Quantum Annealing for Air Traffic Management

Tobias Stollenwerk
German Aerospace Center, Linder Höhe, 51147 Cologne, Germany

Bryan O'Gorman
Berkeley University
NASA Ames Research Center Quantum Artificial Intelligence Laboratory (QuAIL), Mail Stop 269-1, 94035 Moffett Field CA and Stinger Ghaffarian Technologies Inc., 7701 Greenbelt Rd., Suite 400, Greenbelt, MD 20770

Davide Venturelli
NASA Ames Research Center Quantum Artificial Intelligence Laboratory (QuAIL), Mail Stop 269-1, 94035 Moffett Field CA and USRA

Eleanor G. Rieffel
NASA Ames Research Center Quantum Artificial Intelligence Laboratory (QuAIL), Mail Stop 269-1, 94035 Moffett Field CA

Salvatore Mandrà
NASA Ames Research Center Quantum Artificial Intelligence Laboratory (QuAIL), Mail Stop 269-1, 94035 Moffett Field CA and Stinger Ghaffarian Technologies Inc., 7701 Greenbelt Rd., Suite 400, Greenbelt, MD 20770

Olga Rodionova, Hok K. Ng and Banavar Sridhar
tbd

(Dated: May 15, 2017)

In this paper we present the mapping of air traffic management (ATM) problem on quadratic unconstrained boolean optimization (QUBO) problem. After the representation of the ATM problem in terms of a conflict graph, where nodes of the graph represent flights and edges represent a potential conflict between flights, we proceed by discretize the ATM problem and then mapping it in binary variables. As part of our study, we tested the QUBO formulation of the ATM problem using both classical solvers and the D-Wave 2X quantum chip.

CONTENTS

I. INTRODUCTION

Efficiently automating air traffic management is increasingly important (increased volume and diversity, environmental concerns, etc.).

Quantum annealing is a promising computational method.

We investigate the feasibility of applying quantum annealing to a particular problem in air traffic management known as "deconficting", in which the goal is to modify a set of independently optimal trajectories in a way that removes conflicts between them while minimizing the cost of doing so.

II. PROBLEM SPECIFICATION

Formally, an instance of the deconflicting problem is a set of ideal trajectories

The problem at hand is the deconflicting of transatlantic wind-optimal trajectories. As it was done in [1] we are using the same wind-optimal trajectories of a single day, July 29 2012. These wind-optimal trajectories are given as $(\mathbf{x}_i)_{i=1}^{n}$, where $\mathbf{x}_i = (x_{i,t})_{t=t_{i,0}}^{t_{i,1}}$ and $x_{i,t}$ is the location (as latitude, longitude, and altitude) of the ith flight at time t. The times $t_{i,0}$ and $t_{i,1}$ are the times at which the

wind-optimal trajectory for the ith flight begins and ends, respectively. Furthermore, the times are given in units of one minutes $T_i = (t_{i,0}, t_{i,0}+1, \ldots, t_{i,1})$. Each flight i is at a constant speed v_i, to within (classical) machine precision.

A conflict between two flights is defined as a pair of trajectory points which are too close to each other in space and time.

$$\{(x_{i,t}, x_{j,t'}) \mid \mathcal{D}(x_{i,t}, x_{j,t'}) < \Delta_x, |t-t'| < \Delta_t\} \ , \quad (1)$$

where $\mathcal{D}(x,y)$ is the spatial distance between two points x and y given as latitude, longitude and altitude. Following [1], the space threshold is $\Delta_x = 3$ nautical miles and the time threshold is $\Delta_t = 3$ minutes. In this paper, we consider the following means to deconflict the trajectories: First, we can delay each flight i at departure time by a departure delay d_i

$$x_{i,t} \to x_{i,t+d_i} \quad \forall\, t \in T_i$$

Second, we can avoid a conflict by maneuvers of both involved flights. We assume, however, that the maneuvers will not introduce new conflicts. In doing so, these maneuvers can be view as resulting in time shifts only.

