CONTENTS

LIST OF FIGURES

Quantum Annealing for Air Traffic Management

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

Bryan O'Gorman
University of California, Berkeley
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 16, 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.

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 "deconflicting", 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

The basic input of the deconflicting problem is a set of ideal flight trajectories (space-time paths). These ideal trajectories are specified by the individual flight operators. Each ideal trajectory represents some independent optimization from the operator's perspective, especially minimizing fuel costs given expected wind conditions between the desired origin and destination at the desired times; for this reason, they are called the "wind-optimal" trajectories. Because of the number of such trajectories and the correlation between them, these trajectories are likely to conflict; that is, two or more aircraft are likely to get dangerously close to each other if their ideal trajectories are followed without modification. The goal thus is to modify the trajectories to avoid such conflicts.

In theory, the configuration space consists of all physically realistic trajectories; in practice, computational bounds constrain us to consider perturbations of the ideal trajectories. The simplest such perturbation is a departure delay, which is the main focus of the present work. Previous work [1] additionally considered a global perturbation by which a trajectory is sinusoidally shifted parallel to the Earth's surface. We focus instead on local perturbations to the trajectories, in which a modification to the trajectory is parameterized by some choice of active maneuvers near a potential conflict; such a modification does not affect the preceding part of the trajectory and only affects the subsequent part by the additional delay it introduces.

A full accounting of the cost of such modifications would take into account the cost of departure delays, the change in fuel cost due to perturbing the trajectories, the relative importance of each flight, and many other factors. As in previous work, we consider only the total, unweighted arrival delay, aggregated equally over all of the flights.

Formally, each ideal trajectory $\mathbf{x}_i = (x_{i,t})_{t=t_{i,0}}^{t_{i,1}}$ is specified as a time-discretized path from the departure point $x_{i,t_{i,0}}$ at time $t_{i,0}$ to the arrival point $x_{i,t_{i,1}}$ at time $t_{i,1}$.

For each flight i, the geographical coordinates $x_{i,t}$ (as latitude, longitude, and altitude) are specified at every unit of time (i.e. one minute) between $t_{i,0}$ and $t_{i,1}$; we call this interval $T_i = (t_{i,0}, t_{i,0}+1, \ldots, t_{i,1})$.

For notational simplicity, suppose momentarily that each trajectory $\mathbf{x}_i$ is modified only by introducing delays between time steps. Let $\delta_{i,t}$ be the accumulated delay of flight i at the time that it reaches the point $x_{i,t}$, and let $\delta^*_{i,t}$ be the maximum such delay.

A pair of flights (i,j) are in conflict with each other if any pair of points from their respective trajectories is in conflict. (The trajectories are reasonably assumed to be sufficiently time-resolved so that if the continuously interpolated trajectories conflict than there is a pair of discrete trajectory points that conflict.) A pair of trajectory points $(x_{i,s}, x_{j,t})$ conflict if their spatial and temporal separations are both within the respective mandatory separation standards Δ_x and Δ_t (i.e. 3 nautical miles and 3 minutes):

$$\|x_{i,s} - x_{j,t}\| < \Delta_x \text{ and } |(s+\delta_{i,s}) - (t+\delta_{j,t})| \tag{1}$$

. The latter condition can be met for some $(\delta_{i,s}, \delta_{j,t}) \in [0, \delta^*_{i,s}] \times [0, \delta^*_{j,t}]$ if and only if

$$\max\left\{\delta^*_{i,s}, \delta^*_{j,t}\right\} + \Delta_t > |s-t|, \tag{2}$$

in which case we call the pair of trajectory points *potentially* conflicting. The set C of such pairs of potentially conflicting trajectory points contains strongly correlated clusters. To simplify the constraints, we enumerate all such clusters and refer to them simply as *the* conflicts. That is, we partition the potentially conflicting pairs of trajectory points into disjoint sets,

$$C = \bigcup_k C_k, \tag{3}$$

such that if $\{(i,s),(j,t)\}, \{(i',s'),(j',t')\} \in C_k$ for some k then $i = i' < j = j'$ and for all $s'' \in [\min\{s,s'\}, \max\{s,s'\}]$ there exists some $t'' \in [\min\{t,t'\}, \max\{t,t'\}]$ such that $\{(i,s''),(j,t'')\} \in C_k$. Thus every conflict k is associated with a pair of flights $I_k = \{i,j\}$. Let $K_i = \{k | i \in I_k\}$ be the set of conflicts to which flight i is associated.

