Flight Gate Assignment with a Quantum Annealer

Tobias Stollenwerk,[1] Elisabeth Lobe,[1] and Martin Jung[1]

[1]*German Aerospace Center (DLR), Linder Höhe, 51147 Köln, Germany*

Optimal flight gate assignment is a highly relevant optimization problem from airport management. Among others, an important goal is the minimization of the total transit time of the passengers. The corresponding objective function is quadratic in the binary decision variables encoding the flight-to-gate assignment. Hence, it is a quadratic assignment problem being hard to solve in general. In this work we investigate the solvability of this problem with a D-Wave quantum annealer. These machines are optimizers for quadratic unconstrained optimization problems (QUBO). Therefore the flight gate assignment problem seems to be well suited for these machines. We use real world data from a mid-sized German airport as well as simulation based data to extract typical instances small enough to be amenable to the D-Wave machine. In order to mitigate precision problems, we employ bin packing on the passenger numbers to reduce the precision requirements of the extracted instances. We find that, for the instances we investigated, the bin packing has little effect on the solution quality. Hence, we were able to solve small problem instances extracted from real data with the D-Wave 2000Q quantum annealer.

I. INTRODUCTION

Modern airport management requires a more holistic approach to the whole travel chain, aiming at a better situational awareness of airport landside processes and an improved resource management. One main key to achieve this is proactive passenger management [1]. Unlike common reactive approaches, proactive passenger management utilizes an early knowledge about the passengers' status and the expected situation in the terminal along with resulting system loads and resource deployment, e.g. using simulation techniques like in [2]. An appropriate and modern management also considers dependencies of airside and landside operations as well as costs and performance. One part of the proactive passenger management is a proper flight gate assignment. Especially large connecting hub airports have to deal with the still growing demand of traffic and rising numbers of passengers and baggage that needs to be transferred between flights. Although there are several objective functions in the field of airport planning, we focus on reducing the total transit time for passengers in an airport [3]. The main goals are the increase of passenger comfort and punctuality. This problem is related to the quadratic assignment problem, a fundamental problem in combinatorial optimization whose standard formulation is NP-hard [4].

In general, most planning problems belong to the class of discrete optimization problems which are hard to solve classically. Therefore it is worth studying new hardware architectures, like quantum computers, which may outperform classical devices. The first commercially available quantum annealer, developed by the Canadian company D-Wave Systems, is a heuristic solver using quantum annealing for optimizing quadratic functions over binary variables without further constraints (QUBOs). A wide range of combinatorial optimization problems can be brought into such a QUBO format by standard transformations [5]. But usually these transformations produce overhead, e.g. by increasing the number of variables, the required

connections between them or the value of the coefficients. It is important to assess the impact of this overhead in order to estimate the performance of future devices [6]. In this work, we investigate the solvability of the optimal flight gate assignment problem with a D-Wave 2000Q quantum annealer.

The paper is structured as follows: In section II we formally introduce the problem preparing the mapping to QUBO, which is done in section III. The extraction of smaller, but representative problem instances is covered in section IV. In section V we present our results for solving these smaller problem instances with a D-Wave 2000Q quantum annealer.

II. FORMAL PROBLEM DEFINITION

Flight gate assignment was already addressed in different versions [7][8]. The following mainly corresponds to the formulation from [3] as a quadratic binary program with linear constraints.

A. Input Parameters

The typical passenger flow in an airport can usually be divided into three parts: After the airplane has arrived at the gate, one part of the passengers passes the baggage claim and leaves the airport. Other passengers stay in the airport to take connecting flights. These transit passengers can take up a significant fraction of the total passenger amount. The third part are passengers which enter the airport through the security checkpoint and leave with a flight. The parameters of the problem are summarized in table I.