A. Potential Conflicts

It is beneficial to reduce the data to conflicting regions in space and decoupling the spacial and temporal components of the problem. As a first step, we detect all pairs of trajectory points which are separated by a spacial distance below Δ_x

$$\{(x_{i,t}, x_{j,t'}) \mid \mathcal{D}(x_{i,t}, x_{j,t'}) < \Delta_x\} \ ,$$

Two spatially conflicting trajectory points might never become conflicting in time if the corresponding times are far apart. By introducing a constant maximum delay D_{max} we can dismiss all spatial conflicts which can never become conflicting in time

$$\{(x_{i,t}, x_{j,t'}) \mid \mathcal{D}(x_{i,t}, x_{j,t'}) < \Delta_x, |t-t'| \geq \Delta_t + D_{\max}\} \ .$$

With this, we are left with a set of potentially conflicting pairs of trajectory points

$$C_0^{ij} = \{(x_{i,t}, x_{j,t'}) \mid \mathcal{D}(x_{i,t}, x_{j,t'}) < \Delta_x, |t-t'| < \Delta_t + D_{\max}\} \ .$$

As a next step, we group together conflicting trajectory point pairs which are subsequent in time

$$\begin{aligned} C_{\parallel}^{ij} = \{((x_{i,t}, x_{j,t'}), (x_{i,s}, x_{j,s'})) \mid\ & (x_{i,t}, x_{j,t'}) \in C_0^{ij}, \\ & (x_{i,s}, x_{j,s'}) \in C_0^{ij}, \\ & |t-s| < \Delta_t', \\ & |t'-s'| < \Delta_t'\} \end{aligned}$$

where we set $\Delta_t' = 2$ minutes.

FIG. 1. Example of two parallel potential conflicts between two transatlantic flights starting from the east cost of the USA.

FIG. 2. Preprocessing: Reduction in the number of potential conflicts for various upper delay bounds $D_{\max}$.

For a given pair of flights (i,j) there might be multiple "disjoint" subsets in $C_{\parallel}^{ij}$

$$\bigcup_n C_{\parallel n}^{ij} = C_{\parallel}^{ij}$$

where

$$\begin{aligned} & |t-s| \geq \Delta_t' \wedge |t'-s'| \geq \Delta_t' \\ & \forall (x_{i,t}, x_{j,t'}) \in C_{\parallel n}^{ij}, \\ & \forall (x_{i,s}, x_{j,s'}) \in C_{\parallel n'}^{ij}, \\ & n \neq n' \ . \end{aligned}$$

In figure 1 an example of two separated clusters are shown. Together with the remaining, spatially isolated, conflicting trajectory points

$$C_{\times}^{ij} = C_0^{ij} \setminus C_{\parallel}^{ij} \ ,$$

these subsets of trajectory point clusters are called *potential conflicts*.

$$C_k \in C = \{C_{\parallel n}^{ij} | \forall i,j,n\} \cup \{C_{\times}^{ij} | \forall i,j\}$$

Here, we introduced a conflict index $k \in \{1, \ldots, N_c\}$, with $N_c = |C|$. For each conflict index k, we will denote the pair of involved flights by $I_k = (i_k, j_k)$.

Before the preprocessing, the number of conflicts was given by $N_c^{\text{before}} = \sum_{ij} |C_0^{ij}|$. As one can see in figure 2 the preprocessing reduces the number of conflicts by orders of magnitude.

B. Conflict Avoidance

In order to avoid conflicts, a flight i can be either delayed at departure time by d_i or by maneuver introduced delays d_{ik} for each conflict k the flight is involved in. With this, the trajectory times of flight i are shifted

$$\begin{aligned} t &\to t + D_i(t) \ , \\ D_i(t) &= d_i + \sum_{k \in K_i(t)} d_{ik} \ , \end{aligned} \quad (2)$$

where $D_i(t)$ is the delay of flight i at time t and the sum runs over all the conflicts which are before t

$$K_i(t) = \{k \mid \exists x_{i,t'} \in C_k \text{ for which } t' < t\} \ .$$

We introduce the pairs of times of spatially conflicting points inside a conflict k:

$$T_k = \{(t,t') \mid (x_{i,t}, x_{j,t'}) \in C_k, (i,j) \in I_k\} \ .$$

A conflicts k between two flights i and j occurs if the delays are chosen such that a temporal difference between spatially conflicting points is below Δ_t:

$$\begin{aligned} &|t + D_i(t) - t' - D_j(t')| < \Delta_t \\ \Rightarrow &-\Delta_t - (t-t') < D_i(t) - D_j(t') < \Delta_t - (t-t') \ , \end{aligned}$$

for any $(t,t') \in T_k$. Hence a conflict k is avoided if

$$D_i(t) - D_j(t') \notin D_k \qquad \forall\, (t,t') \in T_k \ .$$

with

$$D_k = \left(-\Delta_t - \max_{(t,t') \in T_k} (t-t'), \Delta_t - \min_{(t,t') \in T_k} (t-t') \right) \ .$$

C. Instances

To investigate the problem we consider each flight as a vertex of a graph and each conflict between two flights as an edge of this graph. The connected components of this *conflict graph* represent natural subsets of the problem. Figure (3) shows the number of connected components by varying the maximum delay time (Left), as well as the probability distribution of the number of flights in each connected component. Interestingly, the large part of connected components have small number of flights (for instance, 75% of the connected components for a maximum delay of 60 minutes have no more than 10 flights).