Having identified disjoint sets of conflicts, we relax the supposition that the trajectory modifications only introduce delays between time steps. Instead, we consider modifications to the trajectories that introduce delays local to particular conflicts. Specifically, the configuration space consists of the departure delays $\mathbf{d} = (d_i)_{i=1}^n$ and the set of local maneuvers $\mathbf{a_k} = (\mathbf{a}_k)_k$, where $\mathbf{a}_k$ represents some parameterization of the local maneuvers used to avoid conflict k. Let $d_{i,k}(\mathbf{d}, \mathbf{a_k})$ be the delay introduced to flight i at conflict k, as a function of the departure delays and local maneuvers. With this notation, we can write the total delay as

$$D = \sum_{i=1}^n \left(d_i + \sum_{k \in K_i} d_{i,k}\right). \tag{4}$$

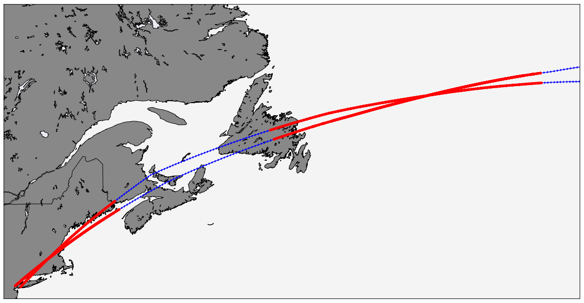

FIG. 1. Example of two parallel potential conflicts between two transatlantic flights starting from the east cost of the USA.

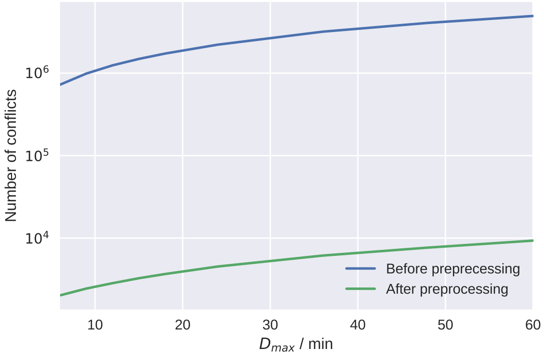

FIG. 2. Preprocessing: Reduction in the number of potential conflicts for various upper delay bounds $D_{\max}$.

This is the quantity we wish to minimize subject to avoiding all potential conflicts.

A conflict can be avoided locally by introducing earlier delays differentially, thereby increasing the temporal separation; by some active maneuver of one or both of the flights; or by some combination thereof. We focus on the former case. Let

$$D_{i,k} = d_i + \sum_{k' \in K_i | k' < k} d_{i,k'} \tag{5}$$

be the accumulated delay of flight i by the time it reaches conflict k. We assume that the set of conflicts K_i associated with flight i is indexed in temporal order, i.e. if $k' < k$ and $k, k' \in K_i$, then flight i reaches conflict k' before conflict k. The pairs of conflicting trajectory points associated with conflict k are given by

$$T_k = \{(s,t) | \{(i,s),(j,t)\} \in C_k, i < j\}. \tag{6}$$

Thus the potential conflict is avoided only if

$$D_{i,k} - D_{j,k} \notin D_k \tag{7}$$

where

$$D_k = \bigcup_{(s,t)\in T_k} (-\Delta_t + t - s, \Delta_t + t - s) = [\Delta_k^{\min}, \Delta_k^{\max}], \tag{8}$$

$$\Delta_k^{\min} = 1 - \Delta_t + \min_{(s,t)\in T_k} \{t - s\}, \tag{9}$$

$$\Delta_k^{\max} = \Delta_t - 1 + \max_{(s,t)\in T_k} \{t - s\}. \tag{10}$$

In the remainder of this paper, we focus on the restricted problem in which only departure delays are allowed. In this simplified case, the configuration space is simply $\mathbf{d} = (d_i)_{i=1}^n$, the cost function simply $D = \sum_{i=1}^n d_i$, and the constraints simply $d_i - d_j \not inD_k$ for all k.

A. Instances

To assess our methods on realistic instances of the problem, we use the actual wind-optimal trajectories for transatlantic flights on July 29, 2012, as was done in previous work [1]. In these trajectories, each flight i has a constant (cruising) altitude and constant speed, to within (classical) machine precision, though our methods generalize to instances without these special properties.

