Solving Inequality-Constrained Binary Optimization Problems on Quantum Annealer

Kouki Yonaga[1,2,3], Masamichi J. Miyama[1,2], and Masayuki Ohzeki[1,2,4]

[1]Graduate School of Information Sciences, Tohoku University, Sendai 980-8579, Japan
[2]Sigma-i Co., Ltd., Tokyo 108-0075, Japan
[3]MathAM-OIL, AIST, Sendai 980-8577, Japan
[4]Institute of Innovative Research, Tokyo Institute of Technology, Yokohama 226-8503

Abstract

We propose a new method for solving binary optimization problems under inequality constraints using a quantum annealer. To deal with inequality constraints, we often use slack variables, as in previous approaches. When we use slack variables, we usually conduct a binary expansion, which requires numerous physical qubits. Therefore, the problem of the current quantum annealer is limited to a small scale. In this study, we employ the alternating direction method of multipliers. This approach allows us to deal with various types using constraints in the current quantum annealer without slack variables. To test the performance of our algorithm, we use quadratic knapsack problems (QKPs). We compared the accuracy obtained by our method with a simulated annealer and the optimization and sampling mode of a D-Wave machine. As a result of our experiments, we found that the sampling mode shows the best accuracy. We also found that the computational time of our method is faster than that of the exact solver when we tackle various QKPs defined on dense graphs.

1 Introduction

Combinatorial optimization problems are essential challenges that emerge in numerous domains such as portfolio optimization [1], traffic flow [2], job-shop scheduling [3], nurse scheduling [4], automated guided vehicles [5], and machine learning [6]. Many researchers have been developing new algorithms to solve these large-sized problems. Quantum annealing (QA) is a recently developed

technology for solving combinatorial optimization problems [7]. This technology was initially proposed in academia, inspired by simulated annealing (SA) [8]. With the recent realization of quantum annealers [9, 10, 11], i.e., D-Wave machines, many researchers have been studying QA for application in industry. Thus, QA is attracting significant attention from numerous people in academia and business. The current D-Wave machine, D-Wave 2000Q, can minimize the following quadratic cost function:

$$E(\boldsymbol{x}) = \boldsymbol{x}^T Q \boldsymbol{x} \tag{1}$$

where $\boldsymbol{x} = \{0,1\}^N$ is an N-dimensional vector of binary variables, and Q is an integer or real matrix. Eq.(1) and Q are called the quadratic unconstrained binary optimization (QUBO) problem and QUBO matrix, respectively. To use the D-Wave 2000Q in practical situations, we represent our task using a QUBO formulation. In this study, we assume that our task is given with the following linear constraints:

$$\begin{array}{lll} \underset{\boldsymbol{x}}{\text{minimize}} & f(\boldsymbol{x}) & \\ \text{,subject to} & \boldsymbol{F}_l \boldsymbol{x} = C_l & (l = 1, \cdots, L) \\ \text{,} & \boldsymbol{G}_m \boldsymbol{x} \leq D_m & (m = 1, \cdots, M), \end{array} \tag{2}$$

where $\boldsymbol{F}_l, \boldsymbol{G}_m \in \mathbb{Z}^N$, $C_l, D_m \in \mathbb{Z}$, and $f(\boldsymbol{x})$ is an objective function given as the QUBO formulation. Here, we represent the equality constraints with the penalty terms as follows [12]:

$$\boldsymbol{F}_l \boldsymbol{x} = C_l \ \ (l = 1, 2, \cdots L) \ \Leftrightarrow \ \sum_{l=1}^{L} \left(\boldsymbol{F}_l \boldsymbol{x} - C_l\right)^2, \tag{3}$$

Then, adding Eq.(3) into $f(\boldsymbol{x})$, we obtain the QUBO-type cost function. In a similar way, the inequality constraints can be written as follows [13, 14, 15]:

$$\boldsymbol{G}_m \boldsymbol{x} \leq D_m \ \ (m = 1, 2, \cdots M) \ \Leftrightarrow \ \sum_{m=1}^{M} \left(\boldsymbol{G}_m \boldsymbol{x} - D_m + s_m\right)^2, \tag{4}$$

where s_m is called the slack variable. Thus, the inequality constraints can be represented using the QUBO formulation by the binary expansion $s_m = 1x_1 + 2x_2 + 4x_3 + \cdots$.

