



Item Listing Optimization for E-Commerce Websites Based on Diversity

Naoki Nishimura^{1,2*}, Kotaro Tanahashi², Koji Suganuma^{1,2}, Masamichi J. Miyama^{3,4} and Masayuki Ohzeki^{3,4,5}

¹ Internet Business Development Division, Recruit Lifestyle Co., Ltd., Tokyo, Japan, ² ICT Solution Department, Recruit Communications Co., Ltd., Tokyo, Japan, ³ Graduate School of Information Sciences, Tohoku University, Sendai, Japan, ⁴ Jij Inc., Tokyo, Japan, ⁵ Institute of Innovative Research, Tokyo Institute of Technology, Yokohama, Japan

OPEN ACCESS

Edited by:

Victor M. Eguíluz,
Institute of Interdisciplinary Physics
and Complex Systems (IFISC), Spain

Reviewed by:

Jonas Maziero,
Universidade Federal de Santa Maria,
Brazil
Hsi-Sheng Goan,
National Taiwan University, Taiwan

*Correspondence:

Naoki Nishimura
nishimura.n.ab@gmail.com

Specialty section:

This article was submitted to
Quantum Computing,
a section of the journal
Frontiers in Computer Science

Received: 27 March 2019

Accepted: 01 July 2019

Published: 16 July 2019

Citation:

Nishimura N, Tanahashi K,
Suganuma K, Miyama MJ and
Ohzeki M (2019) Item Listing
Optimization for E-Commerce
Websites Based on Diversity.
Front. Comput. Sci. 1:2.
doi: 10.3389/fcomp.2019.00002

For e-commerce websites, deciding the manner in which items are listed on webpages is an important issue because it can dramatically affect item sales. One of the simplest strategies for listing items to improve the overall sales is to do so in a descending order of popularity representing sales or sales numbers aggregated over a recent period. However, in lists generated using this strategy, items with high similarity are often placed consecutively. In other words, the generated item list might be biased toward a specific preference. Therefore, this study employs penalties for items with high similarity being placed next to each other in the list and transforms the item listing problem to a quadratic assignment problem (QAP). The QAP is well-known as an NP-hard problem that cannot be solved in polynomial time. To solve the QAP, we employ quantum annealing, which exploits the quantum tunneling effect to efficiently solve an optimization problem. In addition, we propose a problem decomposition method based on the structure of the item listing problem because the quantum annealer we use (i.e., D-Wave 2000Q) has a limited number of quantum bits. Our experimental results indicate that we can create an item list that considers both popularity and diversity. In addition, we observe that using the problem decomposition method based on a problem structure can provide to a better solution with the quantum annealer in comparison with the existing problem decomposition method.

Keywords: item listing, e-commerce, quadratic assignment problem, quantum annealing, D-Wave, problem decomposition

1. INTRODUCTION

Several companies have recently started operating e-commerce websites to sell their items and services to the public considering the widespread use of the internet. For these companies, deciding on the order in which items are listed on their website's pages is important because this ordering has the potential to dramatically affect the sales of their items or services. **Figure 1** shows a snapshot of a hotel reservation website. This is an example of the items being listed on an e-commerce page. On this website, hotels at different locations are listed in the order of popularity calculated based on various indicators from top to bottom. These sorted items for display on webpages are collectively referred to as an item list.

To improve sales on e-commerce websites, placing items in the descending order of popularity representing sales or sales numbers aggregated over a recent period is a simple strategy for determining the list order of items (Long and Chang, 2014). In addition, the popularity of an item can be estimated by it is placing it at different positions in the item list and determining the position of each product to maximize the total popularity estimate. In particular, if p_{ij} is the estimated popularity of an item $i \in I$ when it is placed in a position $j \in J$, then the total popularity of all items can be maximized by solving the following integer programming problem (Wang et al., 2016):

$$\begin{cases} \text{maximize} & \sum_{i \in I} \sum_{j \in J} p_{ij} x_{ij} \\ \text{subject to} & \sum_{i \in I} x_{ij} = 1, \quad j \in J, \\ & \sum_{j \in J} x_{ij} = 1, \quad i \in I, \\ & x_{ij} \in \{0, 1\}, \quad i \in I, \quad j \in J. \end{cases} \quad (1)$$

where x_{ij} is a binary variable that indicates whether or not to assign item i to position j . The abovementioned constraints ensure that only one item is allocated to each position, and only one position is allocated to each item. In this study,

$$P(\mathbf{x}) = \sum_{i \in I} \sum_{j \in J} p_{ij} x_{ij}$$

is referred to as the popularity term for $\mathbf{x} = (x_{11}, x_{12}, \dots)$. This problem can be interpreted as a network flow problem, and an efficient technique to solve such a problem in polynomial time exists. Furthermore, the solution obtained by solving this network flow problem with $x_{ij} \in [0, 1]$ coincides with the solution of the abovementioned integer programming problem (Vazirani, 2013).

However, in the case of the list of items generated using such a strategy, the relationship between the different objects is ignored because the popularity of each item p_{ij} is considered independently. For example, let us assume that customers visit an e-commerce website and browse the page of a particular item group. If the relationships among different items are not considered while placing items in an item list, several items with high similarities can possibly be placed close to each other, thereby reducing the value of the item list for customers in terms of item diversity. Considering this, several attempts have been made to include item diversity in item recommendation lists for users to ensure that they find the recommendation lists useful (Adomavicius and Kwon, 2011; Antikacioglu and Ravi, 2017). In these previous studies, the measures of diversity in the item recommendation lists for customers were improved by solving the maximum matching problem of the bipartite graph obtained after Top- N recommendation.

