scientific reports

OPEN

Travel time optimization on multi-AGV routing by reverse annealing

Renichiro Haba[1,2]✉, Masayuki Ohzeki[1,2,3,4] & Kazuyuki Tanaka[1]

Quantum annealing has been actively researched since D-Wave Systems produced the first commercial machine in 2011. Controlling a large fleet of automated guided vehicles is one of the real-world applications utilizing quantum annealing. In this study, we propose a formulation to control the traveling routes to minimize the travel time. We validate our formulation through simulation in a virtual plant and authenticate the effectiveness for faster distribution compared to a greedy algorithm that does not consider the overall detour distance. Furthermore, we utilize reverse annealing to maximize the advantage of the D-Wave's quantum annealer. Starting from relatively good solutions obtained by a fast greedy algorithm, reverse annealing searches for better solutions around them. Our reverse annealing method improves the performance compared to standard quantum annealing alone and performs up to 10 times faster than a commercial classical solver, Gurobi. This study extends a use of optimization with general problem solvers in the application of multi-AGV systems and reveals the potential of reverse annealing as an optimizer.

Automated guided vehicles (AGVs) have been widely employed for transporting materials in factories, warehouses, and container terminals to improve flexibility and efficiency of manufacturing and distribution[1,2]. They move along markers or wires laid on the floor, which form a network. The design of the network layout can be broadly divided into a general and tandem layout. A tandem layout comprises multiple non-overlapping loops and exactly one AGV is assigned to each loop[3]. By contrast, a general layout does not limit the network topology and all paths can be used by any AGV[4,5]. A general layout leads to higher flexibility and productivity than a tandem layout because AGVs can travel on alternative routes or shortcuts. However, in such a system, multiple AGVs may share the same segment or intersection simultaneously, which could induce collisions or deadlocks depending on their planned routes. To avoid such a situation, AGVs' traveling routes are planned to be collision- and deadlock-free. In addition, the routes should be the shortest possible to minimize transfer time. Many routing algorithms have been examined for various problem settings[6,7]. In centralized AGV systems, information related to all AGVs, such as their delivery locations and current positions, is available for optimized routing[8,9]. Although this centralized optimization is a strong approach, it entails considerable computational time depending on the number of AGVs. To mitigate this issue, we focus on formulating the routing problem as a general mathematical form and exploiting existing fast mathematical solvers or meta-heuristics. As one of the solvers, we are interested in D-Wave Advantage, which is specialized in solving quadratic unconstrained binary optimization (QUBO) problems using quantum annealing.

Quantum annealing is a heuristic algorithm to solve QUBO problems by driving binary variables through quantum fluctuations[10]. Several well-known combinatorial problems can be encoded into QUBO problems[11]. In the ideal procedure, quantum annealing outputs the optimal solution by decreasing the strength of the fluctuation of binary variables. The quantum adiabatic theorem ensures the theoretical assurance of quantum annealing[12–14]. As well as quantum fluctuations, thermal fluctuations can be also utilized to help the system escape from local minima, which approach is called simulated annealing[15]. Quantum annealing is advantageous in the systems with the rugged energy landscape as quantum tunneling assist to tunnel through energy barriers which is hard to climb over by thermal excitation[16]. According to simulation results, quantum annealing finds ground states faster than simulated annealing for some types of problems such as square-lattice Ising model[17].

Currently, quantum annealing is realized with an actual machine by D-Wave Systems[18]. The machine does not perform the ideal quantum annealing because of the hardware and environmental effects and the solutions

[1]Graduate School of Information Sciences, Tohoku University, Sendai 980-8579, Japan. [2]Sigma-i Co., Ltd., Tokyo 108-0075, Japan. [3]Department of Physics, Tokyo Institute of Technology, Tokyo 152-8551, Japan. [4]International Research Frontier Initiative, Tokyo Institute of Technology, Tokyo 108-0023, Japan. ✉email: renichiro.haba.r6@dc.tohoku.ac.jp

are not always optimal, although it rapidly outputs relatively accurate solutions. As the environmental effect cannot be avoided, the quantum annealer is sometimes regarded as a simulator for the quantum many-body dynamics[19,20]. Several practical applications of quantum annealer have been presented across various fields, such as finance[21–23], traffic[24,25], logistics[26,27], manufacturing[9,28], and marketing[29], as well as in decoding problems[30,31]. Its potential for solving the optimization problem with inequality constraints has been enhanced[32], especially in the case that is hard to formulate directly[33]. The comparative study of quantum annealer has also been performed for benchmark tests to solve the optimization problems[34]. The quantum effect on the case with multiple optimal solutions has also been discussed[35,36]. Further, applications of quantum annealing for machine learning for solving optimization problems have been reported[37–42].

The above-mentioned studies have presented a routing algorithm for multiple AGVs to plan their traveling routes for given pairs of source and destination by utilizing a quantum annealer[9]. The routing aims to improve the working rate of AGVs. This objective is translated into the optimization problem to maximize the total travel distance of AGVs with constraints for removing latent collisions. However, as challenges, AGVs move with unnecessary detours, which reduces delivery efficiency. In addition, depending on the system, AGVs' repeated back and forth movement or along a cycle sometimes lacks efficiency. Therefore, we aim to develop an algorithm to control AGVs more efficiently. In this study, we formulate a QUBO problem to output deconflicted routes closest to the destinations of AGVs by modifying the objective function discussed in the previous study[9]. For a given task for each AGV, we minimize the total detour distance from the shortest paths. To express the detouring distance, we introduce an estimated distance of AGV, which is the travel distance remaining from the end of the path to the destination. By reducing it for all AGVs, we compel them to move ahead to their destination at each iteration. Consequently, we expect this method to accelerate cargo-carrying capabilities by shortening the detour of AGVs.