Unfortunately, we can solve only small-sized problems if we apply the slack variables because the binary expansion requires many physical qubits, and D-Wave 2000Q has only approximately 2000 qubits. In addition to the slack variable, the embedding techniques limit the problem size that can be solved. The physical qubits in the D-Wave 2000Q connect to other qubits in the chimera graph. The connection of the hardware, chimera graph, is sparse and differs from that of a logical variable representing the optimization problems. Therefore, we use the embedding technique to represent the logical variables on the chimera

graph [16, 17, 18, 19]. The embedding allows us to solve various QUBO problems but uses numerous additional physical bits. We can compute only 64 logical variables when the problem is defined on a fully connected graph. As a result, the number of logical variables we can use dramatically decreases. Thus, it is difficult for the D-Wave 2000Q to deal with inequality constraints.

In this study, we report a new method for solving inequality-constrained binary optimization problems in the D-Wave 2000Q. Our algorithm is based on the augmented Lagrangian method and the alternating direction method of multipliers (ADMM) [20, 21, 22, 23]. These approaches allow us to solve the inequality constrained problems without the slack variables. Our algorithm applies not only to the D-Wave machine but also to other QUBO solvers. The current digital QUBO solvers can deal with more logical variables than the D-Wave 2000Q. Therefore, with our method, we can solve larger-sized problems involving the inequality constraints.

The remainder of this paper is as follows. In Sec.2, we provide an overview of the QA and D-Wave machine. Furthermore, we show the augmented Lagrangian method and the main algorithm based on the ADMM. In Sec.3, we describe the test results of our method on quadratic knapsack problems. In Sec.4, we compare the accuracy and computation time obtained by our method and exact optimizers. We then discuss the potential superiority of our method over the exact optimizers. Finally, we summarize our study in Sec.5.

2 Methods

2.1 Overview of Quantum Annealing and D-Wave Machine

In QA, we set the system, which consists of the target and driving Hamiltonian [7, 24, 25]. The target Hamiltonian includes the Pauli matrices, whose z-components are given as Ising variables as $+1$ and -1. The target Hamiltonian corresponds to the cost function $E(\boldsymbol{x})$ because the Ising variable s_i can be written as $s_i = 2x_i - 1$. The driving Hamiltonian introduces quantum fluctuations to the system. In the early step of QA, the driving Hamiltonian creates a superposition of all solutions. By gradually reducing the influence of the driving Hamiltonian, we obtain the lowest-cost solution for the target Hamiltonian. Thus, QA achieves the optimal solution if the annealing time is sufficiently long. However, we typically set the annealing time to 20 μs when actually using the D-Wave 2000Q. In addition, it is difficult to remove the effects of noise in the actual system. Therefore, the D-Wave 2000Q is used as a sampler, which provides stochastically approximated solutions [26].

Herein, we introduce postprocessing modes used in the D-Wave 2000Q, i.e., optimization and sampling modes [25]. The optimization model conducts local updates to the samples obtained through QA. Thus, we obtain a set of samples with a lower cost function. In sampling mode, the samples obtain using QA are

modified into a target Boltzmann distribution, which is defined as

$$P(\boldsymbol{x}) = \frac{1}{Z}\exp\left[-\beta E(\boldsymbol{x})\right]. \tag{5}$$

where β is inverse temperature. When we take $\beta \to \infty$, only the lowest-energy samples are obtained. By contrast, when β moves toward zero, diverse samples are generated from $P(\boldsymbol{x})$.

As mentioned in the previous section, we use the embedding technique. Moreover, the unembedding technique is also essential [16]. We use the unembedding to obtain samples on the logical variables after applying QA. The D-Wave 2000Q has several unembedding methods, and the default setting employs the majority-vote method. In this study, we use the minimize-energy method. This method leads us to lower-cost samples by minimizing the local cost function.

2.2 Augmented Lagrangian Method

We define the cost function including the inequality constraints. For simplicity, we consider only the inequality constraints in Eq.(2). The inequality constraints can be written using the penalty terms as follows:

$$E_{\mathrm{ineq}}(\boldsymbol{x}) = f(\boldsymbol{x}) + \gamma \sum_{m=1}^{M} \Theta(\boldsymbol{G}_m\boldsymbol{x} - D_m), \tag{6}$$

where γ is relatively larger than the objective function $f(\boldsymbol{x})$. Here, $\Theta(x)$ is the Heaviside step function, which is defined as follows:

$$\Theta(x) = \begin{cases} 1 & (x > 0) \\ 0 & (x \leq 0). \end{cases}$$

When $\Theta(\boldsymbol{G}_m\boldsymbol{x}^* - D_m)$ is zero for $^\forall m$, $\boldsymbol{x}^*$ is the feasible solution. However, the D-Wave 2000Q cannot directly deal with Eq. (6) because of the Heaviside step function. We introduce the augmented Lagrangian method to transform $E_{\mathrm{ineq}}(\boldsymbol{x})$ into the QUBO formulation [20, 21]. Eq.(6) can be rewritten as follows:

