

Quantum Annealing Applied to De-Conflicting Optimal Trajectories for Air Traffic Management

Tobias Stollenwerk, Bryan O’Gorman, Davide Venturelli, Salvatore Mandrà^{ID}, Olga Rodionova, Hokkwan Ng, Banavar Sridhar, Eleanor Gilbert Rieffel, and Rupak Biswas

Abstract—We present the mapping of a class of simplified air traffic management problems (strategic conflict resolution) to quadratic unconstrained Boolean optimization problems. The mapping is performed through an original representation of the conflict-resolution problem in terms of a conflict graph, where the nodes of the graph represent flights and the edges represent a potential conflict between flights. The representation allows a natural decomposition of a real-world instance related to wind-optimal trajectories over the Atlantic Ocean into smaller subproblems that can be discretized and are amenable to be programmed in quantum annealers. In this paper, we tested the new programming techniques, and we benchmark the hardness of the instances using both classical solvers and the D-Wave 2X and D-Wave 2000Q quantum chip. The preliminary results show that for reasonable modeling choices, the most challenging subproblems which are programmable in the current devices are solved to optimality with 99% of probability within a second of annealing time.

Index Terms—Air traffic management, optimal trajectories, classical optimization, quantum optimization.

I. INTRODUCTION

ONE of the main functions of Air Traffic Control (ATC) is ensuring safe flight progress in the shared airspace. This in particular involves resolving potential conflicts between flights, where “a conflict” stands for a violation of separation norms established in the airspace.

Manuscript received March 14, 2018; revised July 19, 2018; accepted December 14, 2018. This work was supported in part by the NASA Aviation Systems Division, in part by the NASA Advanced Exploration Systems Program, and in part by the NASA Ames Research Center. The work of D. Venturelli was supported by the NASA Academic Mission Services under Contract NNA16BD14C. The Associate Editor for this paper was D.-H. Lee. (*Corresponding author: Salvatore Mandrà*)

T. Stollenwerk is with the German Aerospace Center (DLR), 51147 Cologne, Germany.

B. O’Gorman is with the Department of Electrical Engineering and Computer Sciences and Department of Chemistry, University of California, Berkeley, CA 94720 USA, also with the Quantum Artificial Intelligence Laboratory, NASA Ames Research Center, Moffett Field, CA 94035 USA, and also with Stinger Ghaffarian Technologies Inc., Greenbelt, MD 20770 USA.

D. Venturelli is with the Quantum Artificial Intelligence Laboratory, NASA Ames Research Center, Moffett Field, CA 94035 USA, and also with the Research Institute for Advanced Computer Science, Universities Space Research Association, Mountain View, CA 94035 USA.

S. Mandrà is with the Quantum Artificial Intelligence Laboratory, NASA Ames Research Center, Moffett Field, CA 94035 USA, and also with Stinger Ghaffarian Technologies Inc., Greenbelt, MD 20770 USA (e-mail: salvatore.mandra@nasa.gov).

O. Rodionova, H. Ng, and B. Sridhar are with the NASA Ames Research Center, Moffett Field, CA 94035 USA.

E. G. Rieffel and R. Biswas are with the Quantum Artificial Intelligence Laboratory, NASA Ames Research Center, Moffett Field, CA 94035 USA. Digital Object Identifier 10.1109/TITS.2019.2891235

There is an overall increase in air traffic over the last decades and this trend is believed to continue. As a result, ATC workload is constantly increasing. Nowadays the flights are typically assigned the predefined routes from the air traffic network, which is becoming saturated. With the limited airspace available, novel approaches are necessary to meet the increasing air traffic demand in the coming decades. One promising approach that addresses both traffic congestion and fuel costs is to start with wind-optimal trajectories, i.e. the route that each flight would take to minimize fuel costs if there were no other flights [1]. Such wind-optimal trajectories will conflict with each other [2] and thus be deconflicted. Conflict detection and resolution is a complex problem which has been studied for the decades [3], [4].

Quantum annealing is a promising computational method which became increasingly important in recent years. This development is driven also by first commercially available quantum annealing device by the company D-Wave Systems. In addition to studying the fundamental properties of quantum annealing, it is imperative to find possible real world application for this technology. Hard operational planning problems are a promising candidate for the latter [5]–[7].

In this work, we investigate the feasibility of applying quantum annealing to solve the conflict resolution problem for wind-optimal trajectories. To be amenable to a D-Wave quantum annealer, the conflict-resolution problem has to be formulated as a quadratic unconstrained binary optimization (QUBO) problem. Given the discrete nature of QUBO problems, a tunable discretization must be introduced. Nevertheless, while treating trajectories as continuous functions can be beneficial [2], it is also more computational demanding. On the contrary, a QUBO formulation of the de-conflicting problem allows to “tune” the discretization to trade between quality of solutions and computational effort. For the main part of the paper, we restrict ourselves to a simplified version of the problem by considering departure delays only while neglecting maneuvers. We present a detailed study of the structure of this problem which provides insights beyond the scope of quantum annealing.

In particular, we perform the following analyses

- Given the wind-optimal trajectories, we extract natural subsets of the overall problem and study their hardness. We found that the problems are hard in general and become harder as we increase the maximum allowed value for the departure delays.