TABLE I. Input data for a flight gate assignment instance

F	Set of flights ($i \in F$)
G	Set of gates ($\alpha \in G$)
n_i^{dep}	No. of passengers departing with flight i
n_i^{arr}	No. of passengers arriving with flight i
n_{ij}	No of transfer passengers between flights i and j
t_α^{arr}	Transfer time from gate α to baggage claim
t_α^{dep}	Transfer time from check-in to gate α
$t_{\alpha\beta}$	Transfer time from gate α to gate β
t_i^{in}	Arrival time of flight i
t_i^{out}	Departure time of flight i
t^{buf}	Buffer time between two flights at the same gate

B. Variables and Objective

Assignment problems can easily be represented with binary variables indicating whether or not a resource is assigned to a certain facility. The variables form a matrix indexed over the resources and the facilities. The binary decision variables are $x \in \{0,1\}^{F \times G}$ with

$$x_{i\alpha} = \begin{cases} 1, & \text{if flight } i \in F \text{ is assigned to gate } \alpha \in G, \\ 0, & \text{otherwise.} \end{cases}$$

Like already stated, the passenger flow divides into three parts and so does the objective function: The partial sums of the arriving, the departing and the transfer passengers sum up to the total transfer time of all passengers. For the arrival part we get a contribution of the corresponding time t_α^{arr} for each of the n_i^{arr} passengers if flight i is assigned to gate α. Together with the departure part, which is obtained analogously, the linear terms of the objective are

$$T^{\text{arr/dep}}(x) = \sum_{i\alpha} n_i^{\text{arr/dep}} t_\alpha^{\text{arr/dep}} x_{i\alpha}.$$

The contribution of the transfer passengers is the reason for the hardness of the problem: Only if flight i is assigned to gate α and flight j to gate β the corresponding time is added. This results in the quadratic term

$$T^{\text{trans}}(x) = \sum_{i\alpha j\beta} n_{ij} t_{\alpha\beta}\, x_{i\alpha} x_{j\beta}.$$

The total objective function is

$$T(x) = T^{\text{arr}}(x) + T^{\text{dep}}(x) + T^{\text{trans}}(x). \qquad (1)$$

C. Constraints

Not all binary encodings for x form valid solutions to the problem. There are several further restrictions which need to be added as constraints. In this model a flight corresponds to a single airplane arriving and departing at a single gate. It is obvious, that every flight can only be assigned to a single gate, therefore we have

$$\sum_\alpha x_{i\alpha} = 1 \qquad \forall i \in F. \qquad (2)$$

Furthermore it is clear that no flight can be assigned to a gate which is already occupied by another flight at the same time. These forbidden flight pairs with overlapping time slots can be aggregated in

$$P = \left\{ (i,j) \in F^2 : t_i^{\text{in}} < t_j^{\text{in}} < t_i^{\text{out}} + t^{\text{buf}} \right\}.$$

The resulting linear inequalities $x_{i\alpha} + x_{j\alpha} \leq 1$ are equivalent to the quadratic constraints

$$x_{i\alpha} \cdot x_{j\alpha} = 0 \qquad \forall (i,j) \in P \ \forall \alpha \in G.$$

III. MAPPING TO QUBO

QUBOs are a special case of integer programs minimizing a quadratic objective function over binary variables $x \in \{0,1\}^n$. The standard format is the following

$$q(x) = x^\top Q x = \sum_{j=1}^{n} Q_{jj} x_j + \sum_{\substack{j,k=1 \\ j<k}}^{n} Q_{jk} x_j x_k \qquad (3)$$

with an upper-triangular quadratic matrix $Q \in \mathbb{R}^{n \times n}$. While the presented objective function already follows this format the constraints need to be reduced which is shown in this section.

A. Penalty Terms

The standard way to reduce constraints, like already shown in e.g. [6], is to introduce terms penalizing those variable choices that violate the constraints. Just in these cases a certain positive value is added to the objective function to favor valid configurations while minimizing. The quadratic terms

$$C^{\text{one}}(x) = \sum_i \left(\sum_\alpha x_{i\alpha} - 1 \right)^2$$

and

$$C^{\text{not}}(x) = \sum_\alpha \sum_{(i,j) \in P} x_{i\alpha} x_{j\alpha}$$

fulfill

$$C^{\text{one/not}} \begin{cases} > 0, & \text{if constraint is violated,} \\ = 0, & \text{if constraint is fulfilled,} \end{cases}$$

and therefore are suitable penalty terms which can be combined with the objective function. Since the benefit in the objective function in case of an invalid variable choice should not exceed the penalty, two parameters $\lambda_{\mathrm{one}}, \lambda_{\mathrm{not}} \in \mathbb{R}_+$ need to be introduced:

$$q(x) = T(x) + \lambda^{\mathrm{one}} C^{\mathrm{one}}(x) + \lambda^{\mathrm{not}} C^{\mathrm{not}}(x).$$

In theory these parameters could be set to infinity, but in practice this is not possible and they have to be chosen carefully.

