), \tag{8}$$

$$N_{Dijkstra}^{(1)} \sim O\left(|E_p|\right), \tag{9}$$

and $T_{conv}^{(2)}$ and $N_{Dijkstra}^{(2)}$ in the last stage are given by

$$T_{conv}^{(2)} \sim O\left(N_{Dijkstra}^{(2)} T_{Dijkstra}\right), \tag{10}$$

$$N_{Dijkstra}^{(2)} \sim O\left(|V_p||V_q||E_p|\right), \tag{11}$$

where $T_{Dijkstra}$ represents the computational time for Dijkstra's algorithm:

$$T_{Dijkstra} \sim O\left(|E_q| + |V_q| \log |V_q|\right). \tag{12}$$

Here, $|V_p|$ and $|E_p|$ are the number of vertices and edges in the problem graph, and $|V_q|$ and $|E_q|$ are the number of vertices and edges in the hardware graph, respectively. In this algorithm, the vertex-weighted shortest paths are searched in order to distinguish used and unused qubits. The computational time in the last stage is obviously dominant. So we expect that the computational time will be drastically reduced by avoiding multiple assignments of logical variables in the initial stage, since the implementation of the last stage becomes unnecessary.

Proposed algorithm. Here we focus on the embedding of subproblems and propose a fast algorithm to find minor embeddings of subproblems. In embedding a subproblem, we can select logical variables that are embeddable without multiple assignments as a part of the subproblem, since it is not necessary to embed all of the logical variables included in the problem graph. While in a conventional algorithm, the search for a minor embedding is subject to a strong restriction, that all of the logical variables included in the problem graph must be embedded. The multiple assignments of logical variables are mainly caused by this restriction.

However, for dense problem graphs, the logical variables embeddable without multiple assignments become extinct before all the qubits are exploited. To mitigate this issue, the proposed algorithm includes a reservation system that leaves space to extend the chains. As shown in Fig. 5(a), if a qubit on the left side of a complete bipartite graph $K_{4,4}$ in Chimera graph is selected as the root of x_{add}, qubits to extend the chain vertically are reserved by x_{add}, and assignment of other logical variables to these qubits are forbidden. If a qubit on the right side is selected as the root of x_{add}, qubits to extend the chain horizontally are reserved by x_{add}, as shown in Fig. 5(b). The reserved qubits are released after the embedding of all the logical variables adjacent to x_{add} are completed. The pseudocode of the proposed algorithm is shown in Algorithm 1.

Algorithm 1. Proposed algorithm.

while Unused qubits exist in the hardware graph **do**
  Select x_{add} (Fig. 4)
  Calculate shortest paths to the adjacent variables (Fig. 4)
  if multiple assignments are not necessary [Fig. 4(a)] **then**
    Qubits associated to the root are reserved by x_{add} (Fig. 5)
    Logical variables are assigned to the qubits in the shortest paths
  else
    Drop x_{add} from the subproblem
  end if
end while