In this study, we introduce diversity into the item list for the entire user base, as well as methods for improving the usefulness of recommendation lists. An item list is generated by solving an optimization problem that imposes a penalty when items with high similarity are placed in adjacent to each other. Considering

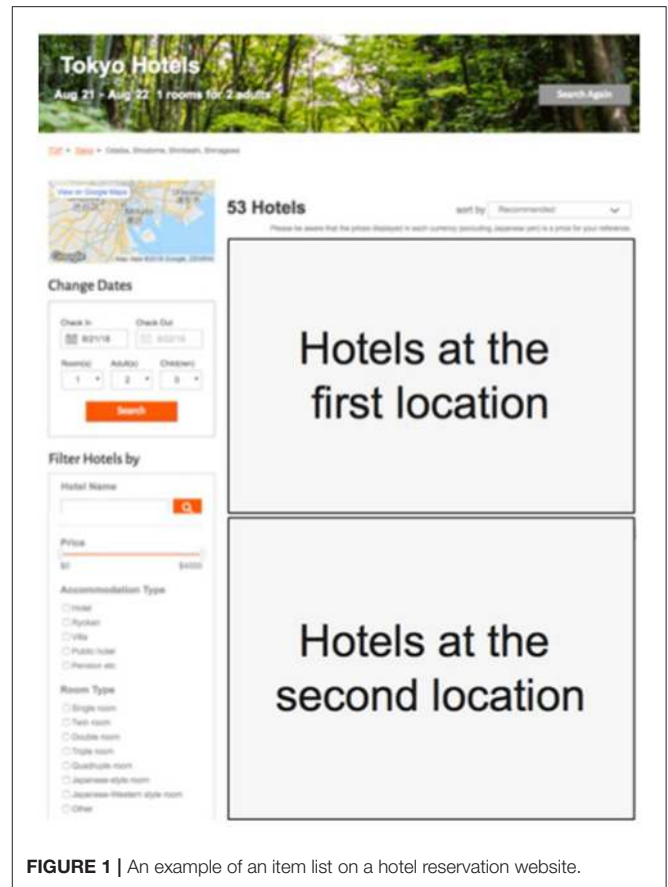


FIGURE 1 | An example of an item list on a hotel reservation website.

both popularity and diversity, the item list generation problem can be formulated as a quadratic assignment problem (QAP) as detailed below.

We employ quantum annealing (QA) herein to solve the QAP (Kadowaki and Nishimori, 1998). An optimization problem formulated with discrete variables can be efficiently solved using the Ising model or a quadratic unconstrained binary optimization problem (QUBO) because of the introduction of the quantum tunneling effect by QA. Currently, the protocol of QA is artificially realized in an actual quantum device known as a quantum annealer (Berkley et al., 2010; Harris et al., 2010; Johnson et al., 2010; Bunyk et al., 2014). The quantum annealer has been tested for numerous applications, including portfolio optimization (Rosenberg et al., 2016), protein folding simulation (Perdomo-Ortiz et al., 2012), online advertisement allocation optimization (Tanahashi et al., 2019), molecular similarity problem (Hernandez and Aramon, 2017), computational biology (Li et al., 2018), job-shop scheduling (Venturelli et al., 2015), traffic optimization (Neukart et al., 2017), election forecasting (Henderson et al., 2018), machine learning (Crawford et al., 2016; Khoshaman et al., 2018; Neukart et al., 2018), and for automated guided vehicles in plants (Ohzeki et al., 2019). In addition, several other studies have been conducted to efficiently solve various problems using the quantum annealer (Arai et al., 2018; Ohzeki et al., 2018a,b; Takahashi et al., 2018; Okada et al., 2019).

In particular, our problem can be solved using the quantum annealer by formulating our QAP as a QUBO. However, a QUBO for such a large number of items cannot be directly solved in one instance with the current state-of-the-art quantum annealer, namely D-Wave 2000Q because it employs the chimera graph. The physical qubits available on D-Wave 2000Q are less than 2048 because the qubits might have defects. In addition, the connection between the physical qubits is sparse and limited on the chimera graph. Thus, several embedding techniques have been proposed; however, the number of logical qubits available to represent the optimization problems to be solved is drastically reduced (Boothby et al., 2016). To address this issue, a heuristic method has been proposed to solve a large-sized problem using a limited number of hardware bits. D-Wave Systems, which is the manufacturer of D-Wave 2000Q, has developed an open-source software `qbsolv` (Booth et al., 2017) that solves a large-sized problem by dividing it into small subproblems. However, the decomposed QAP might not necessarily lead to feasible solutions because `qbsolv` selects the subset of variables in the order of the energy impact of each variable for division of a problem. Furthermore, in the literature (Okada et al., 2019), the division of an original problem into subproblems based on its structure is a promising method to efficiently solve large-sized problems using D-Wave 2000Q. Considering this, we propose herein a method to obtain better objective values for the QAP problem compared to those provided by the existing method (Booth et al., 2017) in the same calculation time by decomposing the problem based on the set of items and positions that they can be assigned to in an item list. In addition, we assess the performance of the proposed method using the actual access log of a hotel reservation website.

The primary contributions of our study are summarized as follows:

- We propose a method of creating item lists on an e-commerce website as a QAP considering the popularity and diversity of the items.
- We convert the QAP to the QUBO to solve the abovementioned problem with D-wave 2000Q.
- We propose a decomposition technique exploiting the structure of the item list.

2. MODELS

2.1. Formulating the Item Listing Optimization Problem as a QAP

We introduce the diversity term in our proposed model to add diversity in the item list. In particular, we calculate the similarity $f_{i'}$ for pairs of items $i, i' \in I$ to introduce diversity in the item list. The diversity of the item list (i.e., the diversity term) is defined as the negative value of the summation of the items' similarity degree $f_{i'}$ for overall adjacent items:

$$D(\mathbf{x}) = - \sum_{i \in I} \sum_{i' \in I} \sum_{j \in J} \sum_{j' \in J} f_{i'} d_{jj'} x_{ij} x_{i'j'}$$

where $d_{jj'}$ is the adjacent flag of the position j and j' ; $d_{jj'} = 1$ is for the adjacent positions; and $d_{jj'} = 0$ is for the

non-adjacent positions. The value of function $D(\mathbf{x})$ decreases because the high-similarity items are adjacent. We solve a multi-objective optimization problem based on two values: the popularity of individual products $P(\mathbf{x})$ and diversity of the item list $D(\mathbf{x})$. Let w be a parameter used to determine the penalty for listing items with high similarity. This problem is formulated as follows:

$$\begin{cases} \text{maximize} & \sum_{i \in I} \sum_{j \in J} p_{ij} x_{ij} - w \sum_{i \in I} \sum_{i' \in I} \sum_{j \in J} \sum_{j' \in J} f_{i'} d_{jj'} x_{ij} x_{i'j'} \\ \text{subject to} & \sum_{i \in I} x_{ij} = 1, \quad j \in J, \\ & \sum_{j \in J} x_{ij} = 1, \quad i \in I, \\ & x_{ij} \in \{0, 1\}, \quad i \in I, \quad j \in J. \end{cases} \quad (2)$$

Two methods for calculating the similarity $f_{i'}$ are introduced: explicit and implicit expressions. In the explicit expression, semantic features, such as product category and average price, can be quantified using the distances between the feature vectors that can be calculated. The smaller the feature distance, the higher the similarity $f_{i'}$ between the items. In the implicit expression, the higher the co-browsing number of an item (i.e., the number of times that the items were viewed in the same session by the same user on the website), the higher the similarity $f_{i'}$.