B. Choosing Penalty Weights

The parameters λ^{one} and λ^{not} need to be large enough to ensure that a solution always fulfills the constraints. However due to precision restrictions of the D-Wave machine it is favorable to choose them as small as possible. In this section we present two different possibilities to obtain suitable values. Besides a worst case analysis for each single constraint a bisection algorithm which iteratively checks penalty weights against the solution validity is presented. In the following we call a variable to be activated if it is set to one.

1. Worst Case Estimation

The minimal contribution of C^{one} and C^{not} when breaking the corresponding constraint is just one. Therefore the corresponding minimal penalties are λ^{one} and λ^{not}. Since the objective function and C^{not} only contain non-negative coefficients, just C^{one} enforces some of the variables to be set to one. Therefore activating more than one $x_{i\alpha}$, for one flight i, does not improve the objective value. Hence λ^{one} just needs to exceed the benefit of deactivating a single variable in a valid solution. This variable usually appears in several summands of the objective function. Hence in the worst case the objective could be reduced by the sum of all coefficients of monomials including this variable, which is

$$T_{i\alpha}(x) := n_i^{\mathrm{dep}} t_\alpha^{\mathrm{dep}} + n_i^{\mathrm{arr}} t_\alpha^{\mathrm{arr}} + \sum_j n_{ij} \sum_\beta t_{\alpha\beta} x_{j\beta}.$$

Assuming the penalty is chosen large enough it will also be sufficient for all gates β appearing in $T_{i\alpha}$. Therefore in the last part of the sum, for every flight j, just one gate β needs to be taken into account for which $x_{j\beta}$ is one. In the worst case this is the one with the maximal time $t_{\alpha\beta}$, which results in

$$T^{\mathrm{one}} := \max_{i,\alpha} \left(n_i^{\mathrm{dep}} t_\alpha^{\mathrm{dep}} + n_i^{\mathrm{arr}} t_\alpha^{\mathrm{arr}} + \max_\beta t_{\alpha\beta} \sum_j n_{ij} \right).$$

Considering C^{not}, it is not preferable to add an additional flight pair that is forbidden. But if the penalty is not large enough, assigning one flight i to a gate γ

although this gate is already occupied by another flight j rather than using the different gate α could reduce the objective value. This means $x_{i\gamma} = 1$ might be preferred to $x_{i\alpha} = 1$ although $(i, j) \in P$ and $x_{j\gamma} = 1$. The resulting benefit can be calculated from the difference of $T_{i\alpha}$ and $T_{i\gamma}$. For the estimation of the worst case this can be simplified by taking the maximum transfer time for $T_{i\alpha}$ and the minimum for $T_{i\gamma}$:

$$
\begin{aligned}
T^{\mathrm{not}} := \max_{i,\alpha,\gamma} \Bigg(& \left(n_i^{\mathrm{dep}} t_\alpha^{\mathrm{dep}} + n_i^{\mathrm{arr}} t_\alpha^{\mathrm{arr}} + \max_\beta t_{\alpha\beta} \sum_j n_{ij} \right) \\
& - \left(n_i^{\mathrm{dep}} t_\gamma^{\mathrm{dep}} + n_i^{\mathrm{arr}} t_\gamma^{\mathrm{arr}} + \min_\beta t_{\gamma\beta} \sum_j n_{ij} \right) \Bigg) \\
= \max_{i,\alpha,\gamma} \Bigg(& \left(n_i^{\mathrm{dep}} t_\alpha^{\mathrm{dep}} - n_i^{\mathrm{dep}} t_\gamma^{\mathrm{dep}} \right) + \left(n_i^{\mathrm{arr}} t_\alpha^{\mathrm{arr}} - n_i^{\mathrm{arr}} t_\gamma^{\mathrm{arr}} \right) \\
& + \max_\beta \left(t_{\alpha\beta} - t_{\gamma\beta} \right) \sum_j n_{ij} \Bigg).
\end{aligned}
$$

All in all using $\lambda^{\mathrm{one}} = T^{\mathrm{one}} + \varepsilon$ and $\lambda^{\mathrm{not}} = T^{\mathrm{not}} + \varepsilon$ for some $\varepsilon > 0$ ensures that the minimum of q satisfies the constraints. The provided boundaries can be calculated easily, but unfortunately they may take pretty large values depending on the given parameters. Since usually the worst case is also not the most probable it is a very rough estimation and there is some room for improvement.