To optimize the QUBO problem, we employ D-Wave's quantum annealer for solving the QUBO problem. The latest version, that is, the D-Wave Advantage System, has two features—forward and reverse methods for quantum annealing[43–46]. To reveal the effectiveness of applying D-Wave's machine to our optimization problem, we explore both forward and reverse annealing features, which have not been hitherto discussed in AGVs' routing applications. In short, forward annealing performs a global search, starting in a superposition of all possible states with equal weights. By contrast, reverse annealing searches locally around a given classical state. Reverse annealing refines a solution around the given classical state unlike forward annealing, which searches for all possible states equally. For some types of problems, the reverse annealing approach improves the performance by utilizing initial solutions obtained by forward annealing or classical algorithms[23,47]. In this study, we present a reverse annealing method combined with a fast greedy search for solving the QUBO problem. The problem is also solved via a linear programming solver called Gurobi[48]. To evaluate the performance, we measure the time-to-solution benchmark for each solver.

The remainder of this paper is organized as follows: In the next section, we first review the previous method for the control of AGVs with less congestion. Next, we formulate the control of AGVs with fewer detours as the QUBO problem by modifying the objective function. Then, we present the optimization method that utilizes reverse annealing. To obtain the initial solution, we construct the greedy algorithm for the preprocessing of the reverse annealing method. In the following section, we evaluate our method through a simulation in a virtual plant. To validate the effectiveness of D-Wave's machine, we compare the results of the obtained solutions by each solver. In the final section, we summarize our study and discuss how future studies can improve the performance of solutions to this problem.

Method

In this section, we introduce our method for controlling AGVs using a quantum annealer. First, we review the study by Ohzeki et al.[9]. We briefly explain the routing algorithm and definition of the optimization problem and specify its issues. Then, we redefine the optimization problem to minimize the total journey time of AGVs. The optimization problem is relaxed to a QUBO form, which is heuristically solved by a quantum annealer. Finally, we describe how to run reverse annealing for the QUBO problem.

Assume that loads assigned to AGVs and their start and target positions are given. In the routing algorithm, we consider finding a route for each AGV to carry its load to its destination node from the source node in the network. We call a pair of source and destination nodes a task. A static routing algorithm determines a route for each task in advance, whereas a dynamic algorithm decides the route based on real-time information. A dynamic algorithm is advantageous as the routes are flexibly adjusted to reduce congestion and travel time.

For using a quantum annealer for AGVs' routing, a dynamic algorithm with QUBO optimization has been proposed[9]. The algorithm focuses on improving the working rate of AGVs. For dynamically updated tasks, the algorithm provides a good route for each AGV during a fixed period T. First, for each AGV, we generate multiple dissimilar candidate routes from its current node v_0 to destination v_t and shorten them to enable reaching within T s. Each candidate route is expressed as a list of nodes such that it connects x to t on the network. Second, we divide each candidate route into lists of segments to implement AGVs' stopping on the route. For example, if one candidate route consists of a path (v_0, v_1, v_t), then we split it into lists (v_0, v_1, v_t), (v_0, v_1), and (v_0), which represent moving the AGV two segments ahead, one segment ahead, and stopping, respectively. Note that in this algorithm, AGVs cannot go ahead during T second once they stop. Third, for all AGVs, we find the routes that maximize the working rate from the candidate routes. Finally, we move AGVs along the selected routes in T second and iterate the above procedure.

The procedure of choosing the best route is regarded as an optimization problem to maximize AGVs' working rate. The optimization problem comprises an objective function to be maximized or minimized with constraints. To represent the working rate, the objective function is defined as the total distance that AGVs move in the network. By maximizing the objective function, the total distance of routes is maximized and the solution leads to the highest working rate in the AGVs' system.

However, we have two issues with the objective function of the algorithm. One is inducing a longer detour distance, while the desired AGV's route should be the shortest possible to ensure fast delivery. The other is the possible suppressing of the system because of AGV's continuous back and forth movement in certain areas. To overcome these challenges, we present a new definition of the optimization problem to minimize travel time.

The travel time is proportional to the route distance and can be directly used for the objective function. However, recalling that we use the divided routes and not all of them reach the destination nodes, the length of the shortest route equals zero and that causes all AGVs to stop. In this study, we introduce the *remaining distance* instead of the route distance. For a given i-th AGV and j-th route, the remaining distance $d^*_{i,j}$ is the length of the shortest path from the end node of the route to its destination. By choosing the shortest route in terms of the remaining distance, AGVs move to nodes closest to their destination nodes for each iteration in the algorithm. Therefore, the algorithm greedily minimizes AGVs' travel time to complete their tasks. Let q_{ij} be a binary variable that takes a value 1 if the i-th AGV moves on the j-route and 0 otherwise. Then, the objective function is defined as

$$f_{\mathrm{obj}}(\mathbf{q}) = \sum_{i=1}^{N}\sum_{j=1}^{M_i} d^*_{i,j}q_{i,j} \tag{1}$$

where N is the number of AGVs and M_i is the number of candidate routes for the i-th AGV.