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. 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 part 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 (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. DISCRETIZING THE CONFIGURATION SPACE

It is important to understand how the restrictions to the configuration space by (**??**) influence the solution quality. Therefore we solve (**??**) 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

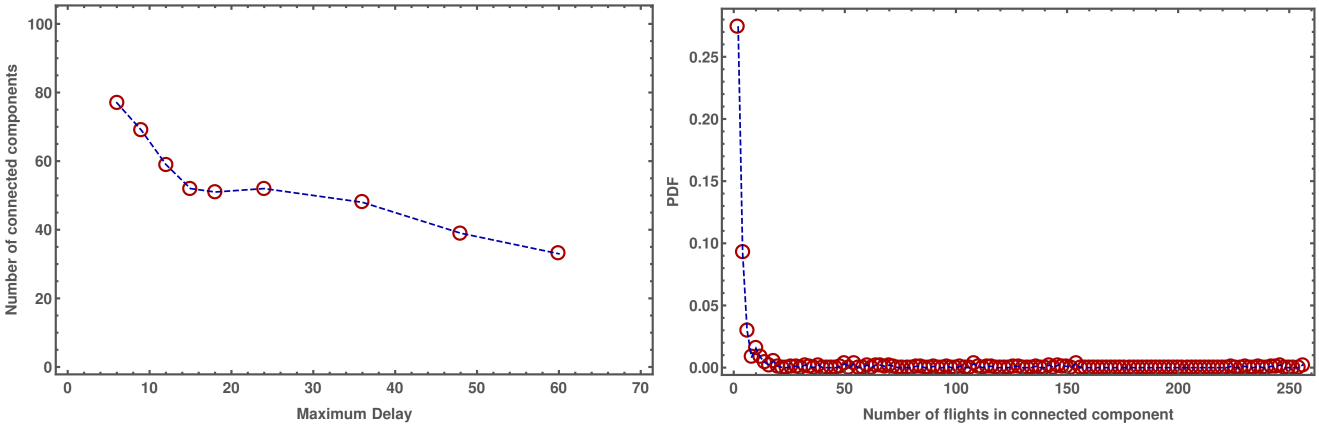

FIG. 3. (Left) Number of connected components by varying the maximum delay time. (Right) Histogram of the number of flights inside a connected component, regardless the maximum delay time.

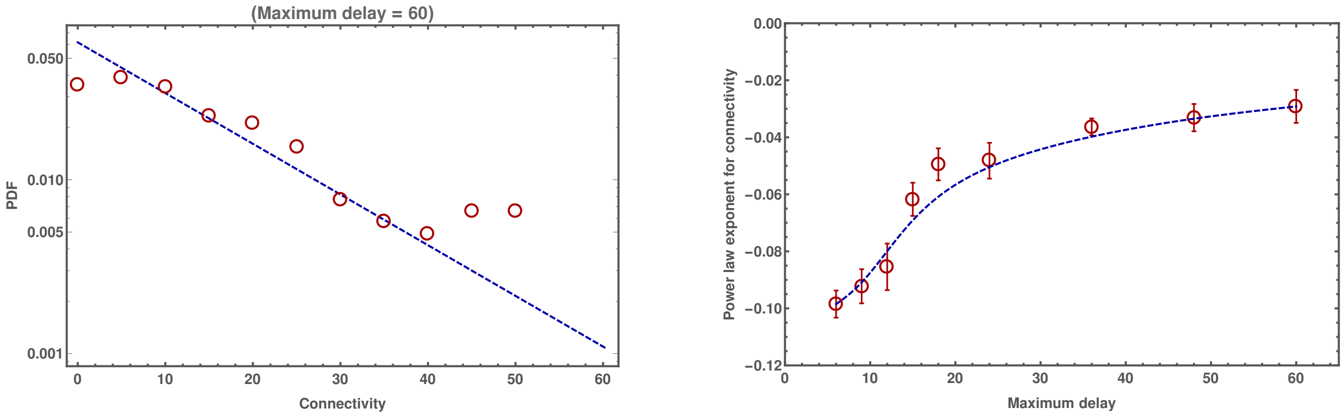

FIG. 4. (Left) Histogram of the connectivity, regardless the connected component, at fixed maximum delay time of 60 minutes. As one can see, the connectivity follows a power-law distribution. The coefficient of this power-law depends on the maximum delay time. (Right) coefficients of the power-law distribution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