2. Bisection Method

If the constraints are independent of each other, a simple bisection method can be used to find an approximation of the boundary between valid and invalid penalty values. In the course of this, all but one penalty is fixed, while the bisection is employed in one dimension at a time. These fixed penalty values as well as an upper starting point of the bisection method need to yield valid solutions. For given values of the penalties, a classical solver like SCIP is used to find the exact solution [9]. But it is imaginable that also the D-Wave machine itself could be used. For very small instances it might happen that one of the constraints becomes redundant and therefore is always fulfilled if the other one is. In these cases just the remaining constraint needs to be evaluated. This method, may lead to much smaller values than the worst case estimation. However it requires solving the problem several times exactly for each instance. Therefore this approach is not viable in an operational setting.

IV. INSTANCE EXTRACTION AND GENERATION

From [2], applying agent based simulation techniques of [10], we extract the data to estimate the transfer times t^{buf}, t_α^{dep}, t_α^{arr} $\forall \alpha \in G$ and $t_{\alpha\beta}$ $\forall (\alpha, \beta) \in G^2$. The second part of our data source consists of a flight schedule of

a mid-sized German airport for a full day. This gives us the number of passengers n_i^{dep}, n_i^{arr} $\forall i \in F$ and n_{ij} $\forall(i,j) \in F^2$. The resulting problem instance is depicted in figure 1.
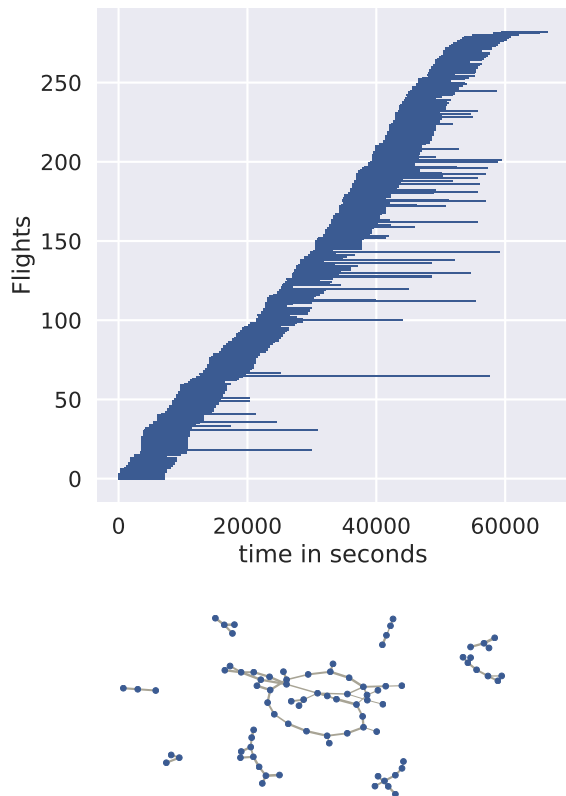

FIG. 1. Top: Time at the airport for all flights during a whole day on a mid-sized German airport. Bottom: Graph representing the transfer passengers for a whole day. Each vertex is a flight, each edge are transfer passengers between two flights.

A. Preprocessing

Since the given data set contains a slice of a flight schedule cut off at one day, there are flights which either do not have an arrival or a departure time. These flights are removed. Furthermore for some of the airplanes, the standing time is so long that it is reasonable to assume the airplane is moved away from the gate for some time before returning to a possibly different gate. Hence, we considered all flights with a time at the airport above 120 minutes as two separate flights. In order to extract typical, hard instances from the data, we removed all flights with no transfer passengers. The remaining instance now consists of 89 flights with 80 transfers and over 35 gates. However, since this corresponds to more than 3000 binary variables further subdivision is needed

to make the problem amenable to the current D-Wave 2000Q hardware.