The advantage of the first approach is that the interpretation of the result is straightforward. Also, the semantic features of each item are available when the item is added to the database, that is, so-called cold start problems are avoided. Nevertheless, it suffers from a disadvantage in that appropriate semantic features must be created and quantified. In contrast, the advantage of the second approach is that it involves easy calculations and can consider various information reflecting customer behavior; however, its disadvantage is that semantic interpretation might be difficult. Section 3.1 describes the two methods for the calculation of the similarity $f_{i'}$ in detail, and section 3.2 compares the item lists created using these measures.

As previously specified, the optimization problem (2) is a QAP. The QAP is well-known as an NP-hard problem that cannot be solved in polynomial time (Anstreicher, 2003; Abdel-Basset et al., 2018).

2.2. Formulating the Item Listing Optimization Problem as a QUBO

We utilize QA to solve our optimization problem. The details of QA pertaining to D-Wave 2000Q are outlined in **Appendix A**. The optimization problem must be expressed in the form of a QUBO to use QA as a solver. QUBO is given as follows (Lucas, 2014):

$$\begin{cases} \text{minimize} & \mathbf{x}^T Q \mathbf{x} \\ \text{subject to} & \mathbf{x} \in \{0, 1\}^N, \end{cases} \quad (3)$$

where $Q \in \mathbb{R}^{N \times N}$. Thus, our optimization problem can be transformed into a QUBO by employing a penalty function for

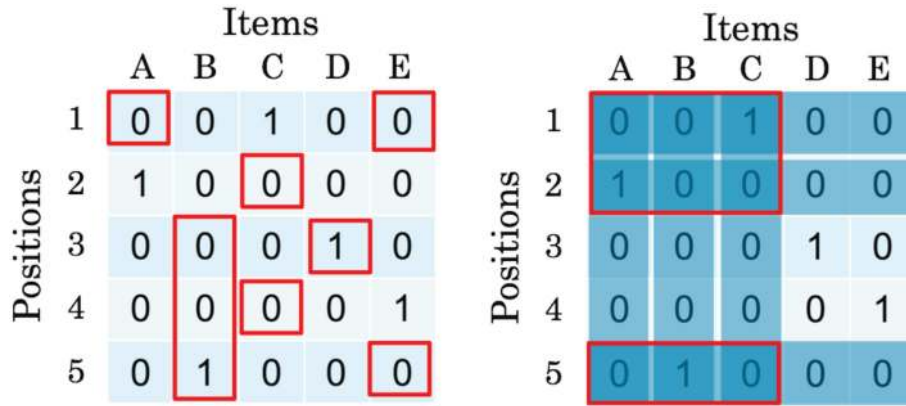


FIGURE 2 | Example of problem decomposition. The red frames represent the variables of the subproblem to be selected. The left figure is a selection of a subproblem without considering the problem structure. A subproblem has no feasible solution. The right figure is a selection of a subproblem based on the logical structure of the problem. A subproblem has feasible solutions.

violating constraints and adding this penalty function to the objective function:

$$\begin{aligned}
 & \text{minimize } - \sum_{i \in I} \sum_{j \in J} p_{ij} x_{ij} + w \sum_{i \in I} \sum_{i' \in I} \sum_{j \in J} \sum_{j' \in J} f_{i'i'} d_{jj'} x_{ij} x_{i'j'} \\
 & \quad + M \left(\sum_{i \in I} \left(\sum_{j \in J} x_{ij} - 1 \right)^2 + \sum_{j \in J} \left(\sum_{i \in I} x_{ij} - 1 \right)^2 \right) \\
 & \text{subject to } x_{ij} \in \{0, 1\}, \quad i \in I, \quad j \in J.
 \end{aligned} \tag{4}$$

where M is a parameter used to prevent the violation of the constraint conditions. This is ensured by setting an appropriate value for M . In theory, M should take an extremely large value. However, we cannot set M to such a large value because of the limitations of the current version of the quantum annealer used (i.e., D-Wave 2000Q). Thus, for simplicity, we present M to the size of the largest element of the absolute value of Q in (3).

2.3. Decomposition Methods for Item Listing Problems

`qbsolv` is a software tool released by D-Wave Systems that enables solving a QUBO larger than one that can be processed using D-Wave 2000Q (D-Wave Systems Inc., 2017). `qbsolv` is essentially a decomposing solver that divides a large problem into smaller parts, which can then be solved by D-Wave 2000Q. Thus, when a large QUBO is inputted, `qbsolv` divides the problem and sends each part of the problem independently to D-Wave 2000Q for calculation to obtain partial solutions. This process is repeated by selecting different parts of the problem using the tabu search until solution improvement stops. See Booth et al. (2017) for the detailed algorithm of `qbsolv`. Furthermore, `qbsolv` selects the subset of variables in the order of the energy impact of each variable for division of a problem. However, in some cases, no feasible solution can be obtained when the target variables are extracted, regardless of the structure of the original problem.

Therefore, we focus herein on the structure of the assignment problem and propose a method to extract problems with feasible solutions. Particularly in the case of an assignment problem, one condition involves each item being necessarily assigned to one position and another condition, in which each position is necessarily assigned to one item. Therefore, while dividing the problem, we have to select variables with candidate combinations of items and positions that are already assigned. **Figure 2** shows an example of the decomposition.

The original problem can be decomposed as follows if the number of items in the original problem is N_{org} and the number of items solved by a partial problem is N_{sub} :

1. Let \mathcal{N}_s be the set of N_{sub} items extracted from N_{org} items.
2. Let \mathcal{P}_s be the set of positions of items of \mathcal{N}_s .
3. Let $\mathcal{N}_s \times \mathcal{P}_s$ be the variables of the decomposed problem.

This procedure involves $N_{\text{org}}^2 C_{N_{\text{sub}}}^2$ candidates for variable combinations in the selection of the subproblems; however, the number of solution candidates can be reduced to $N_{\text{org}} C_{N_{\text{sub}}}$ exploiting the structure of the item list.