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

IV. MAPPING TO QUBO

A. Binary encoding

1. *Departure delays*

To apply quantum annealing to the deconflicting problem, we must encode the configuration space $\mathbf{d}$ in binary-valued variables. To do so, we must first discretize and bound the allowed values. Let Δ_d be the resolution of the allowed delays and $d_{\max} = N_d \Delta_d$ the maximum allowed delay, so that $d_i \in \{\Delta_d l | l \in [0, 1, \dots, N_d]\}$. The value of d_i is encoded in $N_d + 1$ variables $d_{i,0}, \dots, d_{i,N_d+1} \in \{0, 1\}$ using a one-hot encoding:

$$d_{i,\alpha} = \begin{cases} 1, & d_i = \alpha, \\ 0, & d_i \neq \alpha; \end{cases} \qquad d_i = \Delta_d \sum_{l=0}^{N_d} d_{i,l}. \tag{11}$$

To enforce this encoding, we add the penalty function

$$f_{\text{encoding}} = \lambda_{\text{encoding}} \sum_{i=1}^{n} \left(\sum_{l=0}^{N_d} d_{i,l} - 1 \right)^2, \tag{12}$$

where $\lambda_{\text{encoding}}$ is a penalty weight sufficiently large to ensure that any cost minimizing state satisfies $f_{\text{encoding}} = 0$. In terms of these binary variables, the cost function is

$$f_{\text{delay}} = \Delta_d \sum_{i=1}^{n} \sum_{l=0}^{N_d} d_{i,l}, \tag{13}$$

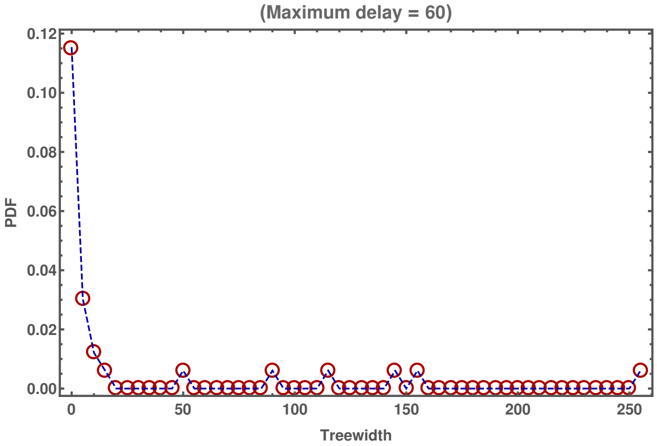

FIG. 5. Histogram of the treewidth of a given connected component, regardless the maximum delay time.

Lastly, actualized conflicts are penalized by

$$f_{\text{conflict}} = \lambda_{\text{conflict}} \sum_{k} \sum_{\substack{l,l' \big| \Delta_d(l-l') \in D_k \\ i,j \in I_k | i<j}} d_{i,l} d_{j,l}, \tag{14}$$

where again $\lambda_{\text{conflict}}$ is a sufficiently large penalty weight. The overall cost function to be minimized is

$$f = f_{\text{encoding}} + f_{\text{delay}} + f_{\text{conflict}}. \tag{15}$$

B. Softening the constraints

The contributions from (**??**) and (**??**) to the QUBO for the departure delay model of section IV A 1 are hard constraints. This means a solution to the QUBO is only valid if both (**??**) and (**??**) 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.

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. Such a violation would correspond to single bit flip in the binary delay variables $d_{i\alpha}$. Therefore the contribution from (**??**) would be reduced maximally by

$$\min_{\alpha} \frac{-\alpha}{d_{\max}} = -1$$

Hence, sufficiently large penalty weights must fulfill the following conditions

$$\begin{aligned} \lambda_{\text{unique}} &> 1 \\ \lambda_{\text{conflict}} &> 1\,. \end{aligned}$$

This corresponds to a box like shape as it appears in figure 8. ¡¡¡¡¡¡¡ HEAD

V. RESULTS FROM ICM

=======

A. 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 rejection-free 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 A, 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.