A way to achieve this becomes apparent when visualizing the dependencies among flights with transfer passengers in a graph as in the bottom part of figure 1. It can be observed that the graph is divided into several connected components, various smaller ones with 3 to 11 flights and a single larger one. The time intervals are much more distributed. Therefore extracting these special subgraphs provides suitable instances to be tested. In addition to using the connected components of the transfer passenger graph, we randomly cut the largest connected component to create larger test instances. This results in 163 instances with number of flights and gates from 3 to 16 and 2 to 16, respectively. The set of these instances will be denoted by $\mathcal{I}_{\mathrm{CC}}$.

An alternative to using only flights with transfer passengers, is to use only flights within a certain time interval. However, this option has two major disadvantages: In such a time slot most of the flight intervals overlap mutually and therefore each flight requires a different gate. Also, in our data set, these flights do have almost no transferring passengers. Both issues simplify the subproblems severely. Therefore this alternative is not pursued in this work.

B. Bin Packing

As we will see, some of the instances from $\mathcal{I}_{\mathrm{CC}}$ are small enough to fit on the D-Wave 2000Q machine. However, they exhibit a strong spread of the coefficients in the QUBO. It is known that this can suppress the success probability due to the limited precision of the D-Wave machine [11]. Therefore, we tried to reduce the spread of the coefficients while retaining the heart of the problem as best as possible. First, we mapped the passenger numbers to natural numbers in $\{0, 1, \ldots, N_{\mathrm{p}}\}$ with bin packing, where N_{p} is the number of bins. Moreover we mapped the transfer times to random natural numbers from $\{1, \ldots, N_{\mathrm{t}}\}$. This is reasonable, since the original time data was drawn from a simulation data, which showed similar behavior. The mapping of the maximum transfer time in the instance to N_{t} introduces a scaling factor to the objective function, which is irrelevant for the solution. In order to assess the impact of the bin packing for the number of passengers, we solved each problem before and after bin packing with the exact solver SCIP. Let the solutions before and after bin packing be denoted as $\mathbf{x}$ and $\hat{\mathbf{x}}$. The approximation ratio is then given by the $R = \frac{T(\hat{\mathbf{x}})}{T(\mathbf{x})}$. Where T is the objective function (1) before bin packing. If $R = 1$, the bin packing has no effect on the solution quality. Figure 2 shows the approximation ratio for various bin packed instances. We used values of $N_{\mathrm{p}} \in \{2, 3, 6, 7, 8, 9, 10\}$ and $N_{\mathrm{t}} \in \{2, 3, 6, 10\}$ and all combinations thereof. Moreover, we restrict ourselves to instances with $|F| \cdot |G| < 100$. One can see, that the approximation ratio is close to one for the majority of the bin packed instances. Meaning,
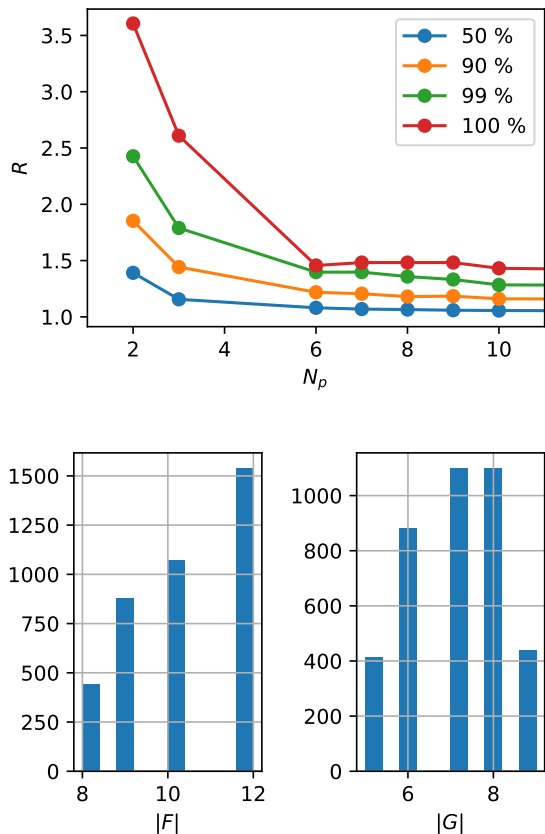
FIG. 2. Top: Approximation ratio for bin packed instances. The different curves show percentiles. Bottom: Distribution of number of flights and gates $|F|$, $|G|$.

the bin packing has little effect on the solution quality, at least for the instances we investigated.