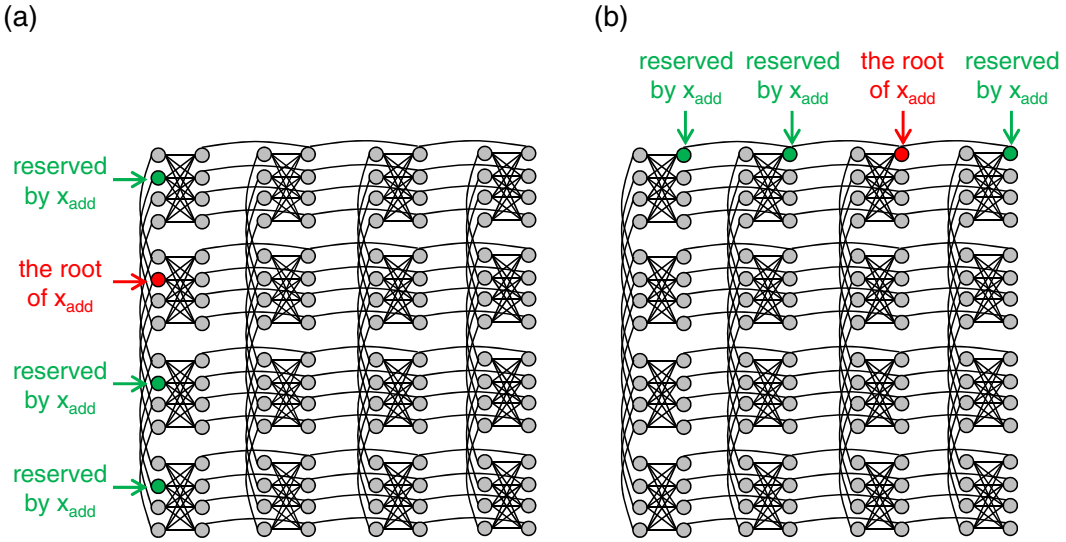

Figure 5. Examples of reserved qubits associated to the root of x_{add}. (**a**) An example showing vertically reserved qubits associated to the root of x_{add}. (**b**) An example showing horizontally reserved qubits associated to the root of x_{add}.

The refinement of the embedding in the last stage of the conventional algorithm can be eliminated in the proposed algorithm. In addition, the breadth-first search in a subgraph of a hardware graph consisting only of unused qubits is sufficient to search the shortest paths, since multiple assignments are forbidden. The computational time T_{prop} for the proposed algorithm and the number N_{breadth} of the shortest paths searched by the breadth-first search are given by

$$T_{\mathrm{prop}} \sim O(N_{\mathrm{breadth}} T_{\mathrm{breadth}}), \tag{13}$$

$$N_{\mathrm{breadth}} \sim |E_{\mathrm{p}}|. \tag{14}$$

The computational time T_{breadth} for the breadth-first search is given by

$$T_{\mathrm{breadth}} \sim O\left(|e_{\mathrm{q}}|\right), \tag{15}$$

where $|e_{\mathrm{q}}|$ is the number of edges included in the subgraph of the hardware graph with unused qubits. For Chimera graph, as shown in Fig. 6, it is sufficient to consider unit cells with used qubits and adjacent unit cells without used qubits. The maximum number of edges included in the subgraph is limited to

$$|e_{\mathrm{q}}| \sim O\left(|V_{\mathrm{q}}|^{1/2} L^{3/2}\right). \tag{16}$$

The embedding algorithm proposed in this section does not strongly depend on the structure of the hardware graphs, except for the reservation system, and we can easily adapt the embedding algorithm for other hardware graphs.