As part of our analysis, we also studied the probability distribution of the connectivity, namely the number of flights for which a given flight share a potential conflict with. It is well known that graphs with a non-trivial (small-world) structure with "hubs" have a connectivity distribution which follows a power-law distribution [2]. In the left panel of Figure (4), we show the connectivity distribution for a given maximum delay of 60 minutes. As one can see, the connectivity distribution shows a power-law decay of the form d^α, where d is the connectivity of a given flight, indicating that the underlying topology of the ATM problem is not trivial. In the right panel of Figure (4) we also show the power-law decay α as a function of the maximum delay time. As expected, α decreases by increasing the maximum delay time. Indeed, the number of potential conflicts shared by two flights increases as well by increasing the maximum delay time.

Even though the connectivity gives already an indication of the underlying structure of the topology of the conflict graph, it cannot be used to determine whether the graph has a tree-like structure or not. Indeed, if a connected component of the conflict graph were a tree, an optimal solution can be trivially found by iteratively propagate the delays along the tree. On the contrary, if flights in a connected components form a fully-connected graph (namely, all flights have pairwise potential conflicts), an optimal solution is hard to find. In order to understand if the connected components of the conflict graph look like tree, we studied the treewidth of the connected components [3, 4].

Intuitively, the treewidth is a property of a given graph and its value ranges from 1 (the graph is a tree) to the size of the graph (the graph is fully-connected). As shown in Figure (5), large parte of the connected components have a tree-like shape while few connected components (the hardest to find an optimal solution) look more like a fully-connected graph.

To be more precise, left panel of Figure (6) shows how the treewidth scales with the number of flights inside a given connected component. It is interesting to observe that the treewidth increases linearly with the number of flights belonging to the given connected component. This implies that larger connected components are also the hardest to optimize. Moreover, the slope γ of the linear correlation between the size of the connected components and their treewidth increases by increasing the maximum delay time (see right panel of Figure (**??**). This is consistent with the idea that preprocessing the flight dataset with a given maximum delay time makes indeed the ATM problem easier to tackle.

III. CONFIGURATION SPACE RESTRICTIONS

It is important to understand how the restrictions to the configuration space by (4) influence the solution quality. Therefore we solve (3) with a constraint programming solver [5] for various delay discretizations and upper bounds as well as for the continuous problem. As problem instances we used most of the connected components of the conflict graph for $D_{\max} = 18$ minutes with up to $N_f = 50$ flights and $N_c = 104$ conflicts.

In figure 7 one can see the results for a problem instance extracted from a connected component of the conflict graph with $N_f = 19$ flights and $N_c = 47$ conflicts. With the exception of the small maximum delay $d_{\max} = 3$ min, the total delay of the solutions is nearly independent of the maximum delay. Moreover it is monotonically increasing with the coarseness of the discretization. Since the original data is discretized in time in units of 1 minute, $\Delta_t = 1$ yield the same result as a continuous variable with the same upper bound. Obviously the total delay for the continuous solution decreases monotonically with $d_{\max}$. Above a certain value $d^0_{\max}$ the total delay stays the same. With one exception, we found that for all the investigated problem instances $d^0_{\max} \leq 6$ minutes (see figure 7). Therefore we conclude, that a moderate maximum delay is sufficient even for larger problem instances. On