For the investigation of quantum annealing, we restrict ourselves to $N_\mathrm{p}, N_\mathrm{t} \in \{2, 3, 6, 10\}$. The corresponding instances will be denoted by $\mathcal{I}_\mathrm{BP}$.

V. QUANTUM ANNEALING

A. Embedding

The hardware layout of the D-Wave 2000Q quantum annealer restricts the connections between binary variables to the so called Chimera graph [12]. In order to make problems with higher connectivity amenable to the machine, we employ minor-embedding [13]. This includes coupling various physical qubits together into one logical qubit, representing one binary variable, with a strong ferromagnetic coupling J_F in order to ensure that all physical qubits have the same value after readout [14]. Since the constraint (2) introduces $|F|$ complete graphs of size $|G|$, we expect at most quadratic increase in the number of physical qubits with the number of logical qubits, which

is $|F| \cdot |G|$. This is supported by our findings in the top part of figure 3. We calculated five different embeddings for each problem instance from $\mathcal{I}_\mathrm{CC}$. With this, we were able to embed instances up to 84 binary variables.

B. Precision Requirements

The D-Wave 2000Q has a limited precision in specifying the linear and quadratic coefficients of the Ising model. As it was shown in [11], this can be an inhibiting factor for solving some problems on the D-Wave 2000Q. In order to assess the precision requirements for each instance, we introduce the maximum coefficient ratio for a QUBO like (3) as

$$C_\mathrm{QUBO} = \frac{\max_{ij} |Q_{ij}|}{\min_{ij} |Q_{ij}|},$$

and for the corresponding embedded Ising model $\sum_i h_i s_i + \sum_{ij} J_{ij} s_i s_j$ as

$$C_\mathrm{Ising} = \max \left\{ \frac{\max_i |h_i|}{\min_i |h_i|}, \frac{\max_{ij} |J_{ij}|}{\min_{ij} |J_{ij}|} \right\}.$$
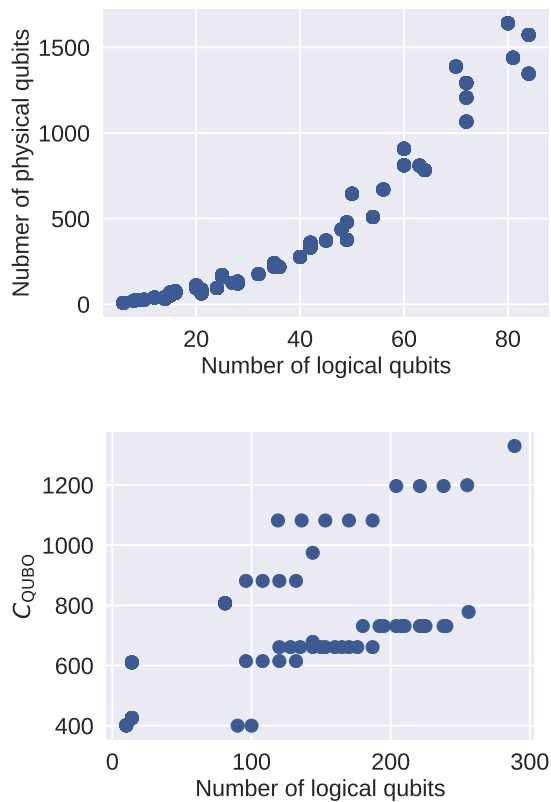

FIG. 3. Top: Number of logical and physical qubits on D-Wave 2000Q for instances $\mathcal{I}_\mathrm{CC}$. Bottom: Maximum coefficient ratio of the QUBOs for instances $\mathcal{I}_\mathrm{CC}$.

The bottom part of figure 3 shows the maximum coefficient ratio for all QUBO from $\mathcal{I}_{\mathrm{CC}}$. We used minimal sufficient penalty weights calculated by bisection as it was described in section III B 2. The corresponding values for the embedded Ising model are orders of magnitude larger (not shown). Therefore the success probability for these instances is highly suppressed. However, this does not seem to be sufficient to reduce the precision requirements of the instances to an acceptable level. In order to mitigate the problem, we will use the bin packed instances $\mathcal{I}_{\mathrm{BP}}$ for the remainder of this work.