¿¿¿¿¿¿¿ 1371b9892d7b6df72d6f2c8fd78c10ac9acd0c86

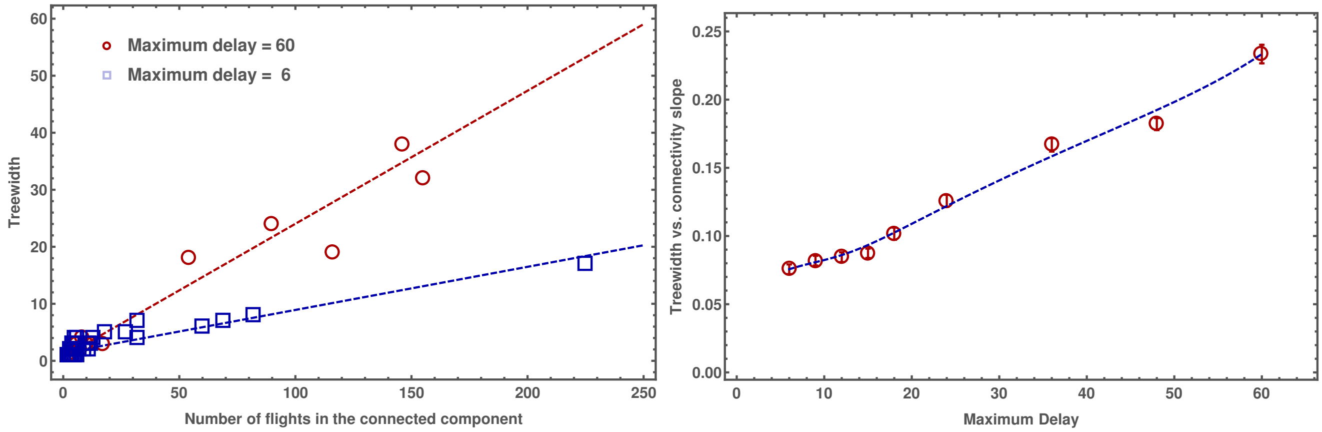

FIG. 6. (Left) Figure shows how the treewidth of a given connected component as a function of the number of flights. Interestingly, the treewidth is linear with the number of flights, with a slope γ which depends the on the maximum delay time. (Right) Slope γ as a function of the maximum delay time.

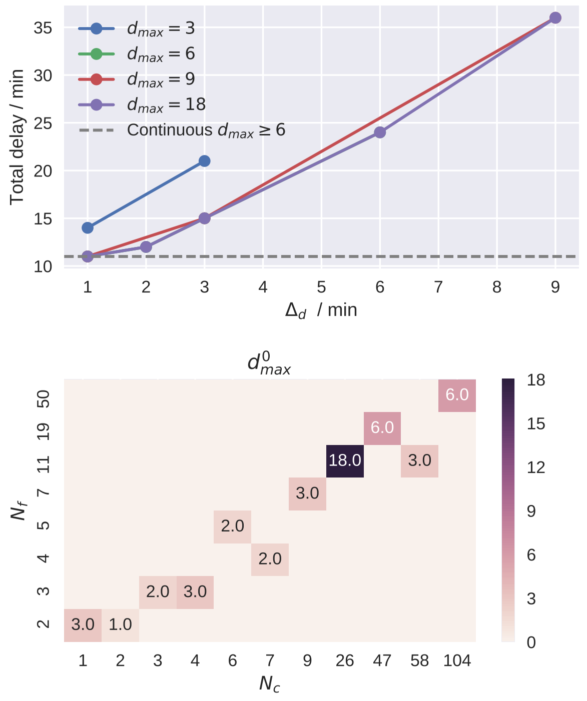

FIG. 7. Top: Total delay of constraint programming solutions for a problem instance with $N_f = 19$ flights and $N_c = 47$ conflicts for various discretization parameters. Bottom: Minimum $d_{\max}$ which yield optimal solution in continuous problem for various problem instances. For all problem instances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

VI. QUANTUM ANNEALING

In this section we report on our efforts to solve problem instances from the departure delay model from section IV A 1 with a D-Wave 2X quantum annealer. We restricted ourselves to instances with $d_{\max} = D_{\max} = 18$ and $\Delta_d \in \{3, 6, 9\}$.