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x}) + \gamma \sum_{m=1}^{M} \Theta(z_m) \\ \text{,subject to} \quad & \boldsymbol{G}_m\boldsymbol{x} - D_m = z_m \qquad (m = 1, \cdots, M), \end{aligned} \tag{7}$$

where $\{z_m\} \in \mathbb{Z}^M$ are auxiliary variables. We obtain the new cost function E_{aug} with the Lagrangian multipliers and the penalty terms as follows:

$$\begin{aligned} E_{\mathrm{aug}}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) &+ \gamma \sum_{m=1}^{M} \Theta(z_m) \\ &+ \sum_{m=1}^{M} \lambda_m(\boldsymbol{G}_m\boldsymbol{x} - D_m - z_m) + \frac{\rho}{2} \sum_{m=1}^{M} (\boldsymbol{G}_m\boldsymbol{x} - D_m - z_m)^2, \end{aligned} \tag{8}$$

where $\{\lambda_m\}$ and ρ are the multipliers and coefficients for the penalty terms, respectively.

2.3 Main Algorithm

To solve $E_{\rm aug}(\boldsymbol{x},\ \boldsymbol{z},$ and $\boldsymbol{\lambda})$, ADMM is widely used [22, 23]. In ADMM, we update $\boldsymbol{x}$, $\boldsymbol{z}$, and the multipliers $\boldsymbol{\lambda}$ by applying the sequential optimizations as follows:

$$\boldsymbol{x}^*[t+1] = \underset{\boldsymbol{x}}{\rm argmin}\ E_{\rm aug}(\boldsymbol{x}, \boldsymbol{z}^*[t+1], \boldsymbol{\lambda}[t]), \tag{9a}$$

$$\boldsymbol{z}^*[t+1] = \underset{\boldsymbol{z}}{\rm argmin}\ E_{\rm aug}(\boldsymbol{x}^*[t+1], \boldsymbol{z}, \boldsymbol{\lambda}[t]]), \tag{9b}$$

$$\lambda[t+1] = \lambda[t] + \rho\left(\boldsymbol{G}_m^T\boldsymbol{x}^*[t+1] - D_m - z_m^*[t+1]\right) \quad (m = 1, \cdots, M), \tag{9c}$$

where t corresponds to the number of iterations. By repeating Eqs.(9a)–(9c) until convergence, we eventually obtain the optimal solution. In this study, we developed a hybrid algorithm that combines ADMM and QA. The main difference between the usual ADMM and our hybrid algorithm is the use of a quantum annealer for solving Eq.(9a). After applying QA for $E_{\rm aug}(\boldsymbol{x},\ \boldsymbol{z},$ and $\boldsymbol{\lambda})$, we obtain the samples $\{\boldsymbol{x}_\nu\}$, where ν is the index for each sample. We define the lowest-cost solution $\boldsymbol{x}^*_{\rm cost}$ that minimizes $E_{\rm aug}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda})$ as

$$\boldsymbol{x}^*_{\rm cost} = \underset{\{\boldsymbol{x}_\nu\}}{\rm argmin}\ E_{\rm aug}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}). \tag{10}$$

Note that $\boldsymbol{x}^*_{\rm cost}$ is not necessarily a feasible solution, but other samples in $\{\boldsymbol{x}_\nu\}$ can be feasible. Here, we define a feasible solution $\boldsymbol{x}^*_{\rm feas}$ that minimizes $f(\boldsymbol{x})$ and satisfies the inequality constraints as follows:

$$\boldsymbol{x}^*_{\rm feas} = \underset{\{\boldsymbol{x}_\nu\}}{\rm argmin}\ f(\boldsymbol{x}) \ \text{ s.t. } \ \boldsymbol{G}_m\boldsymbol{x} \le D_m \ \ (m = 1, \cdots, M). \tag{11}$$

We use $\boldsymbol{x}^*_{\rm cost}$ to update $\boldsymbol{z}$ and $\boldsymbol{\lambda}$, whereas $\boldsymbol{x}^*_{\rm feas}$ is utilized for searching the feasible solution. We show the details of our ADMM algorithm as follows:

1. Initialize the parameters as $\{z_m\} = 0,\ \{\lambda_m\} = 0$, and $t = 1$.
2. Apply the embedding for a fully connected graph with size N.
3. Compute the QUBO matrix using Eq.(8):
4. Obtain the samples $\{\boldsymbol{x}_\nu\}$ by annealing the QUBO matrix.
5. Compute $\boldsymbol{x}^*_{\rm cost}$ and $\boldsymbol{x}^*_{\rm feas}$ using the samples $\{\boldsymbol{x}_\nu\}$.
6. Update $\boldsymbol{z}^*$ as $z_m^* = \min\left(0, \boldsymbol{G}_m\boldsymbol{x}^*_{\rm cost} - D_m\right) \quad (m = 1, \cdots, M)$
7. Update $\boldsymbol{\lambda}$ as $\lambda_m = \lambda_m + \rho\left(\boldsymbol{G}_m\boldsymbol{x}^*_{\rm cost} - D_m - z_m^*\right) \quad (m = 1, \cdots, M)$
8. Check the convergence: When one of the following criteria is satisfied, the calculation is completed.