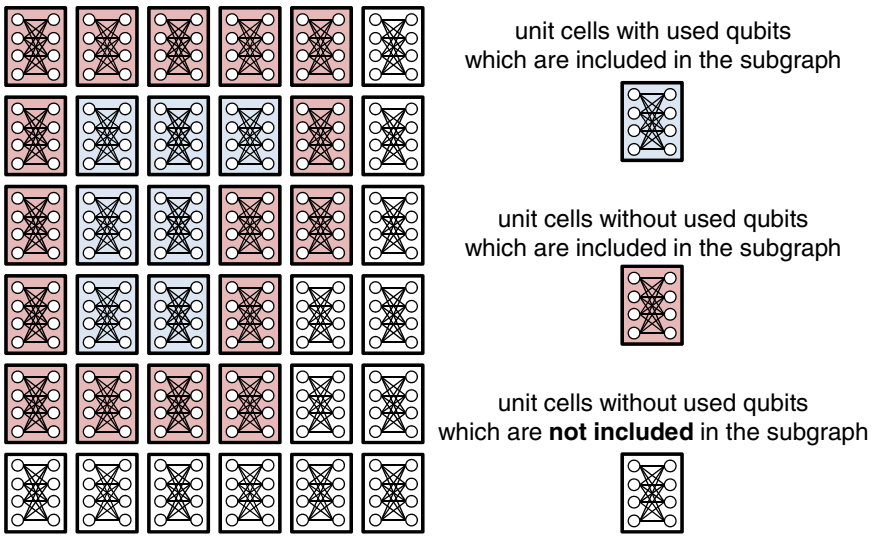

Figure 6. An example of the unit cells included in a subgraph. The unit cells colored blue and red are included. The number of qubits in the subgraph is dominated by the qubits in the unit cells colored red.

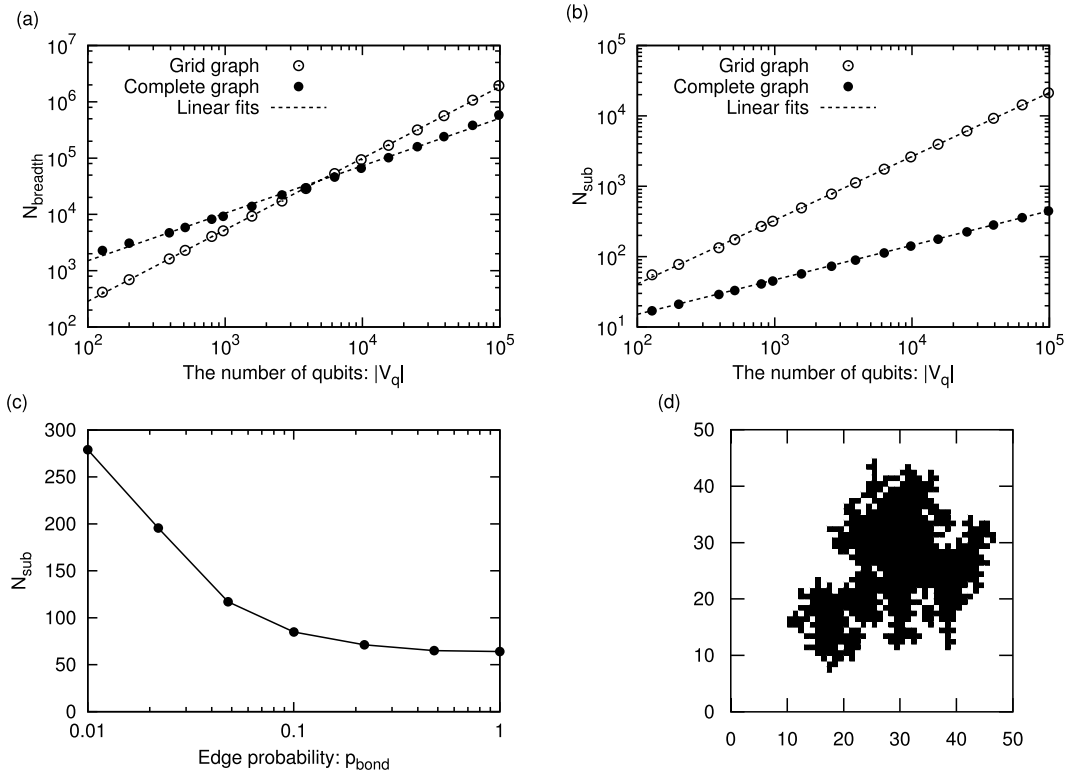

Figure 7. Experimental results from the proposed algorithm. (**a**) The $|V_q|$ dependence of $N_{breadth}$. (**b**) The $|V_q|$ dependence of N_{sub}. (**c**) The size of subproblems extracted from an Erdös-Rényi model with 1,000 logical variables for various edge probabilities p_{bond}. The subproblems are embedded into D-Wave 2000Q. (**d**) An example of a subproblem extracted from a grid graph with 50×50 logical variables. The subproblem is embedded into D-Wave 2000Q.