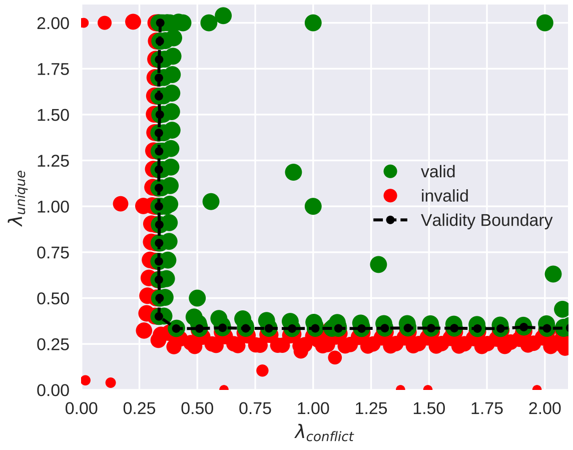

FIG. 8. Validity of exact solution to a QUBO extracted from a problem instance with $N_f = 7$ flights and $N_c = 9$ conflicts in dependence on the choice of the penalty weights, λ_{unique} and $\lambda_{\text{conflict}}$. Here, $\Delta_t = 3$ and $d_{\max} = 18$.

A. Embedding

In order to make a QUBO amenable for a D-Wave 2X quantum annealer, it has to obey certain hardware constraints. For instance the connections between the binary variables are restricted to the so called Chimera graph [10]. However, it is possible to map every QUBO to another QUBO which obeys the Chimera architecture while increasing the number of binary variables used by a so called minor-embedding technique [11].

We found, that non of the instances were suitable for direct calculation on the D-Wave machine. Therefore we used D-Wave's heuristic embedding algorithm [12]

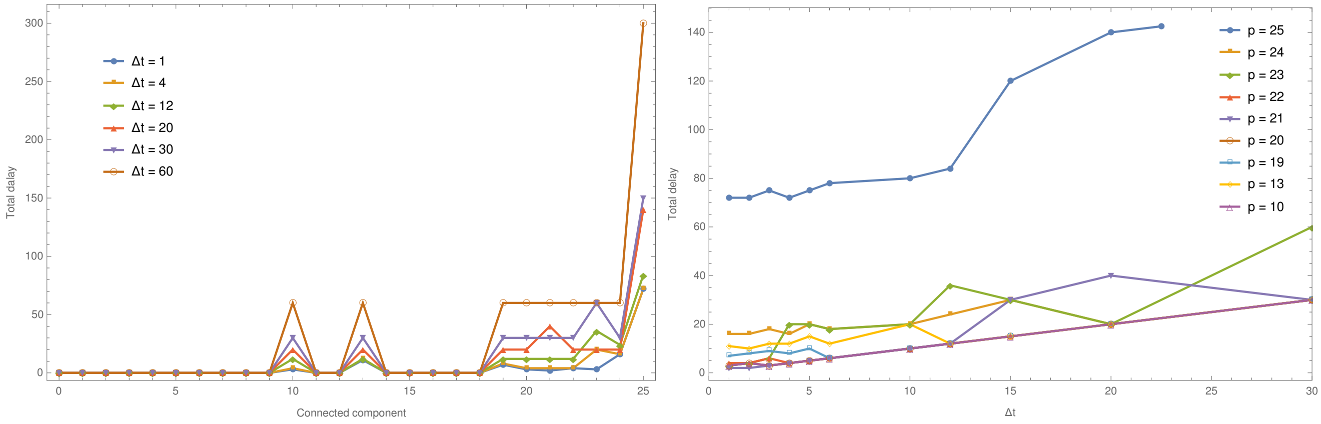

FIG. 9. (Left) Optimal total delay found by using the Isoenergetic Cluster Method (ICM) at fixed time step Δt, by varying the connected component. Results are for maximum delay time of 60 minutes. (Right) Optimal delay found by using ICM at fixed connected component, by varying the time step Δt.

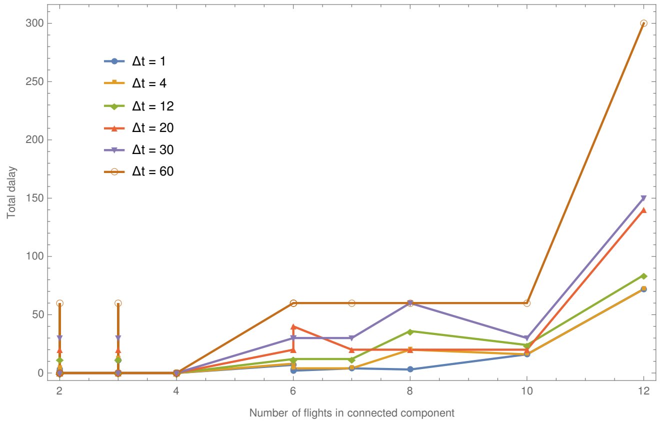

FIG. 10. . Optimal total delay found by using the Isoenergetic Cluster Method (ICM) at fixed time step Δt as a function of numbers of flight within each connected component. ICM was unable to find solutions for connected component with more than 12 flights.