In practice, it is most important to determine the order in which items are listed in the upper positions of the item list because they are the items that are browsed most often. Therefore, it is effective to solve the entire list as an integer programming problem as in Problem (1) first, then only resolve the particularly important upper positions of the list using the QAP (2).

3. RESULTS AND DISCUSSION

3.1. Experimental Setup

For our experiments, we used the actual access log data of the online hotel reservation site Jalan¹. On this e-commerce website, a hotel list is created daily based on each area in Japan and the number of guests, including adults and children, that the hotels can accommodate in their rooms. The access

¹https://www.jalan.net/en/japan_hotels_ryokan/

log includes the date and the time the customer accessed the item list screen, position of each item when the item list screen was accessed, and information on the hotel at which the customer made a reservation. We estimated the popularity p_{ij} and similarity $f_{i'j'}$ for the top 10 accessed areas on the hotel reservation website using the access log for the past 6 months. The similarity $f_{i'j'}$ was estimated using two methods: the co-browsing similarity and the semantic similarity. The co-browsing similarity was estimated using the log of the co-browsed items in the same customer's session. On the other hand, semantic similarity was created and quantified by area and type of hotel. In our experiments, we used the co-browsing similarity, except for the comparison of the similarity measure in **Figure 7**. Various methods, such as machine learning algorithm (e.g., deep learning or gradient boosting) can be considered for estimating p_{ij} and $f_{i'j'}$. Furthermore, p_{ij} and $f_{i'j'}$ were normalized such that their average was 0 and the standard deviation was 1 for each item list.

We conducted two experiments in this study:

- evaluating the effect of the diversity term by comparison of solutions when the diversity control parameter is changed for the QAP (2), and
- evaluating the performance of problem decomposition by comparison of the objective values when the

structure of the item list is considered for the `qbsolv` problem decomposition.

As previously specified, we used D-Wave 2000Q (DW_2000Q_VFYC_2) for our experiments. Coupler strengths mapping logical to physical couplers with two physical couplers connecting each pair of logical qubits were set as 3.0. **Table 1** lists the values set for the parameters of D-Wave 2000Q and `qbsolv`. `num_read`, `annealing_time`, and `repeats` represent the number of requests for problems, time per annealing, and number of times the main loop of the algorithm is repeated with no change in the optimal value before stopping, respectively (Booth et al., 2017; D-Wave Systems Inc., 2017).

3.2. Effect of the Diversity Term

D-Wave 2000Q has less than 2048 qubits because the qubits typically have defects. In addition, as previously specified, the connection between the physical qubits is sparse and limited on the chimera graph, in which D-Wave 2000Q has been based on. Thus, we can consider the problem with eight items per subproblem because complete graph embedding can be applied to arbitrary problem graphs with less than 64 logical variables. Therefore, we first compare the popularity term $P(\mathbf{x})$ and the diversity term $D(\mathbf{x})$ obtained by solving the QAP (2) by changing the diversity control parameter w in the problem when the number of items is 6 and 8. The obtained solution will be the same as that obtained via solving (1) if w is 0.

In **Figures 3, 4**, the horizontal axes represent the value of w in (2); the right hand side in the figures indicates that the larger the similarity between the similar items listed together, the larger the penalty. In addition, the vertical axes in **Figures 3, 4** represent the values of $P(\mathbf{x})$ and $D(\mathbf{x})$, respectively. The average value of all solutions is approximately 0 because p_{ij} and $f_{i'j'}$ are normalized for each area. **Figures 3B, 4D** depict plots of $P(\mathbf{x})$ and $D(\mathbf{x})$, respectively, for three typical areas X, Y, and Z from among the 10 areas for the problem with eight items. **Figures 3A, 4C** are plotted by aggregating the typical values for the 10 areas, wherein

TABLE 1 | Parameters used for solving the problem in our experiments.

Parameter	Value
<code>num_reads</code> (D-Wave Systems Inc., 2017)	1,000
<code>annealing_time</code> (D-Wave Systems Inc., 2017)	20 [μ s]
<code>auto_scale</code> (D-Wave Systems Inc., 2017)	True
<code>postprocess</code> (D-Wave Systems Inc., 2017)	Optimization
<code>num_spin_reversal_transforms</code> (D-Wave Systems Inc., 2017)	4
<code>timeout</code> (Booth et al., 2017)	20 [s]
<code>repeats</code> (Booth et al., 2017)	5
<code>subproblemSize</code> (Booth et al., 2017)	64

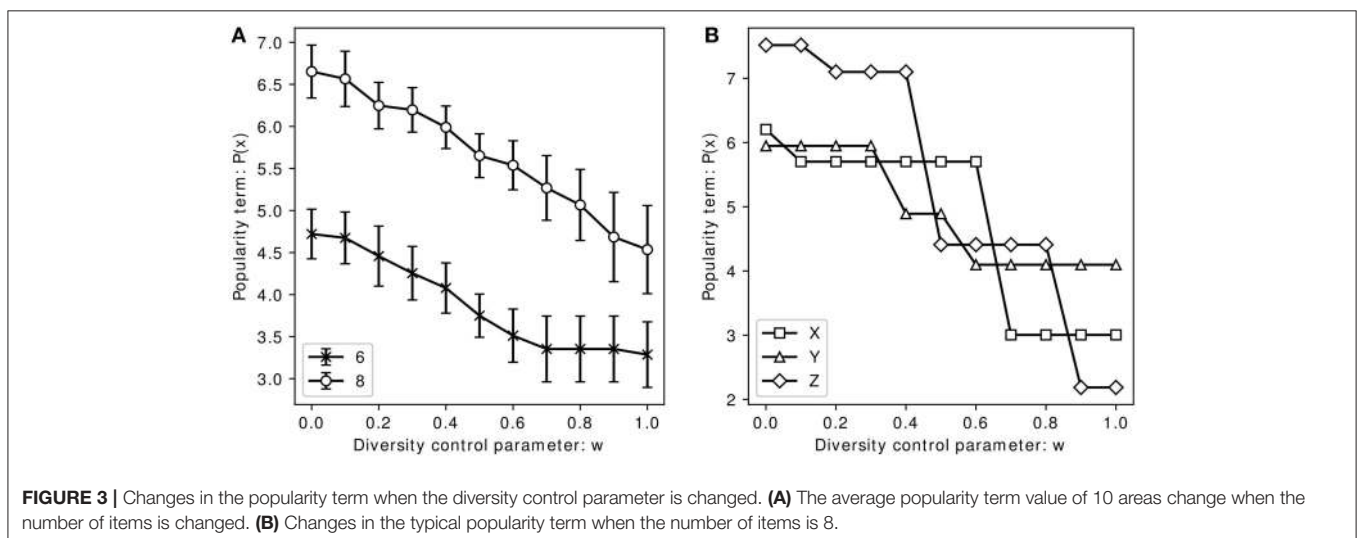


FIGURE 3 | Changes in the popularity term when the diversity control parameter is changed. **(A)** The average popularity term value of 10 areas change when the number of items is changed. **(B)** Changes in the typical popularity term when the number of items is 8.