Experimental Results

In order to confirm the scalability of our algorithm, we evaluated the $|V_q|$ dependence of the number $N_{breadth}$ of the shortest paths searched by the breadth-first search and the size N_{sub} of the embedded subproblems. We have used the proposed algorithm to embed subproblems of a grid graph with 300×300 variables and a complete graph with 1,000 variables into a Chimera graph with $10^2 \sim 10^5$ qubits. The results for $N_{breadth}$ are shown in Fig. 7(a). Linear fits to the experimental results yield

$$N_{\text{breadth}}^{(\text{grid})} \sim O\left(|V_q|^{1.27}\right), \tag{17}$$

$$N_{\text{breadth}}^{(\text{complete})} \sim O\left(|V_q|^{0.84}\right). \tag{18}$$

The $|V_q|$ dependence of N_{breadth} is less than $O(|V_q|^{1.3})$, even for a grid graph with sparse connectivity. As the exponent is not large, we expect the proposed algorithm to be feasible even if the number of qubits is increased in a future version of the D-Wave quantum annealer. The results for N_{sub} are shown in Fig. 7(b). Linear fits to the experimental results yield

$$N_{\text{sub}}^{(\text{grid})} \sim O\left(|V_q|^{0.91}\right), \tag{19}$$

$$N_{\text{sub}}^{(\text{complete})} \sim O\left(|V_q|^{0.50}\right). \tag{20}$$

The size of subproblems for the complete graph $N_{\text{sub}}^{(\text{complete})}$ is identical to the maximum size embeddable into a Chimera graph. Although the sizes $N_{\text{sub}}^{(\text{grid})}$ of the subproblems for the grid graph are smaller than the ideal dependence $O(|V_q|^{1.0})$, they are much larger than those for the complete graph. These results imply that subproblems larger than the complete graph embedding[17,18] can be embedded depending on the connectivity of the problem graphs, and with a computation time shorter than that required for Cai's algorithm[16]. Because the subproblem embedding is searched greedily and refinement of the embedding is not implemented, an optimal embedding is hardly found especially for sparse problem graphs. However, the computational time is drastically reduced.

The sizes of subproblems extracted from an Erdös-Rényi model with 1,000 logical variables for various edge probabilities p_{bond} are shown in Fig. 7(c). Interactions between variables are randomly generated in this model, with edge probability p_{bond}, and the average sizes of subproblems for 100 instances are plotted in the graph. Subproblems are embedded into a Chimera graph in D-Wave 2000Q. As p_{bond} decreases, the size of the embedded subproblem increases. The proposed algorithm can embed larger subproblems depending on the connectivity of the problem graphs even if the interactions between variables are randomly generated.

An example of a subproblem extracted from a grid graph with 50×50 logical variables is shown in Fig. 7(d). The subproblem is embedded into a Chimera graph of the D-Wave 2000Q. The logical variables embedded as the subproblem are colored black. It is desirable that an extracted subproblem consists not of tree structures that are easily optimized but instead of many closed loops that can contain frustrations. The subproblem shown in Fig. 7(d) satisfies this condition.