To safely control AGVs, collisions on roads or at intersections must be avoided. Deconfliction can be implemented by slowing down or stopping the AGVs after the traveling routes are arranged; however, we perform it simultaneously during the optimization. We utilize a similar technique discussed in the previous study to define constraints for collision avoidance[9]. We allow at most one AGV to occupy each line segment and intersection. We define $F_{i,j,t,e}$ such that $F_{i,j,t,e} = 1$ if the segment e is occupied with the j-th route of the i-th AGV at time t and $F_{i,j,t,e} = 0$ otherwise. Note that if an AGV is on a segment uv, we consider any segment coming to v as occupied. For any edge e and time t, $\sum_{i=1}^{N}\sum_{j=1}^{M_i} F_{i,j,t,e}q_{i,j}$ has to be at most one.

Now we define a mixed integer linear programming (MILP) problem as

$$\begin{array}{cc} \min_{\mathbf{q}} & \sum_{i=1}^{N}\sum_{j=1}^{M_i} d^*_{i,j}q_{i,j} \\ \text{subject to} & \sum_{j=1}^{M_i} q_{i,j} = 1, \quad \forall i \in \{1,2,\ldots,N\} \\ & \sum_{i=1}^{N}\sum_{j=1}^{M_i} F_{i,j,t,e}q_{i,j} \le 1, \quad \forall e \in E, \quad \forall t \in \{1,2,\ldots,T\} \end{array} \tag{2}$$

where E represents a set of edges in the network. The first constraint ensures that each AGV picks a single route and the second one ensures deconflicted routes. In general MILP problems, finding the exact solution quickly is difficult, although classical methods such as branch-and-bound algorithm are utilized to traverse the solution space efficiently[49].

Next, we transform the MILP problem into a QUBO problem to utilize quantum annealing. By using the penalty method for the equality and inequality constraints in problem (2), the cost function to be minimized in the QUBO problem can be expressed as:

$$f_{\mathrm{QUBO}}(\mathbf{q}) = \sum_{i=1}^{N}\sum_{j=1}^{M_i} d^*_{i,j}q_{i,j} + a\sum_{i=1}^{N}\left(\sum_{j=1}^{M_i} q_{i,j} - 1\right)^2 + b\sum_{e\in E}\sum_{t=1}^{T}\left(\sum_{i=1}^{N}\sum_{j=1}^{M_i} F_{i,j,t,e}q_{i,j} - \frac{1}{2}\right)^2 \tag{3}$$

where $a > 0$ and $b > 0$ denote penalty parameters that are tuned to ensure that the constraints are satisfied with the minimum energy. The second and third terms represent the equality and inequality constraints in problem (2), respectively. Thus, we have the following quadratic form with a matrix Q by transforming the function (3).

$$f_{\mathrm{QUBO}}(\mathbf{q}) = \mathbf{q}^{\mathrm{T}}Q\mathbf{q} + \mathrm{Const.} \tag{4}$$

General QUBO and MILP problems are known to be hard to strictly optimize. As QUBO problems do not have any constraints, the size of their feasible solution space tends to be fairly larger than that of MILP problems for the same problem size. Thus, heuristic algorithms are widely applied for quickly obtaining an adequate solution. In this study, we employ D-Wave's quantum annealer to solve the QUBO problem heuristically.

QUBO problems are equivalently converted to minimizing the energy of the Ising model, for which the Hamiltonian is expressed as:

$$H_0(\sigma) = -\sum_i h_i\sigma_i - \sum_{i<j} J_{ij}\sigma_i\sigma_j \tag{5}$$

where $\sigma_i \in \{-1, 1\}$ denotes a spin variable.

In quantum annealing, the quantum fluctuation is utilized to effectively find solutions of Ising models. The D-Wave machine operates the quantum system with superconducting qubits in a transverse field and its Hamiltonian is expressed as:

$$\hat{H}(s) = -A(s)\sum_i \hat{\sigma}^x_i + B(s)\hat{H}_0 \tag{6}$$

where $\hat{\sigma}_i^x$ is the x-component of Pauli matrices and $\hat{H}_0$ is a Hamiltonian attained by replacing each spin variable σ_i with the z-component of Pauli matrices $\hat{\sigma}_i^z$. The Hamiltonian (6) is controlled by a predetermined annealing schedule with a time-dependent parameter $0 \leq s \leq 1$. The functions $A(s)$ and $B(s)$ control the magnitude of transverse field to satisfy $A(0) \gg B(0)$ and $A(1) \ll B(1)$. At $A(0) \gg B(0)$, the qubits have the trivial ground state with a uniform superposition of all possible states. At $A(1) \ll B(1)$, the system has a classical state equivalent to the spin variables.

Depending on the annealing schedule, quantum annealing is classified into forward and reverse annealing. Forward annealing starts with $s = 0$ and gradually increases to $s = 1$ in the predetermined annealing time. By gradually decreasing the magnitude of the transverse field, the qubits are dephased into a nontrivial classical state of the Ising system. In an ideal procedure of forward annealing, the optimal state of the Ising model is known to be attained. However, D-Wave's quantum annealer couples to an open system that is affected by its environment, which leads to a limited optimization performance.

Unlike forward annealing, which is initialized with the full superposition, reverse annealing starts with a classical state and searches for a better solution in its vicinity. In reverse annealing, the system starts with $s = 1$ and gradually decreases s to $s = 1 - r$, where $0 < r < 1$ is the reversal distance, which determines the strength of transverse field. Pausing the schedule at $s = 1 - r$ for a certain period exploits the thermal relaxation in a low-temperature bath and enhances the optimization performance. After the pausing, the system is annealed forward to $s = 1$ and the qubits are dephased into a classical state.