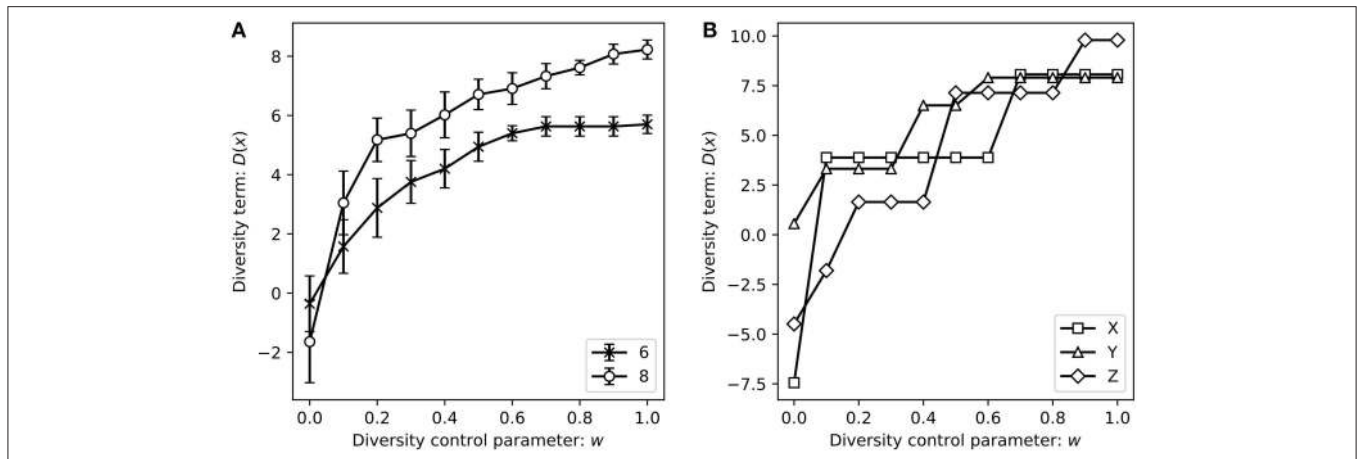


FIGURE 4 | Changes in the diversity term when the diversity control parameter is changed. **(A)** The average diversity term of 10 areas change when the number of items is changed. **(B)** Changes in the typical diversity term when the number of items is 8.

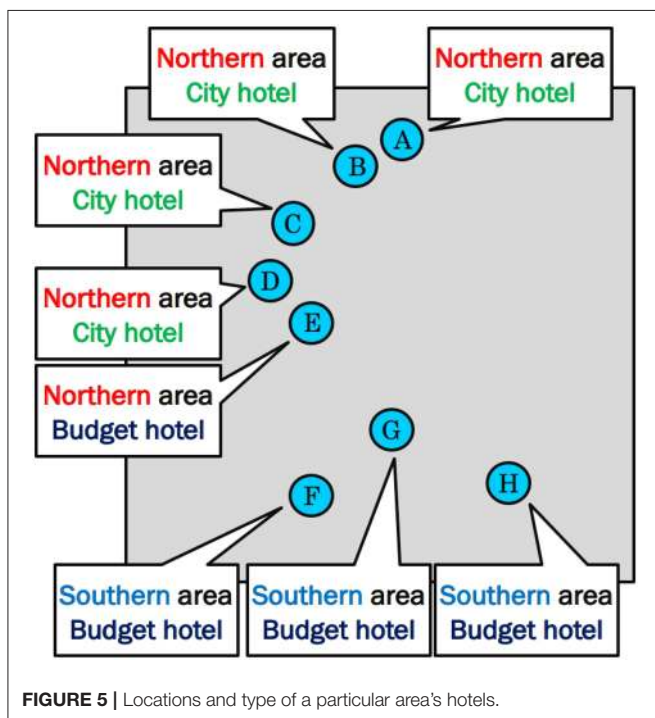


FIGURE 5 | Locations and type of a particular area's hotels.

the marker and the bar represent the average and standard error values, respectively.

Figure 3A shows that the average of the popularity term $P(x)$ is higher than 0 for any w . In addition, $P(x)$ gradually decreases as w increases. In other words, a trade-off relationship exists between increasing the popularity term $P(x)$ and ensuring that similar items are not placed adjacent to each other in the item list. Furthermore, the decrease in the popularity term value slows down as w increases. Figure 4C shows that when diversity is not considered for item listing, the diversity term $D(x)$ is lower than the average value of 0 and increases with increasing w . Furthermore, the increase in the diversity term $D(x)$ gradually slows down as w increases.