(a) $t > t_{\max}$

(b) $E_{\text{ineq}}(\boldsymbol{x}^*_{\text{feas}})$ is not improved in t_{conv} steps

(c) $\sqrt{\sum_m \left(\boldsymbol{G}_m \boldsymbol{x}^*_{\text{feas}} - D_m - z_m\right)^2} < \epsilon$

where $t_{\max}$, t_{conv}, and ϵ are predetermined parameters.

9. $t \leftarrow t + 1$.

10. Iterate (3)-(9) until convergence.

Here, the unembedding process is involved in step 4. Thus, the unembedding is applied after every sampling, whereas the embedding is conducted once before the iterating part.

Note the essential points of our ADMM algorithm in the following. First, we use the auxiliary variable $\boldsymbol{z}$ instead of the slack variable $\boldsymbol{s}$. This leads to an efficient utilization of D-Wave 2000Q because the binary expansion is not necessary. Second, we search for the optimal solution using the sampler. Because D-Wave 2000Q is a stochastic sampler, $\boldsymbol{x}^*_{\text{feas}}$ does not necessarily correspond to the optimal solution even when the ADMM is finished. Therefore, we generate many samples using the D-Wave 2000Q and search for a more accurate and feasible solution.

3 Experiments

We tested the performance of our algorithm using the quadratic knapsack problem (QKP), which is defined as follows:

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{maximize}} \quad & \boldsymbol{x}^T P \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{w}^T \boldsymbol{x} \leq c \end{aligned}$$

where $P = \{p_{i,j}\} \in \mathbb{Z}_+^{N \times N}$ is the profit matrix, $\boldsymbol{w} = \{w_i\} \in \mathbb{Z}_+^N$ is the weight vector, and $c \in \mathbb{Z}$ is the capacity. In this study, we randomly generate P and $\boldsymbol{w}$, which was introduced by Gallo *et al.* [27]. The profits $\{p_{i,j}\}$ are zero with probability $(1-\Delta)$, and non-zero values given by a uniform distribution between 1 and 100 with probability Δ. This means that when Δ is close to 1 (zero), the objective function is given by a random dense (sparse) graph. The weights $\{w_i\}$ are also randomly chosen from $[1, 50]$, and the capacity c is taken from a uniform distribution over $[50, \sum_i w_i]$. We generate 10 instances for testing the typical performance of our algorithm. To deal with maximization problems on the D-Wave 2000Q, we define the objective function as $f(\boldsymbol{x}) = -\boldsymbol{x}^T P \boldsymbol{x}$.

To study the accuracy, we define the mean absolute percentage error (MAPE) as follows:

$$\text{MAPE} = \frac{1}{N_{\text{inst}}} \sum_{k=1}^{N_{\text{inst}}} \frac{|f_k(\boldsymbol{x}^*_{\text{opt}}) - f_k(\boldsymbol{x}^*_{\text{feas}})|}{f_k(\boldsymbol{x}^*_{\text{opt}}),} \tag{12}$$