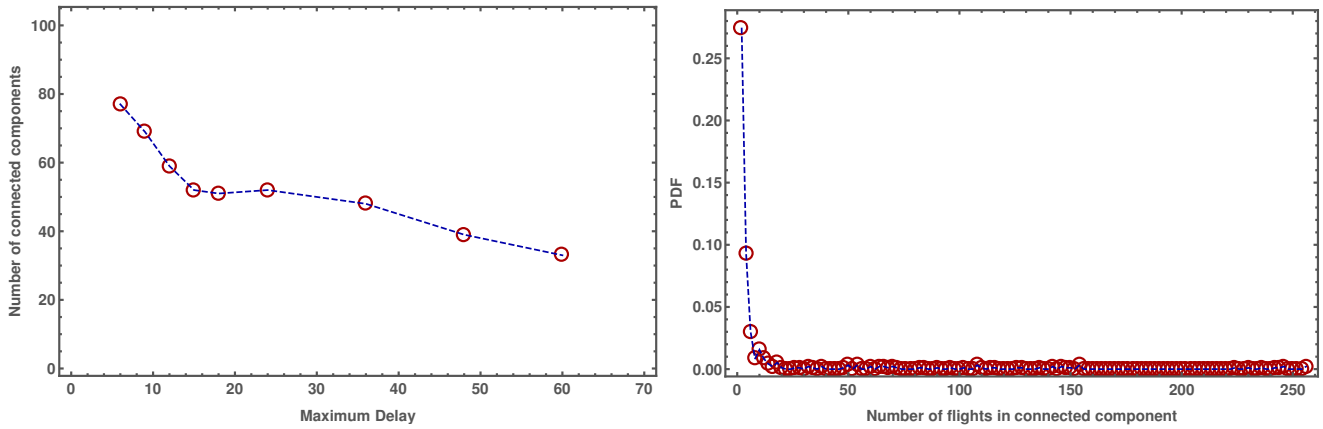

FIG. 3. (Left) Number of connected components by varying the maximum delay time. (Right) Histogram of the number of flights inside a connected component, regardless the maximum delay time.

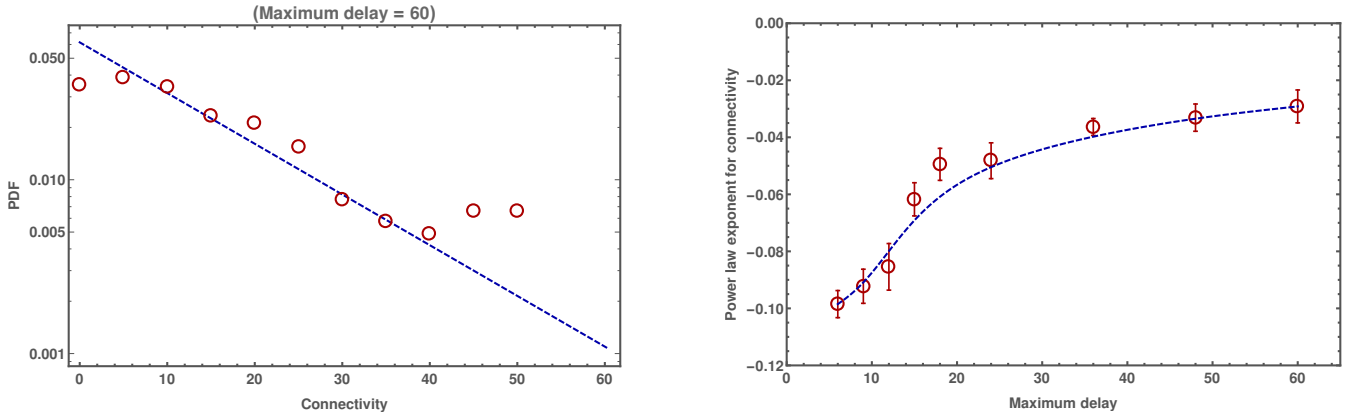

FIG. 4. (Left) Histogram of the connectivity, regardless the connected component, at fixed maximum delay time of 60 minutes. As one can see, the connectivity follows a power-law distribution. The coefficient of this power-law depends on the maximum delay time. (Right) coefficients of the power-law distribution by varying the maximum delay time.

the other hand, the delay discretization should be as fine as possible to obtain a high quality solutions.

IV. MAPPING TO QUBO

A. Departure delays

In this section we describe a simplified version of the above problem. We restrict ourselves to departure delays and neglect maneuvers. Hence the optimization problem can be written as

$$\begin{aligned} &\min_{d_i} \sum_i d_i \\ &\text{s.t.} \quad d_i - d_j \notin D_k \qquad \forall\,(t,t') \in T_k \end{aligned} \tag{3}$$

The problem needs to be mapped to a quadratic binary optimization problem (QUBO) in order to be solvable by a D-Wave quantum annealer. As a first step, we introduce a discretization and upper bound for the departure delay variables

$$d_i \in \{0, \Delta_d, 2\Delta_d, \ldots, d_{\max}\}\,,$$

where $d_{\max} = N_d \Delta_d$ and $(N_d + 1)$ is the number of delay steps. With this we can write the departure delay variables in terms of new binary variables $d_{i\alpha}$ by

$$\begin{aligned} d_i &= \sum_\alpha \alpha d_{i\alpha} \\ \alpha &\in \{0, \Delta_d, 2\Delta_d, \ldots, d_{\max}\}\,. \end{aligned} \tag{4}$$