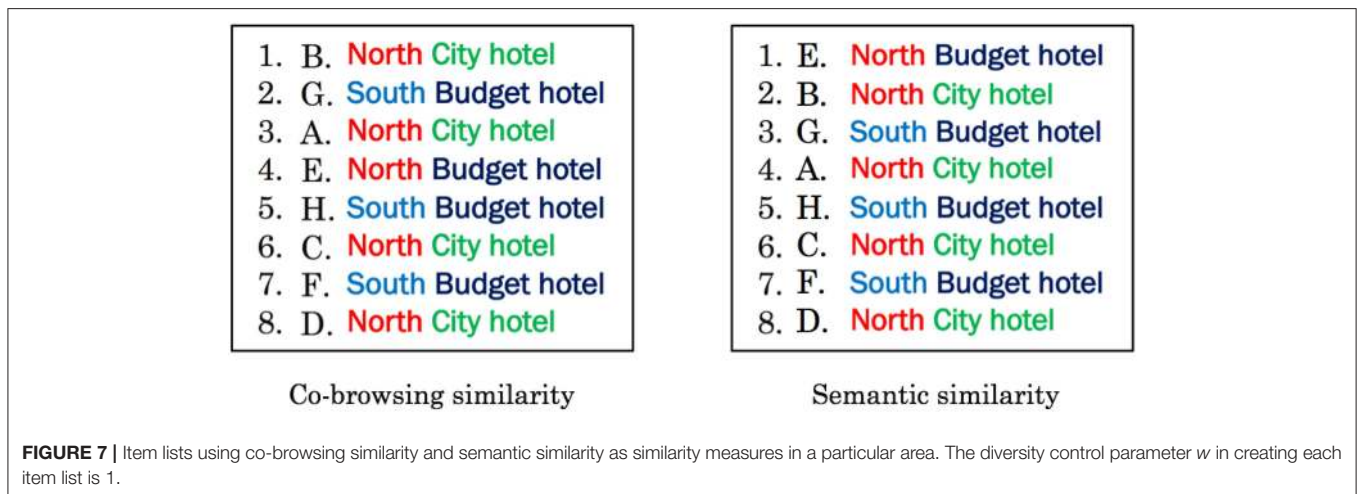
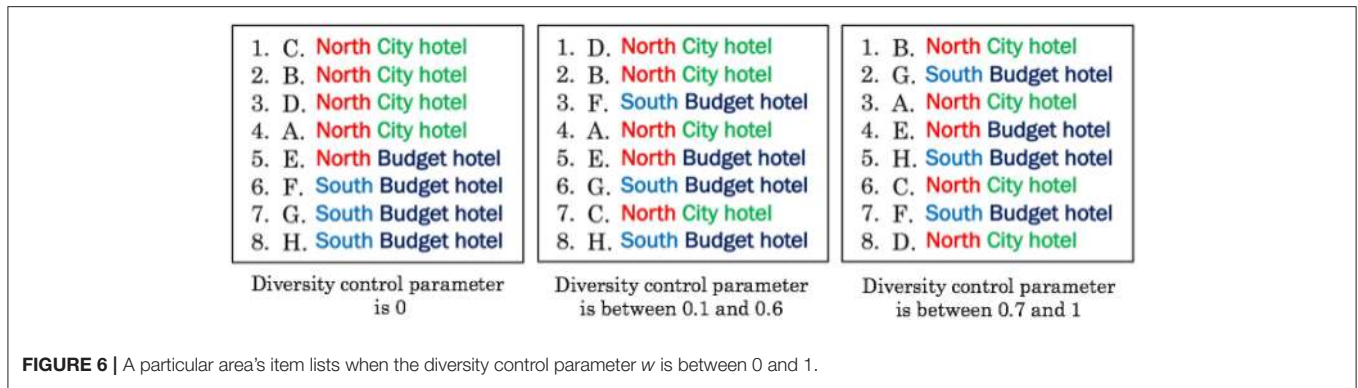
Note that the behavior of $P(x)$ and $D(x)$ based on w is the same whether or not the number of items is 6 or 8.

Figure 5 shows the positional relationships between the top eight hotels and their type in area X of Figures 3B, 4D. The top eight hotels in area X particularly include four city hotels and four budget hotels, which can also be classified as in the north or south region of area X; hence, it was chosen as an example.

Figure 6 represents the actual hotel lists in area X obtained by solving (2) for different values of w . The top five hotels in the northern area are consecutively listed when w is 0. The city hotels are listed in succession among the top four; thus, the list is biased. In contrast, only the top two hotels are the same in terms of both area and hotel type when w is between 0.1 and 0.6, that is, when diversity is considered. Furthermore, no similarity exists between the consecutively listed hotels both in terms of area and hotel type when w is larger than 0.7. Thus, despite not using semantic information (e.g., area and hotel type) in the actual calculations, a semantically diverse item list was created using the number of co-browsers as the similarity measure.

So far, we have used the co-browsing similarity as f_{ij} . In Figure 7, we compared the item lists in area X using the co-browsing similarity and the semantic similarity as similarity measures. Each item list was obtained by solving (2) for $w = 1$. The semantic similarity was calculated using the negative Euclidean distance of each hotel based on the two-dimensional vector of whether the area is in the north or south, and the hotel type (city hotel or budget hotel).

Figure 7 shows that the item list is not consecutive with respect to both the area and the hotel type by solving (2) even if both similarity types are used. The difference between these two item lists is that E, the budget hotel northern area, rose from the fourth to the first place. The item lists created using the explicit semantic similarity have fewer sequences for the hotel type than those created using the co-browsing similarity. This result confirms that the item list, which considered diversity for the purpose of this study, was consistently created regardless of the similarity measure used.



3.3. Performance Evaluation of Problem Decomposition

We compared the method of extracting partial problems by `qbsolv` and our method of extracting partial problems considering the problem structure of the item list. Our experiments were performed by comparing the objective values of (4) when solving the problem of 12, 16, 20 and 24 items using each method.

Table 2 lists the average of the objective values for the 10 areas obtained by solving the problem for different numbers of items and the gap between the objective values of the proposed and original `qbsolv` methods. The goal of the problem was to minimize the objective value, indicating that the smaller the objective value, the better the performance. The gap between the objective values of the proposed and original `qbsolv` methods increased as the number of items increased. The effectiveness of the proposed method also increased. We conducted a Wilcoxon signed-rank test on the difference between the objective values of the original `qbsolv` and proposed methods (Bonferroni correction was performed on the number of items). Consequently, the null hypothesis in this case was rejected at a significance level of 5% (p -value = 0.020).

In terms of application, our proposed method is not only limited to the item list optimization problem, but can also be widely applied to other problems involving similar constraints,

such as the assignment problem (1). For example, our method can be applied to the traveling salesman problem, which typically includes two constants A, B along with the following penalty terms:

$$A \sum_{i \in I} \left(\sum_{j \in J} x_{ij} - 1 \right)^2 + B \sum_{j \in J} \left(\sum_{i \in I} x_{ij} - 1 \right)^2 .$$

4. CONCLUSION

This study proposed a method of creating item lists on an e-commerce website as a QAP considering item popularity and diversity. We converted the QAP to a QUBO such that it can be directly solved by the quantum annealer, D-Wave 2000Q. Direct manipulation to solve the resulting QUBO was not possible in the case with a large number of items because of the limited number of qubits available in the current version of the quantum annealer and the restriction on specifying connections between the qubits. Therefore, we proposed a decomposition technique exploiting the structure of the problem. The original large problem was divided into several subproblems, which can eventually be solved by D-Wave 2000Q individually. Our experiments using actual real-world data demonstrated the efficiency of our proposed approach. A remarkable observation

TABLE 2 | Comparison of the objective values for problem decomposition.