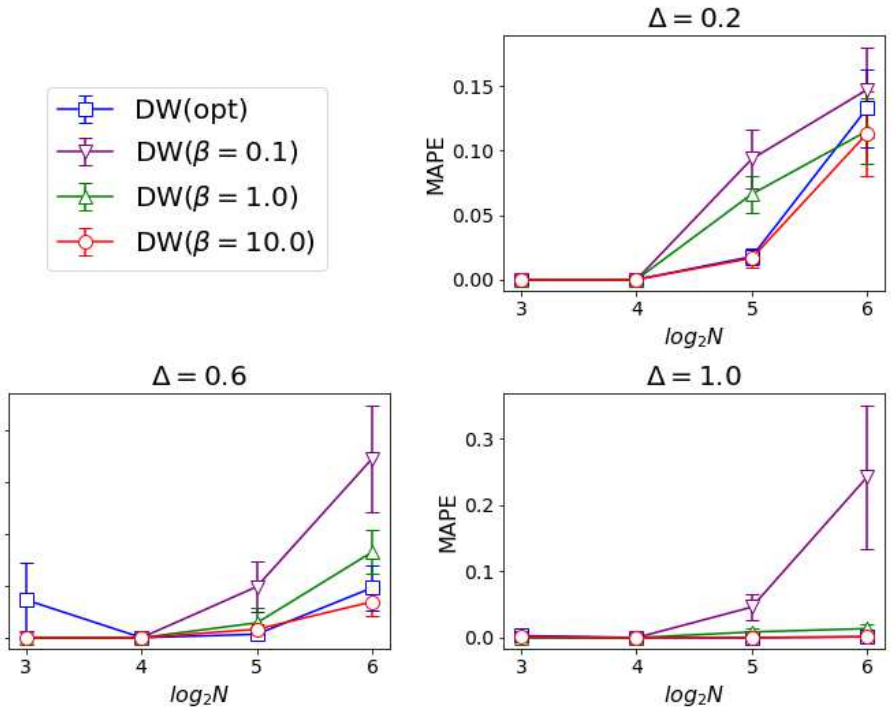

Figure 1: N-dependence of the MAPEs. The squares, lower, upper, and circles correspond to the MAPEs obtained using DW(opt), DW($\beta = 0.1$), DW($\beta = 1.0$), and DW($\beta = 10.0$), respectively.

where $f_k(\boldsymbol{x})$ is the objective function for the kth instance, and $N_{\rm inst}$ is the total number of instances. Here, $\boldsymbol{x}^*_{\rm opt}$ is the optimal solution obtained using the Gurobi optimizer [28], and $\boldsymbol{x}^*_{\rm feas}$ is the feasible solution in the ADMM. Thus, MAPE = 0 corresponds to the ADMM achieving the optimal solutions for all instances. We study the MAPEs obtained using optimization (DW(opt)) and sampling (DW(β)) modes with $\beta = 0.1$, 1.0, and 10.0. During these experiments, we set the annealing time to 20 μs and generate 2000 samples. To check the performance of our algorithm, we calculate the exact solutions using the Gurobi optimizer on a 4-core Intel i7 6700K processor with 64 GB of RAM. We set the maximum calculation time in the Gurobi optimizer to 1000 s. The

predetermined parameters in the ADMM are as follows:

$$\rho = 0.1 \tag{13a}$$
$$t_{\max} = 30 \tag{13b}$$
$$t_{\mathrm{conv}} = 10 \tag{13c}$$
$$\epsilon = 10^{-3}. \tag{13d}$$

Fig.1 shows the N-dependence of the MAPEs in $\Delta = 0.2, 0.6, 1.0$. As shown in Fig.1, we attain the feasible solutions for all instances at up to $N = 64$. The results demonstrate the superiority of our ADMM approach. If we use the slack variables, the D-Wave 2000Q cannot solve the problems at $N = 64$ because of the additional binary variables. The ADMM allows us to deal with larger-size problems on the D-Wave 2000Q than allowed by the previous approach.

We compared the MAPEs obtained by DW(opt), DW($\beta = 0.1$), DW($\beta = 1.0$), and DW($\beta = 10.0$). Fig.1 shows that all MAPEs increase with an increase in N in $\Delta = 0.2$. By contrast, for $\Delta = 1.0$, the MAPEs by DW(opt), DW($\beta = 1.0$), and DW($\beta = 10.0$) remain at near zero even when N increases. These results indicate that the ADMM can accurately find feasible solutions for the QKP on a dense graph. Table 1 shows the Δ-dependence of the MAPEs for $N = 64$. The MAPEs in DW(opt), DW($\beta = 0.1$), DW($\beta = 1.0$), and DW($\beta = 10.0$) decrease as Δ increases. DW($\beta = 10.0$) outperforms the other postprocessing at $N = 64$. In addition, the accuracy of DW(opt) is comparable to that of DW($\beta = 10.0$).

Table 1: Δ-dependence of MAPEs at $N = 64$.

	optimization	$\beta = 0.1$	$\beta = 1.0$	$\beta = 10.0$
$\Delta = 0.2$	0.1329	0.1473	0.1152	0.1133
$\Delta = 0.6$	0.0192	0.0690	0.0330	0.0138
$\Delta = 1.0$	0.0014	0.2423	0.0136	0.0012

The difference in accuracy between the postprocessing modes can be explained by the efficient sampling near the lowest-cost solution $\boldsymbol{x}^*_{\mathrm{cost}}$. Fig.2 shows the histograms for the instance when the ADMM is finished in $(\Delta, N) = (1.0, 64)$. Here, the horizontal axis corresponds to the objective function $f(\boldsymbol{x})$. As can be seen from Fig.2, $\boldsymbol{x}^*_{\mathrm{opt}}$ is near $\boldsymbol{x}^*_{\mathrm{cost}}$, and does not correspond to the one. Therefore, to find an optimal or accurate solution, sampling near $\boldsymbol{x}^*_{\mathrm{cost}}$ is necessary. In fact, DW(opt), DW($\beta = 1.0$), and DW($\beta = 10.0$) have broad histograms located near $\boldsymbol{x}^*_{\mathrm{cost}}$, and are successful in finding $\boldsymbol{x}^*_{\mathrm{opt}}$. For this reason, DW(opt) and DW($\beta = 10.0$) show an accurate performance in our QKP experiments.