However, this approach requires $\sum_\alpha d_{i\alpha} = 1$ to have a unique representation of d_i by $d_{i\alpha}$. This is achieved by adding the following contribution to the QUBO

$$Q_{\text{unique}} = \lambda_{\text{unique}} \sum_i \left(\sum_\alpha d_{i\alpha} - 1 \right)^2 . \tag{5}$$

where λ_{unique} is a penalty weight sufficiently large to ensure $Q_{\text{unique}} = 0$ for the solution. The cost function in

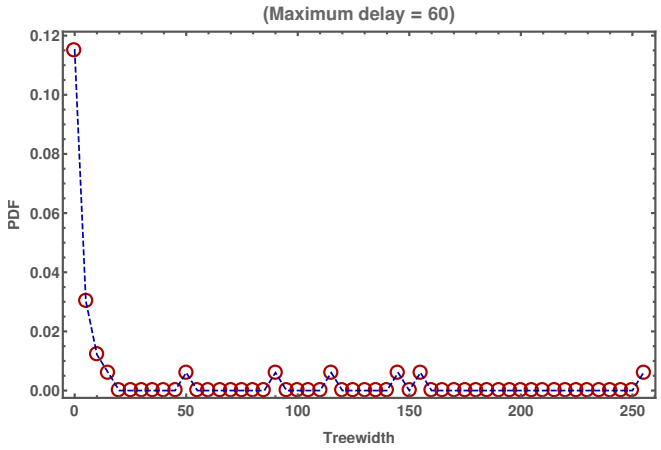

FIG. 5. Histogram of the treewidth of a given connected component, regardless the maximum delay time.

(3) yields contribution

$$Q_{\text{delay}} = \frac{1}{d_{\max}} \sum_{i\alpha} \alpha d_{i\alpha} \ , \tag{6}$$

where we have chosen the prefactor $1/d_{\max}$ for convenience. The last contribution to the QUBO represents the conflict avoidance

$$Q_{\text{conflict}} = \lambda_{\text{conflict}} \sum_{k} \sum_{(\alpha,\beta)\in A_k} d_{i\alpha} d_{j\alpha} \Big|_{i,j\in I_k} \tag{7}$$

where A_k is the set of all (α, β) which correspond to a conflict

$$A_k = \{(\alpha, \beta) \mid \alpha - \beta \in D_k\}$$

Again, $\lambda_{\text{conflict}}$ is the penalty weight for this contribution which have to be chosen large enough to ensure $Q_{\text{conflict}} = 0$ for the solution. The total QUBO of the departure delay model then reads

$$Q_{\text{DDM}} = Q_{\text{delay}} + Q_{\text{unique}} + Q_{\text{conflict}}$$

B. Maneuvers

A more realistic model of the problem can be created by including maneuvers. As mentioned above the maneuvers enter our formulation as additional delays d_{ik} at the conflict time. In the course of mapping to a QUBO formulation, we need to make sure to retain the combinatorial nature of the problem. We do this by restricting the vast realm of maneuvers to two distinct choices: Only one of the two involved flights is delayed while leaving the other flight untouched

$$\text{if } d_{ik} \neq 0 \Rightarrow d_{jk} = 0 \qquad \forall (i,j) \in I_k \ \forall k \ . \tag{8}$$

Moreover, we set the resulting maneuver delays to a constant value d_M large enough to capture all kinds of real maneuvers. A natural choice for this is the temporal conflict threshold $d_M = \Delta_t$.

With (2) we can introduce the delay a flight i at the conflict k as

$$D_{ik} = d_i + \sum_{k'<k} d_{ik} \ , \tag{9}$$

where we have defined a temporal ordering of the conflicts for each flight i by

$$\begin{aligned} k < p \text{ if } t < t' \\ \text{for } t = \arg\min_s x_{i,s} \in C_k \ , \\ t' = \arg\min_s x_{i,s} \in C_p \end{aligned}$$

The departure delay variables are represented by binary variables as it was done in section IV A. The maneuver delays are given by

$$d_{ik} = d_M a_{ik} \qquad a_{ik} \in \{0, 1\}$$

Since the total delay is given by $\sum_{ik} D_{ik}$, we can write the corresponding QUBO contribution as

$$\tilde{Q}_{\text{delay}} = \sum_{i\alpha} \alpha d_{i\alpha} + \sum_{ik} d_M a_{ik} \ ,$$

For the conflict avoidance, we need to introduce another variable representing the delay at a given conflict

$$D_{ik} = \sum_{\delta} \delta \Delta_{ik\delta} \qquad \Delta_{ik\delta} \in \{0, 1\} \ .$$