Number of items	Methods		Gap
	qbsolv	Proposed method	
12	-160.038	-160.337	-0.299
16	-268.777	-270.176	-1.399
20	-391.428	-393.051	-1.623
24	-505.634	-509.266	-3.632

made from the experimental results was that the output item list changed based on the diversity control parameter. Our formulation led to the antiferromagnetic Ising model with a random field. The resulting lists were “aligned” along the random field when the diversity control parameter was small. In contrast, increasing the diversity control parameter eliminated the order in the item list and introduced diversity.

However, our research has some limitations. The item list created by our method will not work well when the data needed to calculate the popularity p_{ij} and similarity f_{ij} are insufficient, because no reliable estimate values will exist in that case. Obtaining good estimates for p_{ij} and f_{ij} requires access logs pertaining to when items are placed at various positions. The determination of the diversity consideration parameter w was also a limitation. w depends on the scale of diversity a customer of an e-commerce service wants; thus, it should be adjusted by changing w and monitoring performance, which is cumbersome to implement in real-world scenarios.

As the number of qubits available in the quantum annealer increases in the future, our method for division of a large-sized problem into smaller subproblems will become more useful. The experiments in this study clearly showed that our method performed better than `qbsolv` when the size of the problem

became large. The structure of our problem is similar to that of the traveling salesman problem and the scheduling problem in that it takes the form of a QUBO with two quadratic functions based on two constraints (i.e., one for distance or time and the other for list locations or tasks).

Thus, our method has a wide range of applications involving optimization problems that can be solved via QUBO formulation, such as those using the QA, D-Wave 2000Q, and other types of QUBO solvers. Our results indicate that not only the evolution of hardware devices, but also the development of better software based on the structure of problems are essential for future QA applications.

DATA AVAILABILITY

The datasets generated for this study can be found in the GitHub repository of the Recruit Communications Co., Ltd².

AUTHOR CONTRIBUTIONS

NN contributed to the conception and design of the study, performed all the experiments, and wrote the first draft of the manuscript. KT implemented the program of our problem decomposition method. MO, MM, and KS contributed to the manuscript revision. All authors discussed this study, then reviewed and approved the final version of the manuscript.

ACKNOWLEDGMENTS

The authors would like to thank Recruit Lifestyle Co., Ltd. and Recruit Communications Co., Ltd. for their support in this exploratory research project.

²<https://github.com/recruit-communications/Item-Listing-Datasets>

REFERENCES

- Abdel-Basset, M., Manogaran, G., Rashad, H., and Zaied, A. N. H. (2018). A comprehensive review of quadratic assignment problem: variants, hybrids and applications. *J. Amb. Intell. Human. Comput.* doi: 10.1007/s12652-018-0917-x. [Epub ahead of print].
- Adomavicius, G., and Kwon, Y. (2011). “Maximizing aggregate recommendation diversity: a graph-theoretic approach,” in *Proceedings of the 1st International Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011)* (Chicago, IL), 3–10.
- Anstreicher, K. M. (2003). Recent advances in the solution of quadratic assignment problems. *Math. Progr.* 97, 27–42. doi: 10.1007/s10107-003-0437-z
- Antikacioglu, A., and Ravi, R. (2017). “Post processing recommender systems for diversity,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Halifax, NS), 707–716. doi: 10.1145/3097983.3098173
- Arai, S., Ohzeki, M., and Tanaka, K. (2018). Deep neural network detects quantum phase transition. *J. Phys. Soc. Jpn.* 87:033001. doi: 10.7566/JPSJ.87.033001
- Berkley, A. J., Johnson, M. W., Bunyk, P., Harris, R., Johansson, J., Lanting, T., et al. (2010). A scalable readout system for a superconducting adiabatic quantum optimization system. *Superconduct. Sci. Technol.* 23:105014. doi: 10.1088/0953-2048/23/10/105014
- Booth, M., Reinhardt, S. P., and Roy, A. (2017). *Partitioning Optimization Problems for Hybrid Classical/Quantum Execution*. Available online at: <https://www.dwavesys.com/resources/publications>
- Boothby, T., King, A. D., and Roy, A. (2016). Fast clique minor generation in chimera qubit connectivity graphs. *Quant. Inform. Process.* 15, 495–508. doi: 10.1007/s11128-015-1150-6
- Bunyk, P., Hoskinson, E. M., Johnson, M. W., Tolkacheva, E., Altomare, F., Berkley, A. J., et al. (2014). Architectural considerations in the design of a superconducting quantum annealing processor. *IEEE Trans. Appl. Superconduct.* 24, 1–10. doi: 10.1109/TASC.2014.2318294
- Crawford, D., Levit, A., Ghadermarzy, N., Oberoi, J. S., and Ronagh, P. (2016). Reinforcement learning using quantum boltzmann machines. *arXiv: 1612.05695*.
- D-Wave Systems Inc. (2017). *D-Wave Solver Properties and Parameters Reference*. Available online at: <https://docs.dwavesys.com/docs/latest>
- D-Wave Systems Inc. (2018). *Getting Started With the D-Wave System*. Available online at: <https://docs.dwavesys.com/docs/latest>
- Harris, R., Johnson, M. W., Lanting, T., Berkley, A. J., Johansson, J., Bunyk, P., et al. (2010). Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor. *Phys. Rev. B* 82:024511. doi: 10.1103/PhysRevB.82.024511
- Henderson, M., Novak, J., and Cook, T. (2018). Leveraging adiabatic quantum computation for election forecasting. *arXiv: 1802.00069*.