Here, we comment on the β-dependence of the sampling mode. If the samples are indeed generated from the Boltzmann distribution, we obtain the widely spread histogram with $\beta = 0.1$. However, DW($\beta = 0.1$) has spike-like distributions that are far from $\boldsymbol{x}_{\mathrm{cost}}$. The reason for this is not clarified because we cannot access the postprocessing on D-Wave 2000Q. Therefore, we should be

careful in tuning β of the sampling mode. From our results, we recommend using DW($\beta = 10.0$) for the QKP.

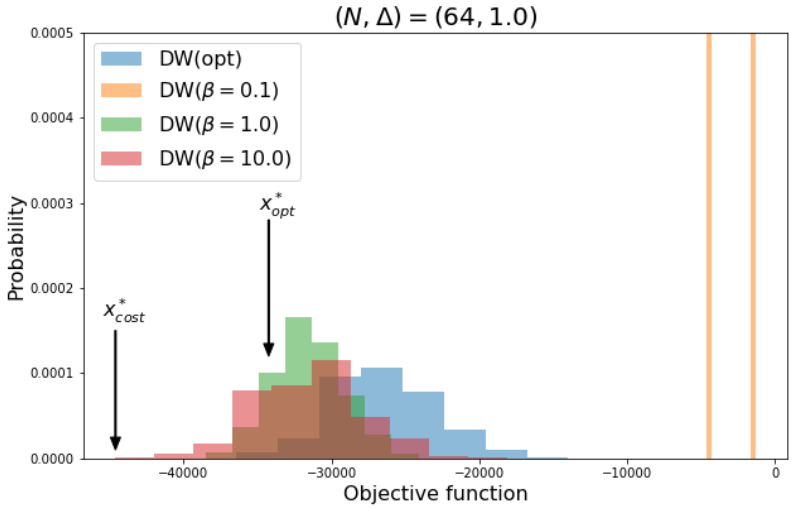

Figure 2: Histograms obtained using DW(opt), DW($\beta = 0.1$), DW($\beta = 1.0$), and DW($\beta = 10.0$). The horizontal axis is the objective function, and we show the vertical axis with only $[0.0, 0.0005]$.

4 Discussion

We discuss improving the accuracy of the ADMM. We obtain $\epsilon_{\text{ave}} > 0.0$ in our experiments, which means that the ADMM cannot achieve the optimal solutions for several instances. A simple way to improve the accuracy is by tuning the value of t_{conv}. In this study, we terminate the ADMM when the feasible solution is not improved in t_{conv} steps. We can obtain more accurate solutions by iterating more ADMM updates. Another way is to generate more samples on the D-Wave 2000Q. Because D-Wave 2000Q is a stochastic sampler, we need many samples to obtain an optimal solution.

We compare the computation times obtained by the ADMM and Gurobi optimizer. We define the total QA, sampling, unembedding with t_{QA}, t_{sampling}, and t_{unemb}, respectively. The t_{QA} value is the total access time for the quantum processing unit on the D-Wave 2000Q. The total sampling time t_{sampling} involves the Internet latency, t_{QA}, and other processing on the D-Wave 2000Q. In addition, t_{unemb} is the total unembedding time in the ADMM steps. We set the total computation times by the ADMM and Gurobi optimizer as t_{ADMM} and t_{Gurobi}, respectively. Here, t_{ADMM} is given as the summation of t_{QA}, t_{sampling}, t_{unemb},