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. *Science* **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. *Science* **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. 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).
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. Johnson, M. W. *et al.* Quantum annealing with manufactured spins. *Nature* **473**, 194–198, https://doi.org/10.1038/nature10012 (2011).
9. Rønnow, T. F. *et al.* Defining and detecting quantum speedup. *science* **345**, 420–424, https://doi.org/10.1126/science.1252319 (2014).
10. 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).
11. Neukart, F. *et al.* Traffic flow optimization using a quantum annealer. *Frontiers in ICT* **4**, 29, https://doi.org/10.3389/fict.2017.00029 (2017).
12. 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).
13. Morita, S. & Nishimori, H. Mathematical foundation of quantum annealing. *Journal of Mathematical Physics* **49**, 125210, https://doi.org/10.1063/1.2995837 (2008).
14. Bunyk, P. I. *et al.* Architectural considerations in the design of a superconducting quantum annealing processor. *IEEE Transactions in Applied Superconductivity* **24**, 1700110, https://doi.org/10.1109/TASC.2014.2318294 (2014).
15. Robertson, N. & Seymour, P. D. Graph minors.xiii. the disjoint paths problem. *Journal of Combinatorial Theory, Series B* **63**, 65–110, https://doi.org/10.1006/jctb.1995.1006 (1995).
16. Cai, J., Macready, B. & Roy, A. A practical heuristic for finding graph minors. Preprint at https://arxiv.org/abs/1406.2741 (2014).
17. 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).
18. 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).
19. 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).
20. Goodrich, T. D., Sullivan, B. D. & Humble, T. S. Optimizing adiabatic quantum program compilation using a graph-theoretic framework. *Quantum Inf Process* **17**, 118, https://doi.org/10.1007/s11128-018-1863-4 (2018).

21. Hamilton, K. E. & Humble, T. S. Identifying the minor set cover of dense connected bipartite graphs via random matching edge sets. *Quantum Inf Process* **16**, 94, https://doi.org/10.1007/s11128-016-1513-7 (2017).
22. Zaribafiyan, A., Marchand, D. J. J. & Changiz Rezaei, S. S. Systematic and deterministic graph minor embedding for cartesian products of graphs. *Quantum Inf Process* **16**, 136, https://doi.org/10.1007/s11128-017-1569-z (2017).
23. Harris, R. *et al*. Phase transitions in a programmable quantum spin glass simulator. *Science* **361**, 162–165, https://doi.org/10.1126/science.aat2025 (2018).
24. King, A. D. *et al*. Observation of topological phenomena in a programmable lattice of 1,800 qubits. *Nature* **560**, 456–460, https://doi.org/10.1038/s41586-018-0410-x (2018).
25. Wang, J.-S. & Swendsen, R. H. Cluster monte carlo algorithms. *Physica A: Statistical Mechanics and its Applications* **167**, 565–579, https://doi.org/10.1016/0378-4371(90)90275-W (1990).
26. 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).
27. Narimani, A., Saeed, S. S., Changiz Rezaei & Zaribafiyan, A. Combinatorial optimization by decomposition on hybrid cpu–non-cpu solver architectures. Preprint at https://arxiv.org/abs/1708.03439 (2017).
28. Chancellor, N. Modernizing quantum annealing using local searches. *New Journal of Physics* **19**, 023024, https://doi.org/10.1088/1367-2630/aa59c4 (2017).
29. 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).
30. Amin, M. H. Searching for quantum speedup in quasistatic quantum annealers. *Phys. Rev. A* **92**, 052323, https://doi.org/10.1103/PhysRevA.92.052323 (2015).
31. 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).
32. Benedetti, M., Realpe-Gómez, J., Biswas, R. & Perdomo-Ortiz, A. Estimation of effective temperatures in quantum annealers for sampling applications: A case study with possible applications in deep learning. *Phys. Rev. A* **94**, 022308, https://doi.org/10.1103/PhysRevA.94.022308 (2016).
33. 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).
34. 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).
35. Reverse quantum annealing for local refinement of solutions. https://www.dwavesys.com/sites/default/files/14-1018A-A_Reverse_Quantum_Annealing_for_Local_Refinement_of_Solutions.pdf (2017).

Acknowledgements

The authors are deeply grateful to Shu Tanaka, Masamichi J. Miyama, Tadashi Kadowaki, Hirotaka Irie and Akira Miki for fruitful discussions. One of the authors M.O. is grateful to the financial support from JSPS KAKENHI 15H03699 and 16H04382, the JST-START, JST-CREST(No. JPMJCR1402), and the ImPACT program.

Author Contributions

S.O. conceived and developed the concept, and carried out 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. M.T. and 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.