- Hernandez, M., and Aramon, M. (2017). Enhancing quantum annealing performance for the molecular similarity problem. *Quant. Inform. Process.* 16:133. doi: 10.1007/s11128-017-1586-y
- Johnson, M., Bunyk, P., Maibaum, F., Tolkacheva, E., Berkley, A., Chapple, E., et al. (2010). A scalable control system for a superconducting adiabatic quantum optimization processor. *Superconduct. Sci. Technol.* 23:065004. doi: 10.1088/0953-2048/23/6/065004
- Kadowaki, T., and Nishimori, H. (1998). Quantum annealing in the transverse ising model. *Phys. Rev. E* 58:5355. doi: 10.1103/PhysRevE.58.5355
- Khoshaman, A., Vinci, W., Denis, B., Andriyash, E., and Amin, M. H. (2018). Quantum variational autoencoder. *Quant. Sci. Technol.* 4:014001. doi: 10.1088/2058-9565/aada1f
- Li, R. Y., Felice, R. D., Rohs, R., and Lidar, D. A. (2018). Quantum annealing versus classical machine learning applied to a simplified computational biology problem. *NPJ Quant. Inform.* 4:14. doi: 10.1038/s41534-018-0060-8
- Long, B., and Chang, Y. (2014). *Relevance Ranking for Vertical Search Engines*. Waltham, MA: Elsevier. Available online at: <https://www.elsevier.com/books/relevance-ranking-for-vertical-search-engines/long/978-0-12-4077171-1>
- Lucas, A. (2014). Ising formulations of many np problems. *Front. Phys.* 2:5. doi: 10.3389/fphy.2014.00005
- Neukart, F., Compostella, G., Seidel, C., von Dollen, D., Yarkoni, S., and Parney, B. (2017). Traffic flow optimization using a quantum annealer. *Front. ICT* 4:29. doi: 10.3389/fict.2017.00029
- Neukart, F., Von Dollen, D., Seidel, C., and Compostella, G. (2018). Quantum-enhanced reinforcement learning for finite-episode games with discrete state spaces. *Front. Phys.* 5:71. doi: 10.3389/fphy.2017.00071
- Ohzeki, M., Miki, A., Miyama, M. J., and Terabe, M. (2019). Control of automated guided vehicles without collision by quantum annealer and digital devices. *arXiv: 1812.01532*.
- Ohzeki, M., Okada, S., Terabe, M., and Taguchi, S. (2018a). Optimization of neural networks via finite-value quantum fluctuations. *Sci. Rep.* 8:9950. doi: 10.1038/s41598-018-28212-4
- Ohzeki, M., Takahashi, C., Okada, S., Terabe, M., Taguchi, S., and Tanaka, K. (2018b). Quantum annealing: next-generation computation and how to implement it when information is missing. *Nonlin. Theory Its Appl.* 9, 392–405. doi: 10.1587/nolta.9.392
- Okada, S., Ohzeki, M., Terabe, M., and Taguchi, S. (2019). Improving solutions by embedding larger subproblems in a d-wave quantum annealer. *Sci. Rep.* 9:2098. doi: 10.1038/s41598-018-38388-4
- Perdomo-Ortiz, A., Dickson, N., Drew-Brook, M., Rose, G., and Aspuru-Guzik, A. (2012). Finding low-energy conformations of lattice protein models by quantum annealing. *Sci. Rep.* 2:571. doi: 10.1038/srep00571
- Rosenberg, G., Haghnegahdar, P., Goddard, P., Carr, P., Wu, K., and de Prado, M. L. (2016). Solving the optimal trading trajectory problem using a quantum annealer. *IEEE J. Select. Top. Sig. Process.* 10, 1053–1060. doi: 10.1109/JSTSP.2016.2574703
- Takahashi, C., Ohzeki, M., Okada, S., Terabe, M., Taguchi, S., and Tanaka, K. (2018). Statistical-mechanical analysis of compressed sensing for hamiltonian estimation of ising spin glass. *J. Phys. Soc. Jpn.* 87:074001. doi: 10.7566/JPSJ.87.074001
- Tanahashi, K., Takayanagi, S., Motohashi, T., and Tanaka, S. (2019). Application of ising machines and a software development for ising machines. *J. Phys. Soc. Jpn.* 88:061010. doi: 10.7566/JPSJ.88.061010
- Vazirani, V. V. (2013). *Approximation Algorithms*. Springer Science & Business Media.
- Venturelli, D., Dominic, J. M., and Rojo., G. (2015). Quantum annealing implementation of job-shop scheduling. *arXiv: 1506.08479*.
- Wang, Y., Yin, D., Jie, L., Wang, P., Yamada, M., Chang, Y., et al. (2016). “Beyond ranking: optimizing whole-page presentation,” in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (San Francisco, CA), 103–112.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2019 Nishimura, Tanahashi, Suganuma, Miyama and Ohzeki. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

APPENDIX A

Quantum Annealing (QA) on D-Wave 2000Q

QA belongs to a class of meta-heuristic algorithms, which exploit the quantum tunneling effect to efficiently solve an optimization problem (Kadowaki and Nishimori, 1998). The quantum processing unit (QPU) is designed to find the lowest energy state of a spin glass system. This energy state is described by an Ising Hamiltonian:

$$H_P = \sum_{i \in V} h_i s_i + \sum_{(i,j) \in E} J_{ij} s_i s_j,$$

where h_i is the on-site energy of qubit i , and J_{ij} denotes the interaction energies of two qubits i and j . The binary variables $s_i \in \{-1, +1\}$ are called spins, and are fixed in a lattice graph G with vertices and edges (V, E) . Finding the ground state of such a spin glass system, that is, the state with the lowest energy is an NP problem. Therefore, QA can find the solutions of NP problems by mapping them onto spin glass systems. The basic process of QA is to interpolate physically between an initial Hamiltonian H_0 , with an easy-to-implement ground state, and a problem Hamiltonian

H_P , whose minimal configuration needs to be explored. Then, we change the Hamiltonian slowly such that it is the spin glass Hamiltonian at time T :

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \left(\frac{t}{T}\right) H_P.$$

If T is long enough, according to the adiabatic theorem, the system will be in the ground state of the spin glass Hamiltonian H_P .

For computation on D-Wave 2000Q, the problem is first mapped to the Ising binary and quadratic structures. Then, it is embedded in the available qubit lattice. The qubits are arranged according to a chimera graph on D-Wave 2000Q. Each qubit couples to five or six others, except when the qubit has defects. If the problem does not embed directly, auxiliary qubits can be introduced to augment the available couplings. However, introducing auxiliary qubits is a significant cost in qubits. Both mapping and embedding imply restrictions on the types of problems that can effectively be solved with the D-Wave 2000Q. For more details on QA in the D-Wave 2000Q, see D-Wave Systems Inc. (2018).