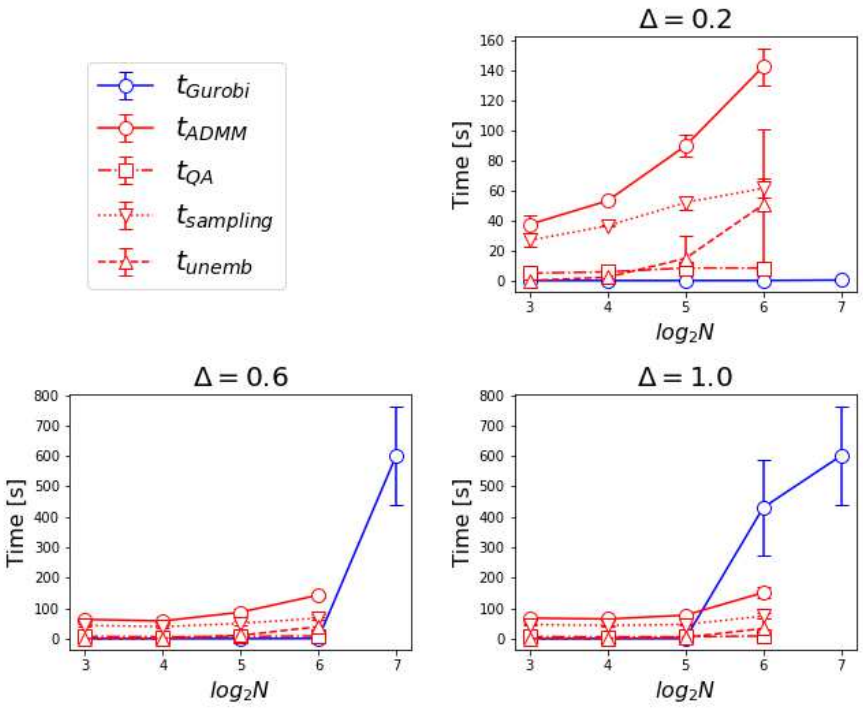

Figure 3: N-dependence of the computation times in $\Delta = 0.2, 0.6, 1.0$. The red and blue circles represent the total computation times by the ADMM and Gurobi optimizer, respectively. We also show the total QA times with the red squares. The red lower and upper triangles correspond to the total sampling and unembedding times, respectively.

and other processes on the CPU. Fig.3 shows the N-dependence of $t_{\rm Gurobi}$ and $t_{\rm ADMM}$. The red and blue circles show the instance-averaged computation time, $t_{\rm ADMM}$ and $t_{\rm Gurobi}$, obtained using the DW($\beta = 10.0$) and Gurobi optimizer, respectively. We conducted a simulation using the Gurobi optimizer at up to $N = 128$. Fig.3 also shows $t_{\rm QA}$, $t_{\rm sampling}$, and $t_{\rm unemb}$ with the red squares and lower and upper triangles, respectively. In $\Delta = 0.2$ and 0.6, the Gurobi optimizer is significantly faster than the ADMM. However, $t_{\rm Gurobi}$ increases dramatically as Δ and N increase. In fact, we obtain $t_{\rm ADMM} < t_{\rm Gurobi}$ in $(\Delta, N) = (1.0, 64)$. Thus, the ADMM can be faster than the exact optimizer with an increase in Δ and N.

Herein, we focus on $t_{\rm QA}$, $t_{\rm sampling}$, and $t_{\rm unemb}$. As can be seen from Fig.3, $t_{\rm sampling}$ and $t_{\rm unemb}$ grow as N increases, whereas $t_{\rm QA}$ remains almost constant. Therefore, $t_{\rm ADMM}$ can be much faster if we reduce the computational overhead, such as $t_{\rm sampling}$ and $t_{\rm unemb}$. In particular, the embedding and unembedding techniques are necessary only when the D-Wave 2000Q implements a sparse graph. Thus, our method can outperform the exact optimizers if a quantum annealer on a larger and denser graph is developed in the future.

5 Summary

In this study, we reported a new algorithm for solving inequality-constrained binary optimization using the D-Wave 2000Q. We defined the new cost function with the augmented Lagrangian method and developed a hybrid algorithm that combines QA and ADMM. We tested the performance of our algorithm for QKP and obtained three significant results. First, our algorithm finds feasible solutions for large-sized problems that cannot be computed using a previous approach. Next, the denser the coupling of the logical variables is, the more accurately the ADMM can find the feasible solutions. Finally, the optimization or sampling mode with $\beta = 10.0$ is appropriate for our ADMM algorithm. We also compare the computation times obtained by the ADMM and the exact optimizer. We show that the ADMM can be faster than the exact optimizer when a QKP is given on a large and dense graph.

References

[1] G. Rosenberg, P. Haghnegahdar, P. Goddard, P. Carr, K. Wu, and M. L. de Prado. Solving the optimal trading trajectory problem using a quantum annealer. *IEEE Journal of Selected Topics in Signal Processing*, 10(6):1053–1060, 2016.

[2] Florian Neukart, Gabriele Compostella, Christian Seidel, David von Dollen, Sheir Yarkoni, and Bob Parney. Traffic flow optimization using a quantum annealer. *Frontiers in ICT*, 4:29, 2017.

[3] D. Venturelli, D. J. J. Marchand, and G Rojo. Quantum annealing implementation of job-shop scheduling. ArXiv e-prints 1506.08479, 2016.

[4] Kazuki Ikeda, Yuma Nakamura, and Travis S. Humble. Application of quantum annealing to nurse scheduling problem. *Scientific Reports*, 9(1):12837, 2019.