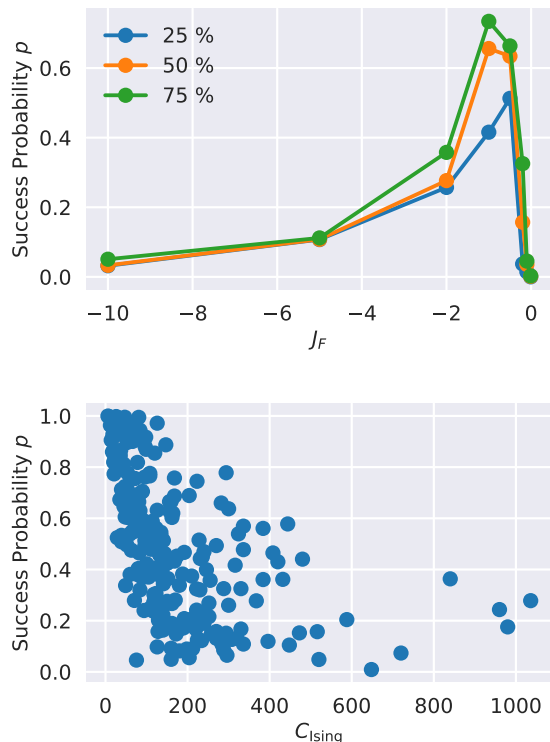## C. Quantum Annealing Results

FIG. 4. Top: Success probability against the intra-logical qubit coupling in units of the largest coefficient of the embedded Ising model. The data is for a representative instance from $\mathcal{I}_{\mathrm{BP}}$. The different colors represent the 25th, 50th and 75th percentiles. Bottom: Maximum success probability against C_{Ising} for the instances from $\mathcal{I}_{\mathrm{BP}}$.

We used all embeddable instances from $\mathcal{I}_{\mathrm{BP}}$ for investigating the performance of the D-Wave 2000Q. As penalty weights, we used $\lambda_{\mathrm{one}} = f^{\mathrm{one}} T^{\mathrm{one}}$ and $\lambda_{\mathrm{not}} = f^{\mathrm{not}} T^{\mathrm{not}}$, with $f^{\mathrm{one}}, f^{\mathrm{not}} \in \{\frac{1}{2}, 1\}$, if the corresponding exact solution was valid. Note, that this is always the case for the worst case estimation $f^{\mathrm{one}} = f^{\mathrm{not}} = 1$. Again, we used 5 different embeddings for each problem instance. The annealing solutions were obtained using 1000 annealing runs, no gauging and majority voting as a un-embedding

FIG. 5. Top: Success probability against the number of flights for the instances from $\mathcal{I}_{\mathrm{BP}}$. Bottom: Maximum C_{Ising} against the number of flights for the instances from $\mathcal{I}_{\mathrm{BP}}$. The different colors represent the 25th, 50th and 75th percentiles.

strategy for broken chains of logical qubits. In order to calculate the time-to-solution with 99% certainty we use

$$T_{99} = \frac{\ln(1 - 0.99)}{\ln(1 - p)} T_{\mathrm{anneal}},$$

where T_{anneal} is the annealing time, which we fixed to $20\mu s$, and p is the success probability. The latter is calculated by the ratio of the number of runs where the optimal solution was found to the total number of runs. The best annealing solution was compared to an exact solution obtained with a MaxSAT solver [15]. As intra-logical qubit coupling we used $J_F = -1$ in units of the largest coefficient of the embedded Ising model of the problem instance at hand. The top part of figure 4 shows the dependence of the success probability on the choice of J_F for a single instance from $\mathcal{I}_{\mathrm{BP}}$. As expected, for large J_F the success probability is suppressed due to increased precision requirements and for very small J_F the logical qubit chains are broken (cf. [11]). The former is substantiated by the bottom part of figure 4, where the success probability in dependence of C_{Ising} is shown for a fixed $J_F = -1$ in units of the largest coefficient of the embedded Ising model. The success probability is suppressed for larger values of C_{Ising} due to the increased precision requirements.

The top part of figure 5 shows the time-to-solution in

dependence of the number of flights. There is an increase in the time-to-solution with the number of flights, and therefore the problem size. This can be explained by the increase in C_{Ising} with the number of flights as it can be seen on the bottom part of figure 5.