In this study, we employ reverse annealing mainly for two reasons. One is to improve the performance of optimizations by combining a fast classical heuristic algorithm. The other is to ensure the feasibility of solutions initially obtained by some classical algorithms. Both forward and reverse annealing have no assurance of the output precision and infeasible solutions could be obtained, which induce fatal errors in manufacturing systems. Thus, an alternative technique to find feasible solutions is necessary for practical use, which can be utilized if the annealing method fails to find feasible solutions. Furthermore, reverse annealing provides an opportunity to refine such feasible solutions and exceed the performance of forward annealing alone.

We introduce a greedy algorithm to derive feasible solutions and utilize them as initial states for reverse annealing. The simplified flow of the algorithm is as follows.

1. Allocate the shortest route to each vehicle.
2. For any one of the two vehicles that collide, assign the next route.
3. Iterate 2 until all collisions are removed.

In the algorithm, the selections of any two conflicting routes rely on the calculation of an impact value, which is the number of conflicting AGVs for their next candidate routes. One route with a smaller impact value is dismissed and the other route with a larger impact value is selected. By exploring the candidate routes in the ascending orders of the remaining distance, the algorithm greedily searches the solutions. As the candidate routes are separated to stop on the way, the algorithm always finds conflict-free routes. This algorithm runs much faster than solving the MILP or QUBO problems. Specifically, the number of iterations in the algorithm is $O(nm)$, where n and m are the number of AGVs and candidate routes, respectively.

The greedy algorithm does not consider any detour distance in the selection of routes and aims to quickly find the deconflicted routes, which results in exploiting the high working rate. This feature enables us to imitate conditions similar to those in the previous study. The working rate of AGVs is also preferred to be high in the AGV systems to avoid traffic congestion. In the next section, we compare the algorithms in terms of the working rate and carrying efficiency.

To exploit the performance of reverse annealing, we maintain the annealing schedule properly. The most dominant parameter is reversal distance r. If it is too small, the solution stays in the initial state and if it is too large, the probability of achieving a global minimum by performing the global search equivalently as forward annealing is low. To find the reversal distance with a high possibility to output optimal solutions, for 10 randomly chosen QUBO matrices during the algorithm for controlling 20 AGVs, we analyzed the solutions for different reversal distances. The result is illustrated in Fig. 1. Obtaining the same initial state had a high probability with a small reversal distance, which gradually decreases as reversal distance increases. When reversal distance is too large, reverse annealing fails to find optimal states in most cases. Overall, the count of obtaining optimal solutions peaks around $r = 0.45$. Thus, we find the reversal distance 0.45 as the probability of obtaining the ground states peaked around the value and the probability of obtaining the same states is small. Finally, we have a calibrated reverse annealing schedule shown in Fig. 2. The default annealing time of D-Wave's quantum annealer is $20\,\mu\text{s}$ and we use this setting in forward annealing. We set the annealing time in reverse annealing to $13.3\,\mu\text{s}$ with a pause of $10\,\mu\text{s}$, which is shorter than that in forward annealing.

Results

In this section, we report the results of the dynamic algorithm solving the QUBO problem at each time. To verify the effectiveness of the algorithm, we simulate the AGV system in a simple virtual plant, as illustrated in Fig. 3. In the plant, 20 AGVs are active and they are provided a list of tasks that have to be completed at the earliest. In fixed simulation time, the number of completed tasks corresponds to the efficiency of carrying, which is also evaluated by the total travel time to finish the fixed number of tasks. The speed of each AGV is set to 0.5 m/s.

We iterate the algorithm 500 times and observe the number of tasks completed and the working rate. The time interval T per iteration is set by 2 s. In the simulation, different solvers are utilized to obtain the solution to each QUBO problem, specifically, greedy algorithm, Gurobi, forward annealing, and reverse annealing. As mentioned above, the greedy algorithm attempts to avoid congestion by reducing the number of AGVs stopping.

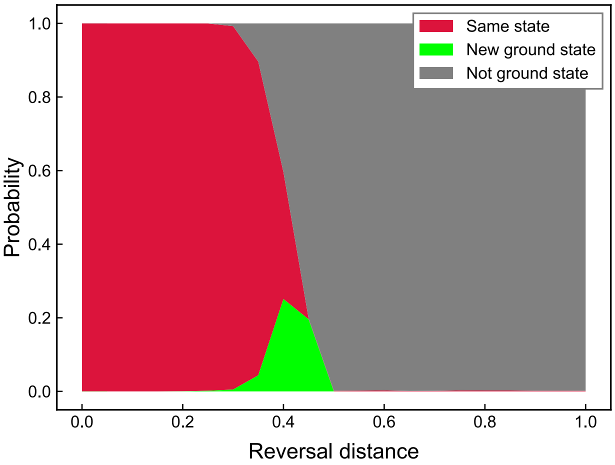

Figure 1. Calibration of reversal distance. We performed reverse annealing on 10 random nontrivial QUBO problems that appeared while running the algorithm with 20 AGVs and analyzed the energies of each of the 1000 samples. For a given reversal distance, the height of the red, green, and gray areas represents the mean probability of attaining the same state, the ground state, and the other state, respectively.

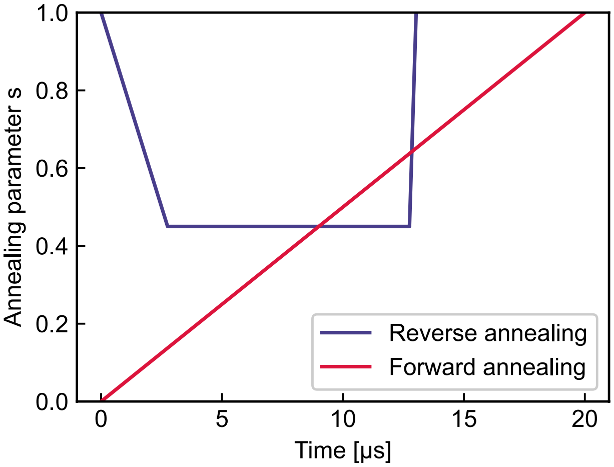

Figure 2. Annealing schedule of forward and reverse annealing. The red and blue lines indicate the chronological changes in annealing parameter s in forward and reverse annealing, respectively.