By restricting ourselves to $\Delta_d = \Delta_t$ the values of δ in the above equation are given as

$$\delta \in \{0, \Delta_t, 2\Delta_t, \ldots, (N_d + M_{ik})\Delta_t\} \ .$$

Here, M_{ik} is the number of conflicts the flight i is involved in before k. In order to fulfill (9) we add the following contribution to the QUBO

$$\tilde{Q}_{\Delta} = \lambda_{\Delta} \sum_{ik} \left(\sum_{\alpha} \alpha d_{i\alpha} + \sum_{k'<k} d_M a_{ik'} - \sum_{\delta} \delta \Delta_{ik\delta} \right)^2 \Big|_{i,j\in I_k}$$

For unique representation of the variables we add

$$\tilde{Q}_{\text{unique}} = \lambda_{\text{unique}} \left\{ \sum_{i} \left(\sum_{\alpha} d_{i\alpha} - 1 \right)^2 + \sum_{ik} \left(\sum_{\delta} \Delta_{ik\delta} - 1 \right)^2 \right\} \ .$$

Conflicts are avoided if $D_{ik} - D_{jk} \notin D_k$, $(i,j) \in I_k$. The corresponding QUBO contribution reads

$$\tilde{Q}_{\text{conflict}} = \lambda_{\text{conflict}} \sum_{k} \sum_{(\delta,\delta')\in B_k} \Delta_{ik\delta} \Delta_{jk\delta'} \Big|_{i,j\in I_k}$$

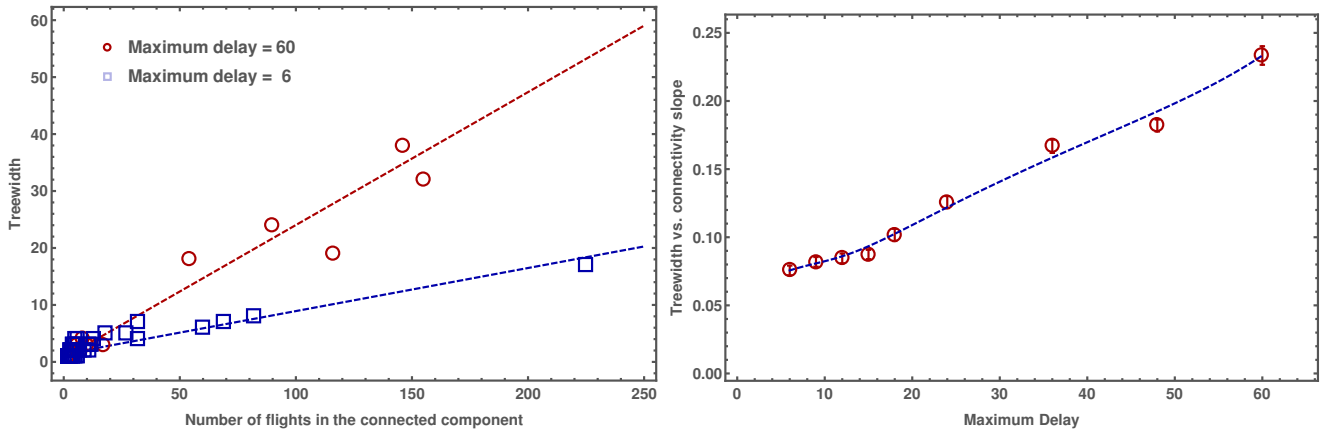

FIG. 6. (Left) Figure shows how the treewidth of a given connected component as a function of the number of flights. Interestingly, the treewidth is linear with the number of flights, with a slope γ which depends the on the maximum delay time. (Right) Slope γ as a function of the maximum delay time.

FIG. 7. Top: Total delay of constraint programming solutions for a problem instance with $N_f = 19$ flights and $N_c = 47$ conflicts for various discretization parameters. Bottom: Minimum $d_{\max}$ which yield optimal solution in continuous problem for various problem instances. For all problem instances we used $D_{\max} = 18$ minutes.

where B_k is the set of all (δ, δ') which correspond to a conflict

$$B_k = \{(\delta, \delta') \mid \delta - \delta' \in D_k\}$$

The penalty weights λ_Δ, λ_{unique} and $\lambda_{\text{conflict}}$ must be chosen large enough to ensure vanishing contributions from the corresponding QUBO terms for the solution.