VI. CONCLUSION

We showed, that the flight gate assignment problem can be solved with a quantum annealer with some restrictions. First, the size of the amenable problems is very small. Due to the high connectivity of the problem there is a large embedding overhead. Therefore a conclusive assessment of the scaling behavior is not possible at the moment. Future generations of quantum annealers with more qubits and higher connectivity are needed to investigate larger problems. Second, extracting problem instances directly from the data can lead to distributed coefficients in the resulting QUBOs. As a result, the success probability is mostly suppressed for these instances due to their high precision requirements. However, bin packing the coefficients can strongly decrease the precision requirements while retaining the heart of the problem.

For future work we leave the investigation of hybrid algorithms in order to recombine partial solutions and solve the whole problem, like in [16], as well as the further investigation of the influence of bin packing on the solution quality.

VII. ACKNOWLEDGMENTS

[1] A. B. Classen and F. Rudolph, in *Transportation Research Forum - TRF 2015* (2015), pp. 1–19, URL https://elib.dlr.de/96079/.

[2] M. Jung, A. B. Classen, F. Rudolph, A. Pick, and U. Noyer, in *Winter Simulation Conference* (IEEE, 2017), pp. 2518–2529.

[3] S. H. Kim, E. Feron, J.-P. Clarke, A. Marzuoli, and D. Delahaye, *Airport gate scheduling for passengers, aircraft, and operations*, Journal of Air Transportation **25**, 109 (2017).

[4] M. R. Garey and D. S. Johnson, *Computers and intractability*, vol. 29 (wh freeman New York, 2002).

[5] P. L. Hammer and S. Rudeanu, *Boolean methods in operations research and related areas*, vol. 7 (Springer Science & Business Media, 2012).

[6] E. G. Rieffel, D. Venturelli, B. O'Gorman, M. B. Do, E. M. Prystay, and V. N. Smelyanskiy, *A case study in programming a quantum annealer for hard operational planning problems*, Quantum Information Processing **14**, 1 (2015).

[7] A. Haghani and M.-C. Chen, *Optimizing gate assignments at airport terminals*, Transportation Research Part A: Policy and Practice **32**, 437 (1998).

[8] R. Mangoubi and D. F. Mathaisel, *Optimizing gate assignments at airport terminals*, Transportation Science **19**, 173 (1985).

[9] A. Gleixner, L. Eifler, T. Gally, G. Gamrath, P. Gemander, R. L. Gottwald, G. Hendel, C. Hojny, T. Koch, M. Miltenberger, et al., Technical Report, Optimization Online (2017), URL http://www.optimization-online.org/DB_HTML/2017/12/6385.html.

[10] E. Bonabeau, *Agent-based modeling: Methods and techniques for simulating human systems*, vol. 99 (National Acad Sciences, 2002).

[11] T. Stollenwerk, B. O'Gorman, D. Venturelli, S. Mandrà, O. Rodionova, H. K. Ng, B. Sridhar, E. G. Rieffel, and R. Biswas, *Quantum annealing applied to deconflicting optimal trajectories for air traffic management*, arXiv preprint (2017), URL https://arxiv.org/abs/1711.04889.

[12] C. C. McGeoch and C. Wang, in *Proceedings of the ACM International Conference on Computing Frontiers* (ACM, 2013), p. 23.

[13] J. Cai, W. G. Macready, and A. Roy, *A practical heuristic for finding graph minors*, arXiv preprint (2014), URL http://arxiv.org/abs/1406.2741.

[14] D. Venturelli, D. J. J. Marchand, and G. Rojo, *Quantum Annealing Implementation of Job-Shop Scheduling*, arXiv preprint (2015), URL http://arxiv.org/abs/1506.08479.

[15] A. Kügel, *Improved exact solver for the weighted max-sat problem*, Pos@ sat **8**, 15 (2010), URL https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.190/Mitarbeiter/kuegel/maxsat.pdf.

[16] T. T. T. M. Do, E. G. Rieffel, J. Frank, Z. Wang, B. O'Gorman, D. Venturelli, and J. C. Beck, *A hybrid quantum-classical approach to solving scheduling problems*, Ninth Annual Symposium on Combinatorial Search (2016).