Thus, we regard it as similar to the previous algorithm and employ it to compare with our method. For forward and reverse annealing, we set the number of samples to 1, 000 and the state with the lowest energy is chosen in the routing algorithm. The greedy algorithm and Gurobi are operated on Intel(R) Core(TM) i7-8569U CPU.

The results of the simulation are presented in Table 1. At each optimization, Gurobi always outputs the optimal solution and leads to the highest performance in completed tasks. Although the working rate of the greedy algorithm was higher than that of Gurobi, the completed tasks of the greedy algorithm were less than that of Gurobi. This is because AGVs move with unnecessary detours in the greedy algorithm, which indicates that travel time is reduced by solving our optimization problem. In our case, forward annealing performed worst both in the completed tasks and working rate because of failing to find low-energy solutions. By contrast, reverse annealing outperformed forward annealing and resulted in almost the same scores as those of Gurobi. To conclude, the optimization of the QUBO problem 3 is effective in realizing a time-efficient multi-AGV system as well as solving the MILP problem. In addition, Gurobi and reverse annealing are comparable as solvers to control 20 AGVs.

To compare the performance of solvers, we benchmark time-to-solutions (TTS), defined as:

$$\mathrm{TTS}(p) = t_c \frac{\log(1-p)}{\log(1-p_s)}, \tag{7}$$

where p is the probability of obtaining the optimal solution at least once with a fixed number of trials, p_s is the probability of obtaining the optimal solution with a single trial, and t_c is the computational time for a single trial. For example, TTS(0.99) shows the estimated amount of time to obtain the optimal solution with a 99%

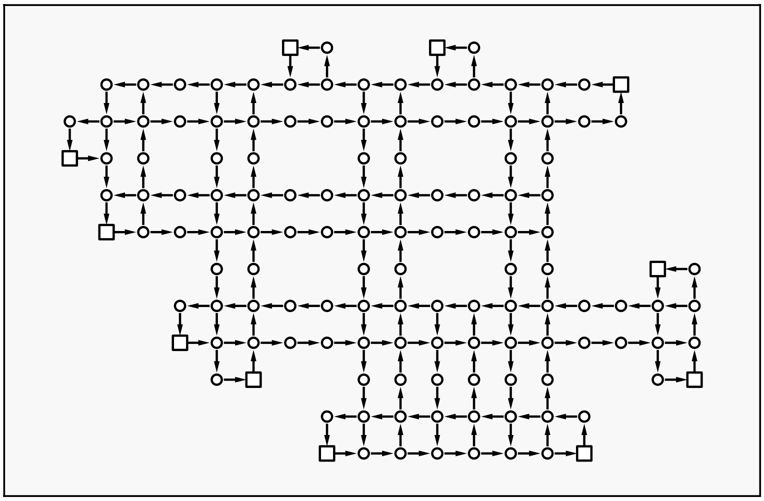

Figure 3. Virtual plant we used in this study. The network shows the roads on which AGVs can move. Each circle represents a node, which is an intersection or a mid-point. Each arrow represents the directed connection between two nodes. The length of each arrow is one meter. The square nodes indicate the pick-up or drop-off points that can be the source and destination of AGVs. In the plant, 20 AGVs are active on their task.

	Greedy	Gurobi	Forward annealing	Reverse annealing
Completed tasks	603.0	629.0	542.1 ± 2.3	627.4 ± 1.9
Working rate (%)	98.9	97.8	84.9 ± 0.2	97.8 ± 0.1

Table 1. Comparison of algorithms and solvers. The mean number of completed tasks for 500 iterations and the working rate of AGVs obtained by the greedy algorithm, Gurobi, forward annealing, and reverse annealing. Data are drawn from 10 identical experiments. The standard deviation is shown on the right hand side of the plus-minus sign.

chance. For the computation time of reverse annealing, we use annealing time during access time on the quantum processing unit. The annealing time is the total amount of time to complete a given annealing schedule. For reverse annealing, we apply the annealing schedule as shown in Fig. 2 and the annealing time of a single anneal is 13.3 μs. To explore the problem size scalability of the solvers, we measure TTS for different numbers of AGVs. The problem size is the summation of the number of candidate routes for each AGV. Note that the problem size differs in the algorithm because of separating routes to let AGVs wait. We calculate the TTS(0.99) for 10 QUBO problems appearing in the simulation. The number of samples in reverse annealing is set to 10, 000. The result is depicted in Fig. 4. We plot the wall-clock time for Gurobi to obtain the optimal solution instead of TTS. Reverse annealing outperforms Gurobi with almost 10 times shorter time when the problem size is small. As the problem size increased, reverse annealing needed a longer time to obtain the optimal solution, and Gurobi outperformed reverse annealing. Reverse annealing failed to find the optimal solution when the problem size was over 100 and benchmarking TTS was impossible. We assume this is because initial solutions obtained by the greedy algorithm become worse for larger problems.