Finally, the maneuver decision described by (8) is incorporated by a antiferromagnetic coupling between the two maneuver delay variables

$$\tilde{Q}_{\text{maneuver}} = J \sum_k (s_{ik} s_{jk} + 1)_{i,j \in I_k} \ .$$

with

$$s_{ik} = 2a_{ik} - 1 \in \{-1, 1\}$$

and $J > 0$ has to be chosen large enough. A solution is considered to be valid only if $\tilde{Q}_{\text{maneuver}} = 0$. Hence, the total QUBO for the maneuver model reads

$$Q_{\text{MM}} = \tilde{Q}_{\text{delay}} + \tilde{Q}_\Delta + \tilde{Q}_{\text{unique}} + \tilde{Q}_{\text{conflict}} + \tilde{Q}_{\text{maneuver}}$$

C. Choice of the Penalty Weights

The contributions (5) and (7) to the QUBO for the departure delay model of section IV A are hard constraints. This means a solution to the QUBO is only valid if both (5) and (7) vanish. Therefore, the penalty weights λ_{unique} and $\lambda_{\text{conflict}}$ must be chosen sufficiently large to ensure that the hard constraints are fulfilled for the solution to the problem. On the other hand, large penalty weights lead to large differences between the largest and smallest non-vanishing coefficients in the QUBO. Since the D-Wave quantum annealers have a limited resolution for the specification of the QUBO, this can lead to a misspecification of the problem. [6] Hence, it is desirable to find a sweet spot of the smallest penalty weights which still yield valid solutions.

FIG. 8. Validity of exact solution to a QUBO extracted from a problem instance with $N_f = 7$ flights and $N_c = 9$ conflicts in dependence on the choice of the penalty weights, λ_{unique} and $\lambda_{\text{conflict}}$.

In order to find these optimal penalty weights, we employed an exact solver [7] to explore the validity of a solution in dependence on the penalty weights. We investigated problem instances with up to $N_f = 7$ flights and $N_c = 9$ conflicts. For all these problem instances we found a box like shape of the boundary between valid and invalid solutions as it is depicted in figure 8.

One can give an upper bound for the sufficiently large penalty weights by the following considerations. A minimal violation of the hard constraints yield an additional contribution to the QUBO cost function of λ_{unique} or $\lambda_{\text{conflict}}$, respectively. As a result of this violation, the contribution from (6) can be reduced maximally by

$$\min_{\{d_{i\alpha}\}} \left\{ \frac{1}{d_{\max}} \sum_{i\alpha} \alpha d_{i\alpha} - \frac{1}{d_{\max}} \sum_{j\beta} \beta d_{j\beta} \right\} = -1$$

Therefore an upper bound for sufficiently large penalty weights is given by

$$\lambda_{\text{unique}} > 1$$
$$\lambda_{\text{conflict}} > 1$$

D. Optimization of the QUBO formulation using classical heuristics

In this Section we present the result for the optimization of the QUBO formulation of the ATM problem using the Isoenergetic Cluster Method (a rejectionfree cluster algorithm for spin glasses that greatly improves thermalization) [8], which has been shown to be one of the fastest classical heuristic to optimize QUBO problems [9].

Figure (9) shows the total delay time optimized by ICM either by varying the partition at fixed the delay step Δt (left panel) or by varying the delay step Δt at fixed partition (right panel). As one can see, the total delay decreases by decreasing Δt and it eventually reaches an optimal plateau. Results are for maximum delay of 60 minutes. This is consistent with the idea that smaller Δt allows a finer optimization of the delays of the flights.

In Figure (10) we show the optimal delay time found by ICM as a function of the number of the flights in the connected components. Results are for a maximum delay of 60 minutes. Unfortunately, ICM was unable to optimize connected components with more than 12 flights. This can be explained by recalling that ICM works the best for almost-planar problem while the its performance quickly decreases for fully-connected problems. Indeed, as shown in Section II C, the underlying graph of connected components look more like a fully-connected graph rather than a tree graph by increasing the number of flights inside the connected component.

V. D-WAVE

A. Embedding

B. Quantum Annealing Results

VI. CONCLUSIONS

TODO

VII. ACKNOWLEDGEMENTS

[1] O. Rodionova, D. Delahaye, B. Sridhar, and H. Ng., *Deconflicting wind-optimal aircraft trajectories in north atlantic oceanic airspace*, Proceedings of Advanced Aircraft Efficiency in a Global Air Transport System (AEGATS'16) Conference (2016).

[2] R. Albert, H. Jeong, and A.-L. Barabási, *Internet: Diameter of the world-wide web*, nature **401**, 130 (1999).

[3] U. Bertele and F. Brioschi, *Nonserial dynamic programming* (Academic Press, 1972).

[4] R. Halin, *S-functions for graphs*, Journal of geometry **8**, 171 (1976).

[5] E. Hebrard, E. O'Mahony, and B. O'Sullivan, *Constraint Programming and Combinatorial Optimisation in Numberjack* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010), pp. 181–185, ISBN 978-3-642-13520-0, URL http://dx.doi.org/10.1007/978-3-642-13520-0_22.