[5] Masayuki Ohzeki, Akira Miki, Masamichi J. Miyama, and Masayoshi Terabe. Control of automated guided vehicles without collision by quantum annealer and digital devices. *Frontiers in Computer Science*, 1:9, 2019.

[6] Vivek Dixit, Raja Selvarajan, Muhammad A. Alam, Travis S. Humble, and Sabre Kais. Training and classification using a restricted boltzmann machine on the d-wave 2000q. ArXiv e-prints 2005.03247, 2020.

[7] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Phys. Rev. E*, 58:5355–5363, Nov 1998.

[8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[9] A J Berkley, M W Johnson, P Bunyk, R Harris, J Johansson, T Lanting, E Ladizinsky, E Tolkacheva, M H S Amin, and G Rose. A scalable readout system for a superconducting adiabatic quantum optimization system. *Superconductor Science and Technology*, 23(10):105014, sep 2010.

[10] R. Harris, M. W. Johnson, T. Lanting, A. J. Berkley, J. Johansson, P. Bunyk, E. Tolkacheva, E. Ladizinsky, N. Ladizinsky, T. Oh, F. Cioata, I. Perminov, P. Spear, C. Enderud, C. Rich, S. Uchaikin, M. C. Thom, E. M. Chapple, J. Wang, B. Wilson, M. H. S. Amin, N. Dickson, K. Karimi, B. Macready, C. J. S. Truncik, and G. Rose. Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor. *Phys. Rev. B*, 82:024511, Jul 2010.

[11] M W Johnson, P Bunyk, F Maibaum, E Tolkacheva, A J Berkley, E M Chapple, R Harris, J Johansson, T Lanting, I Perminov, E Ladizinsky, T Oh, and G Rose. A scalable control system for a superconducting adiabatic quantum optimization processor. *Superconductor Science and Technology*, 23(6):065004, apr 2010.

[12] Andrew Lucas. Ising formulations of many np problems. *Frontiers in Physics*, 2:5, 2014.

[13] Tomáš Vyskočil, Scott Pakin, and Hristo N. Djidjev. Embedding inequality constraints for quantum annealing optimization. In Sebastian Feld and Claudia Linnhoff-Popien, editors, *Quantum Technology and Optimization Problems*, pages 11–22, Cham, 2019. Springer International Publishing.

[14] Cristian S. Calude, Elena Calude, and Michael J. Dinneen. Guest column: Adiabatic quantum computing challenges. *SIGACT News*, 46(1):40–61, March 2015.

[15] Dawid Tomasiewicz, Maciej Pawlik, Maciej Malawski, and Katarzyna Rycerz. Foundations for workflow application scheduling on d-wave system. In Valeria V. Krzhizhanovskaya, Gábor Závodszky, Michael H. Lees, Jack J. Dongarra, Peter M. A. Sloot, Sérgio Brissos, and João Teixeira, editors, *Computational Science – ICCS 2020*, pages 516–530, Cham, 2020. Springer International Publishing.

[16] D-Wave Systems Inc. D-wave ocean software documentation. https://docs.ocean.dwavesys.com/en/stable/index.html.

[17] Jun Cai, William G. Macready, and Roy Aidan. A practical heuristic for finding graph minors. ArXiv e-prints 1406.2741, 2014.

[18] Christine Klymko, Blair D. Sullivan, and Travis S. Humble. Adiabatic quantum programming: minor embedding with hard faults. *Quantum Information Processing*, 13(3):709–729, 2014.

[19] Shuntaro Okada, Masayuki Ohzeki, Masayoshi Terabe, and Shinichiro Taguchi. Improving solutions by embedding larger subproblems in a d-wave quantum annealer. *Scientific Reports*, 9(1):2098, 2019.

[20] Magnus R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

[21] Powell M.J.D. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298, New York, 1969. Academic Press.

[22] R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 9(R2):41–76, 1975.

[23] Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17 – 40, 1976.

[24] Tameem Albash and Daniel A. Lidar. Adiabatic quantum computation. *Rev. Mod. Phys.*, 90:015002, Jan 2018.

[25] D-Wave Systems Inc. D-wave system documentation. https://docs.dwavesys.com/docs/latest/index.html.

[26] Marcello Benedetti, John Realpe-Gómez, Rupak Biswas, and Alejandro Perdomo-Ortiz. Estimation of effective temperatures in quantum annealers for sampling applications: A case study with possible applications in deep learning. *Phys. Rev. A*, 94:022308, Aug 2016.

[27] Gallo G., Hammer P.L., and Simeone B. Quadratic knapsack problems. In Padberg M.W., editor, *Combinatorial Optimization. Mathematical Programming Studies*, volume 12. Springer, Berlin, Heidelberg, 1980.

[28] LLC Gurobi Optimization. Gurobi optimizer reference manual. http://www.gurobi.com, 2020.