To investigate the relationship between initial states and outputs in reverse annealing, we evaluate their closeness to optimal in terms of energy, which is measured by residual energy defined as:

$$E_{\mathrm{res}} = \frac{\langle E \rangle - E_{\mathrm{min}}}{E_{\mathrm{min}}} \tag{8}$$

where $\langle E \rangle$ is the mean energy of samples and E_{min} is the energy of the optimal solution. The residual energy corresponds to the closeness between the mean and minimum energy. We obtain 10,000 samples by forward and reverse annealing. The result is depicted in Fig. 5. The greedy algorithm always leads to a unique solution, and we plot the fraction of its energy over the minimum one. The greedy algorithm has extremely high residual energy as problem size increases. When the problem size is less than 50, the residual energy of forward annealing is higher than that of the greedy algorithm, which indicates that the greedy algorithm performs better than forward annealing. By contrast, reverse annealing resulted in lower residual energy even for small problems. In reverse annealing, the residual energy tends to be higher but still less than forward annealing as the problem size increases up to 215. For 250 variables, reverse annealing has higher residual energy than forward annealing and may turn worse for problem sizes larger than 250.

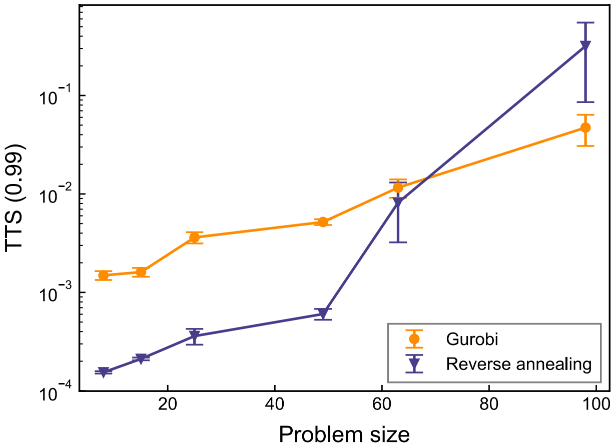

Figure 4. Comparison of TTS performance. The orange circle and purple triangle show mean value of TTS(0.99) of Gurobi and reverse annealing, respectively. The error bars indicate the standard error.

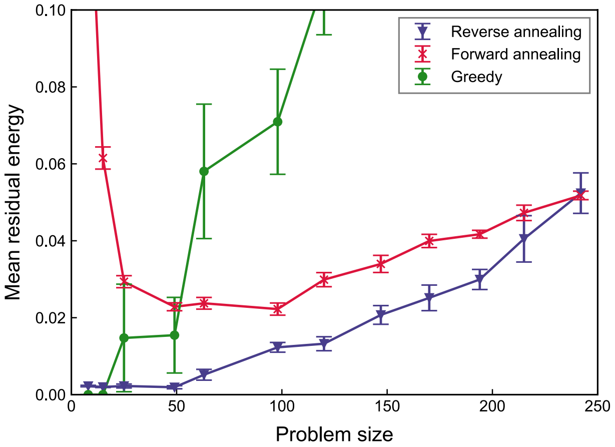

Figure 5. Comparison of residual energy. The purple triangle, red cross, and green circle show mean residual energy of samples on 10 problems solved by reverse annealing, forward annealing, and the greedy algorithm, respectively. The error bars indicate the standard error.

Discussion

In this study, we formulated a QUBO problem to control multiple AGVs' routes with minimized detours for a given period. Using a simulation, we demonstrated that our algorithm yields time-efficient routing. However, controlling AGVs as simulated is not always possible because of unpredictable events such as human errors and communication delays. To test our methods under such practical conditions and realize robust routing algorithms is interesting.

As an optimization method, we utilized reverse annealing with initial states obtained by a fast greedy algorithm. We confirmed that our reverse annealing method has the potential to exceed forward annealing alone or even a classical commercial solver, Gurobi. Practically, reverse annealing performed up to 10 times faster than Gurobi to obtain optimal solutions for small problem sizes. We believe this superiority of reverse annealing over Gurobi is caused by the specialty in short-time sampling on D-Wave's quantum annealer. In contrast, forward annealing suffered in finding optimal solutions even with small problems. We reckon this unexpected behavior of D-Wave Advantage 1.1 is due to its unique characteristic. According to the technical report by D-Wave Systems, such behavior is not announced and the system seems to perform as expected[50]. Thus, D-Wave Advantage 1.1 may have a weakness for the types of problems that we seek to solve.

By contrast, reverse annealing incorporates the benefits of both the greedy algorithm and quantum annealing and seems to have stable performance. One possible approach to improve our reverse annealing methods is switching the initial algorithm from the greedy algorithm to forward annealing depending on their performance. The residual energy of the greedy algorithm is considerably higher than the minimum one, although reverse annealing corrects the initial state to a certain extent. Choosing the right initial solver for large problems will improve the performance of reverse annealing. One choice of the approximate solvers can be a mean-field approximation and its variants[51]. The performance of reverse annealing with the mean-field approximation has been investigated[31]. Employing approximation solvers with a theoretical guarantee for setting initial conditions on reverse annealing can enable us to investigate the performance and design a theoretical method for assessing its limitation. As demonstrated in our study, although it is just one of the optimization problems, the performance of reverse annealing with the greedy algorithm is comparable with that of the commercial best optimizer. By assessing various combinations of reverse annealing and approximate solvers, we may find a new classical-quantum hybrid scheme to solve large-scale optimization problems.

In this study, simulated annealing is beyond our scope because several studies pointed out the advantageous features of theoretical quantum annealing over simulated annealing. However, the current D-Wave's physical machine does not yet follow the ideal adiabatic condition, and simulated annealing possibly outperforms standard forward annealing. Hence, it is an open question that the current D-Wave's quantum annealer still performs better than simulated annealing, if both annealing time are set identically.