[6] Note1, reference to D-Wave limited resolution.

[7] Note2, reference to mapping from QUBO to max sat and to akmaxsat solver.

[8] Z. Zhu, A. J. Ochoa, and H. G. Katzgraber, *Efficient cluster algorithm for spin glasses in any space dimension*, Physical review letters **115**, 077201 (2015).

[9] S. Mandrà, Z. Zhu, W. Wang, A. Perdomo-Ortiz, and H. G. Katzgraber, *Strengths and weaknesses of weak-strong cluster problems: A detailed overview of state-of-the-art classical heuristics versus quantum approaches*, Physical Review A **94**, 022337 (2016).

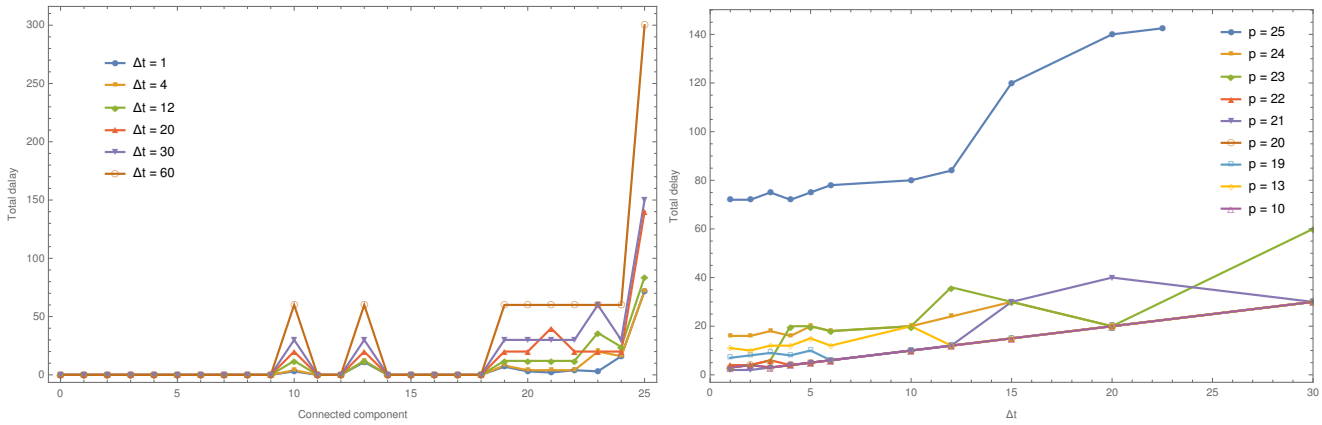

FIG. 9. (Left) Optimal total delay found by using the Isoenergetic Cluster Method (ICM) at fixed time step Δt, by varying the connected component. Results are for maximum delay time of 60 minutes. (Right) Optimal delay found by using ICM at fixed connected component, by varying the time step Δt.

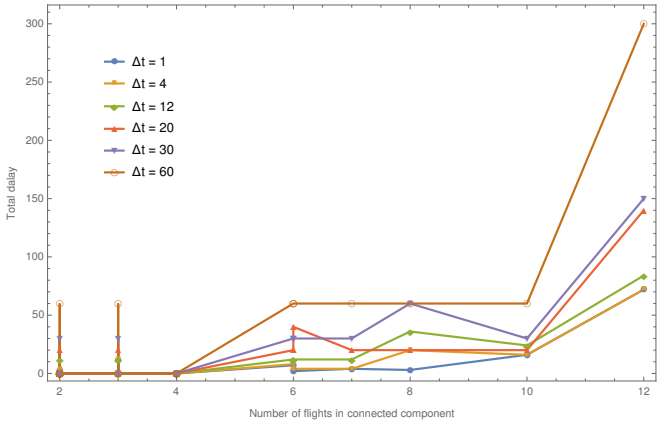

FIG. 10. . Optimal total delay found by using the Isoenergetic Cluster Method (ICM) at fixed time step Δt as a function of numbers of flight within each connected component. ICM was unable to find solutions for connected component with more than 12 flights.

FIG. 11. Number of physical qubits versus the number of logical qubits after embedding of QUBO instances with up to $N_f = 50$ and $N_c = 104$

FIG. 12. Success probability for QUBO instances in dependence of the number of flights N_f and the number of conflicts N_c. The error bars indicate the standard deviation. We used 10000 annealing runs for each instance and penalty weights $\lambda = \lambda_{\text{conflict}} = \lambda_{\text{unique}} \in \{0.5, 1, 2\}$.