Δ_d	3	6	9
Number of flights N_f	13	19	50
Number of conflicts N_c	27	47	104
Number of logical qubits	91	76	150
Average number of physical qubits	631	395	543

TABLE I. Parameters of the largest embeddable instances for the D-Wave 2X

to embed instances with up to $N_f = 50$ and $N_c = 104$ depending on discretization (cf. table I). In figure 11 on can see the dependence of the number of physical qubits on the number of logical qubits.

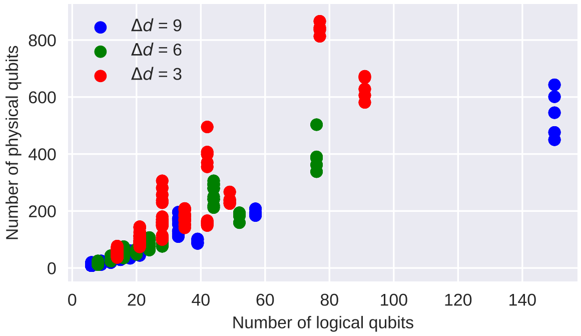

FIG. 11. Number of physical qubits versus the number of logical qubits after embedding of QUBO instances for the departure delay model.

B. Success Probability

VII. CONCLUSIONS

TODO

VIII. ACKNOWLEDGEMENTS

a. 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

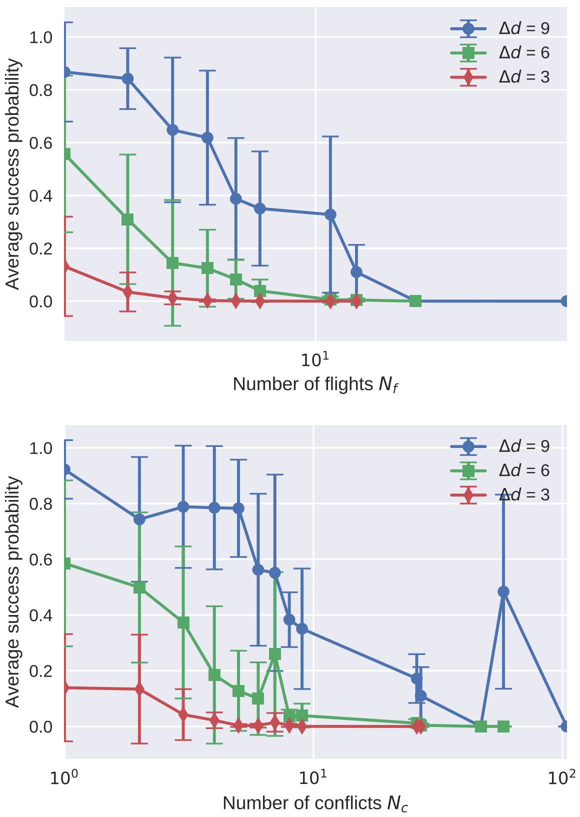

FIG. 12. Success probability for QUBO instances in dependence of the number of flights N_f and the number of conflicts N_c. The error bars indicate the standard deviation. We used 10000 annealing runs for each instance and penalty weights $\lambda = \lambda_{\text{conflict}} = \lambda_{\text{unique}}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

flight untouched

$$\text{if } d_{ik} \neq 0 \Rightarrow d_{jk} = 0 \qquad \forall (i,j) \in I_k \ \forall k \, . \tag{16}$$

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 (**??**) we can introduce the delay a flight i at the conflict k as

$$D_{ik} = d_i + \sum_{k'<k} d_{ik} \, , \tag{17}$$

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

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

The departure delay variables are represented by binary variables as it was done in Section IV A 1. 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 (17) we add the following contribution to the QUBO

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

For unique representation of the variables we add

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

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

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

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 (16) 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

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

[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).

[10] Note3, reference to Chimera graph.

[11] Note4, reference to minor-embedding.

[12] Note5, reference to D-Wave embedding algorithm.