Data availability

The datasets used during the current study are available from the corresponding author on reasonable request.

Received: 2 May 2022; Accepted: 18 October 2022
Published online: 22 October 2022

References

1. Ullrich, G. *Automated Guided Vehicle Systems* (Springer, 2015).
2. De Ryck, M., Versteyhe, M. & Debrouwere, F. Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *J. Manuf. Syst.* **54**, 152–173. https://doi.org/10.1016/j.jmsy.2019.12.002 (2020).
3. Bozer, Y. A. & Srinivasan, M. M. Tandem configurations for automated guided vehicle systems and the analysis of single vehicle loops. *IIE Trans.* **23**, 72–82. https://doi.org/10.1080/07408179108963842 (1991).
4. Gaskins, R. J. & Tanchoco, J. M. A. Flow path design for automated guided vehicle systems. *Int. J. Product. Res.* **25**, 667–676. https://doi.org/10.1080/00207548708919869 (1987).
5. Kaspi, M. & Tanchoco, J. M. A. Optimal flow path design of unidirectional AGV systems. *Int. J. Product. Res.* **28**, 1023–1030. https://doi.org/10.1080/00207549008942772 (1990).
6. Qiu, L., Hsu, W.-J., Huang, S.-Y. & Wang, H. Scheduling and routing algorithms for AGVs: A survey. *Int. J. Product. Res.* **40**, 745–760. https://doi.org/10.1080/00207540110091712 (2002).
7. Fazlollahtabar, H., Saidi-Mehrabad, M. & Balakrishnan, J. Mathematical optimization for earliness/tardiness minimization in a multiple automated guided vehicle manufacturing system via integrated heuristic algorithms. *Robot. Auton. Syst.* **72**, 131–138. https://doi.org/10.1016/j.robot.2015.05.002 (2015).
8. Jose, K. & Pratihar, D. K. Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods. *Robot. Auton. Syst.* **80**, 34–42. https://doi.org/10.1016/j.robot.2016.02.003 (2016).
9. Ohzeki, M., Miki, A., Miyama, M. J. & Terabe, M. Control of automated guided vehicles without collision by quantum annealer and digital devices. *Front. Comput. Sci.* **1**, 9. https://doi.org/10.3389/fcomp.2019.00009 (2019).
10. 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).
11. Lucas, A. Ising formulations of many NP problems. *Front. Phys.* **2**, 5. https://doi.org/10.3389/fphy.2014.00005 (2014).
12. Suzuki, S. & Okada, M. Residual energies after slow quantum annealing. *J. Phys. Soc. Jpn.* **74**, 1649–1652. https://doi.org/10.1143/JPSJ.74.1649 (2005).
13. Morita, S. & Nishimori, H. Mathematical foundation of quantum annealing. *J. Math. Phys.* **49**, 125210. https://doi.org/10.1063/1.2995837 (2008).
14. Ohzeki, M. & Nishimori, H. Quantum annealing: An introduction and new developments. *J. Comput. Theor. Nanosci.* **8**, 963–971. https://doi.org/10.1166/jctn.2011.1776963 (2011).
15. Kirkpatrick, 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).
16. Das, A. & Chakrabarti, B. K. Colloquium: Quantum annealing and analog quantum computation. *Rev. Mod. Phys.* **80**, 1061–1081. https://doi.org/10.1103/RevModPhys.80.1061 (2008).
17. Heim, B., Rønnow, T. F., Isakov, S. V. & Troyer, M. Quantum versus classical annealing of Ising spin glasses. *Science* **348**, 215–217. https://doi.org/10.1126/science.aaa4170 (2015).
18. Johnson, M. W. *et al.* Quantum annealing with manufactured spins. *Nature* **473**, 194–198. https://doi.org/10.1038/nature10012 (2011).
19. Bando, Y. *et al.* Probing the universality of topological defect formation in a quantum annealer: Kibble-Zurek mechanism and beyond. *Phys. Rev. Res.* **2**, 033369. https://doi.org/10.1103/PhysRevResearch.2.033369 (2020).
20. Bando, Y. & Nishimori, H. Simulated quantum annealing as a simulator of nonequilibrium quantum dynamics. *Phys. Rev. A* **104**, 022607. https://doi.org/10.1103/PhysRevA.104.022607 (2021).
21. 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).
22. Orús, R., Mugel, S. & Lizaso, E. Forecasting financial crashes with quantum computing. *Phys. Rev. A* **99**, 060301. https://doi.org/10.1103/PhysRevA.99.060301 (2019).
23. Venturelli, D. & Kondratyev, A. Reverse quantum annealing approach to portfolio optimization problems. *Quantum Mach. Intell.* **1**, 17–30. https://doi.org/10.1007/s42484-019-00001-w (2019).
24. Neukart, F. *et al.* Traffic flow optimization using a quantum annealer. *Front. ICT* **4**, 29. https://doi.org/10.3389/fict.2017.00029 (2017).
25. Hussain, H., Javaid, M. B., Khan, F. S., Dalal, A. & Khalique, A. Optimal control of traffic signals using quantum annealing. *Quantum Inf. Process.* **19**, 312. https://doi.org/10.1007/s11128-020-02815-1 (2020).
26. Feld, S. *et al.* A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer. *Front. ICT* **6**, 13. https://doi.org/10.3389/fict.2019.00013 (2019).
27. Ding, Y., Chen, X., Lamata, L., Solano, E. & Sanz, M. Implementation of a hybrid classical-quantum annealing algorithm for logistic network design. *SN Comput. Sci.* **2**, 68. https://doi.org/10.1007/s42979-021-00466-2 (2021).
28. Venturelli, D., Marchand, D. J. J. & Rojo, G. Quantum Annealing Implementation of Job-Shop Scheduling. arXiv:1506.08479 [quant-ph] (2016).
29. Nishimura, N., Tanahashi, K., Suganuma, K., Miyama, M. J. & Ohzeki, M. Item listing optimization for E-commerce websites based on diversity. *Front. Comput. Sci.* **1**, 2. https://doi.org/10.3389/fcomp.2019.00002 (2019).

30. Ide, N., Asayama, T., Ueno, H. & Ohzeki, M. Maximum likelihood channel decoding with quantum annealing machine, in *2020 International Symposium on Information Theory and Its Applications (ISITA)*, 91–95 (2020).
31. Arai, S., Ohzeki, M. & Tanaka, K. Mean field analysis of reverse annealing for code-division multiple-access multiuser detection. *Phys. Rev. Res.* **3**, 033006. https://doi.org/10.1103/PhysRevResearch.3.033006 (2021).
32. Yonaga, K., Miyama, M. J. & Ohzeki, M. Solving Inequality-Constrained Binary Optimization Problems on Quantum Annealer. arXiv:2012.06119 [quant-ph] (2020).
33. Koshikawa, A. S., Ohzeki, M., Kadowaki, T. & Tanaka, K. Benchmark test of black-box optimization using d-wave quantum annealer. *J. Phys. Soc. Jpn.* **90**, 064001. https://doi.org/10.7566/JPSJ.90.064001 (2021).
34. Oshiyama, H. & Ohzeki, M. Benchmark of quantum-inspired heuristic solvers for quadratic unconstrained binary optimization. *Sci. Rep.* **12**, 2146. https://doi.org/10.1038/s41598-022-06070-5 (2022).
35. Yamamoto, M., Ohzeki, M. & Tanaka, K. Fair sampling by simulated annealing on quantum annealer. *J. Phys. Soc. Jpn.* **89**, 025002. https://doi.org/10.7566/JPSJ.89.025002 (2020).
36. Maruyama, N., Ohzeki, M. & Tanaka, K. Graph minor embedding of degenerate systems in quantum annealing. arXiv:2110.10930 [quant-ph] (2021).
37. 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).
38. Kumar, V., Bass, G., Tomlin, C. & Dulny, J. Quantum annealing for combinatorial clustering. *Quantum Inf. Process.* **17**, 39. https://doi.org/10.1007/s11128-017-1809-2 (2018).
39. Adachi, S. H. & Henderson, M. P. Application of Quantum Annealing to Training of Deep Neural Networks. arXiv:1510.06356 [quant-ph, stat] (2015).
40. 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).
41. Arai, S., Ohzeki, M. & Tanaka, K. Teacher-student learning for a binary perceptron with quantum fluctuations. *J. Phys. Soc. Jpn.* **90**, 074002. https://doi.org/10.7566/JPSJ.90.074002 (2021).
42. Sato, T., Ohzeki, M. & Tanaka, K. Assessment of image generation by quantum annealer. *Sci. Rep.* **11**, 13523. https://doi.org/10.1038/s41598-021-92295-9 (2021).
43. Chancellor, N. Modernizing quantum annealing using local searches. *N. J. Phys.* **19**, 023024. https://doi.org/10.1088/1367-2630/aa59c4 (2017).
44. Ohkuwa, M., Nishimori, H. & Lidar, D. A. Reverse annealing for the fully connected *p*-spin model. *Phys. Rev. A* **98**, 022314. https://doi.org/10.1103/PhysRevA.98.022314 (2018).
45. Reverse Quantum Annealing for Local Refinement of Solutions. Technical Report, D-Wave Systems Inc. (2017).
46. Kadowaki, T. & Ohzeki, M. Experimental and theoretical study of thermodynamic effects in a quantum annealer. *J. Phys. Soc. Jpn.* **88**, 061008. https://doi.org/10.7566/JPSJ.88.061008 (2019).
47. Golden, J. & O'Malley, D. Reverse annealing for nonnegative/binary matrix factorization. *PLoS ONE* **16**, e0244026. https://doi.org/10.1371/journal.pone.0244026 (2021).
48. Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual* (2020).
49. Land, A. H. & Doig, A. G. An automatic method of solving discrete programming problems. *Econometrica* **28**, 497–520. https://doi.org/10.2307/1910129 (1960).
50. McGeoch, C. & Farre, P. The D-Wave Advantage System: An Overview. Technical Report, D-Wave Systems Inc. (2020).
51. Ohzeki, M. Message-passing algorithm of quantum annealing with nonstoquastic Hamiltonian. *J. Phys. Soc. Jpn.* **88**, 061005. https://doi.org/10.7566/JPSJ.88.061005 (2019).

Acknowledgements

The authors would like to express sincere gratitude to Assistant Prof. Manaka Okuyama for fruitful discussions and kind support for this study. This work was financially supported by JSPS KAKENHI Grant Nos. 20H02168, 19H01095, and 18H03303, partly supported by JST-CREST (No. JPMJCR1402), the Next Generation High-Performance Computing Infrastructures and Applications R &D Program of MEXT, and by MEXT-Quantum Leap Flagship Program Grant Number JPMXS0120352009.

Author contributions

R.H. conceived of the presented idea and performed the experiments. M.O. and K.T. verified the analytical methods and supervised the findings of this work. All authors discussed the results and contributed to the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to R.H.

Reprints and permissions information is available at www.nature.com/reprints.

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