- Restrictions to the configurations space are necessary for the reformulation of the problem as a QUBO. Therefore, we employ classical solvers to investigating the influence of discretization on the solution quality. As a result, we found that finer discretization increases the solution quality and sufficiently large maximum allowed value for the departure delays is enough for an acceptable solution
- We demonstrate the mapping of the deconflicting problem to a QUBO formulation for models with (see Appendix) and without maneuvers (in main text). In the course of this, we investigate the sufficient penalty weights for the hard constraints in the problem. Here we found that these penalty weights are largely independent of the problem instances.
- We investigate the possibility to embed deconfliction-derived QUBO instances onto the D-Wave quantum annealer. More precisely, we were able to embed and run smaller problem instances and found that finer model discretizations as well as larger problem sizes decrease the success probability due to the limited precision of the D-Wave 2X machine.

The paper is organized as follows: We begin by formulating the conflict-resolution problem as a combinatorial optimization problem and describing the preprocessing necessary for this mapping in Section II. In Section III we investigate the structure and hardness of problem instances before we study the impact of discretization on the solution quality in Section IV. Afterwards, we discuss the mapping of the problem to a QUBO in Section V. We report on the embeddability of the QUBO instances and their solution quality on a D-Wave 2X device in Section VI. Finally, we conclude with discussion on improvements and future works. In the Appendix, we present more general mappings (including maneuvers) of the original deconflicting problem to QUBOs.

II. PROBLEM SPECIFICATION

The basic input of the conflict-resolution problem is a set of optimal flight trajectories (space-time paths). Such trajectories are the results of optimizations performed by the flight operators.

In the present study, we consider wind-optimal trajectories. Such trajectories are obtained by minimizing the fuel cost over the routes with given origins and destinations and desired departure times in the presence of forecast winds. Because of the correlation between such trajectories arising from exploiting favorable winds, these trajectories are likely to conflict; that is, two or more aircraft are likely to get dangerously close to each other if their optimal 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 limits constrain us to consider certain perturbations of the optimal trajectories. The simplest way to perturb a trajectory is to delay the corresponding flight on the ground prior to departure. These are the type of perturbations we mainly analyze in this work. We also consider local spatial modifications of the trajectories

so that no new potential conflicts are induced. Such local modifications can then parametrized as effective additional delay. Previous work [2] additionally considered a global modification of the trajectory geometrical shape.

A full accounting of the cost of such modifications would include 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 optimal trajectory $\mathbf{x}_i = (x_{i,t})_{t=\tau_{i,0}}^{\tau_{i,1}}$ is specified as a time-discretized path from the departure point $x_{i,\tau_{i,0}}$ at time $\tau_{i,0}$ to the arrival point $x_{i,\tau_{i,1}}$ at time $\tau_{i,1}$. 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 $\tau_{i,0}$ and $\tau_{i,1}$.

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 at the point (given the modifications under consideration). Then, the total delay over N_f flights is

$$D = \sum_{i=1}^{N_f} \delta_{i,\tau_{i,1}}. \quad (1)$$

A pair of flights (i, j) are in spatial conflict with each other if any pair of points from their trajectories is in conflict. That is, a pair of trajectory points $(x_{i,s}, x_{j,t})$, at time s and t respectively, conflict if their spatial and temporal separations are both less than the respective mandatory separation standards Δ_x and Δ_t :

$$\|x_{i,s} - x_{j,t}\| < \Delta_x, \quad (2a)$$

and

$$|(s + \delta_{i,s}) - (t + \delta_{j,t})| < \Delta_t. \quad (2b)$$

For the North Atlantic oceanic airspace, the separation standard are set to be: 30 nautical miles for horizontal separation Δ_x and 3 minutes for temporal separation Δ_t . Observe that 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

$$\max \{\delta_{i,s}^*, \delta_{j,t}^*\} + \Delta_t > |s - t|, \quad (3)$$

in which case we call the pair of trajectory points *potentially* conflicting. Let us partition the set C of potentially conflicting pairs of trajectory points into disjoint sets, or clusters, C_k :

$$C = \bigcup_k C_k, \quad (4)$$

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 $\{(i, s''), (j, t'')\} \in C_k$ and vice versa. We will further refer to such clusters C_k simply as the conflicts. Note that choosing $i < j$ in the definition is just a convention to uniquely determine the two flights involved in the conflict, and the index

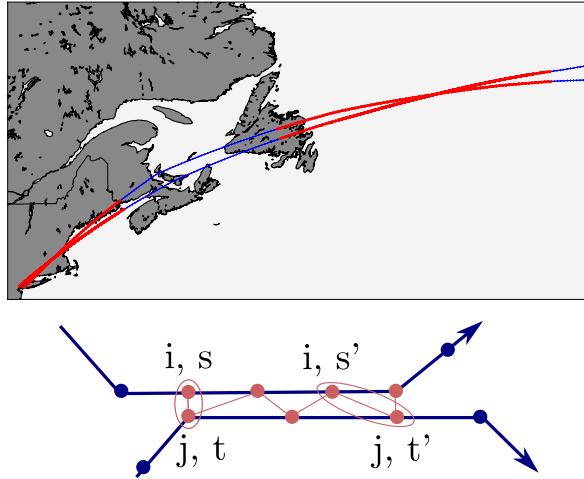


Fig. 1. Top: Example of two potential conflicts between a pair of transatlantic flights originating on the East Coast of the USA. Bottom: Conflict definition for a pair of flights i and j . If we have two pairs of conflicting trajectory points $\{(i, s), (j, t)\}, \{(i, s'), (j, t')\}$, all the intermediate points must also be in conflict to each other in order to qualify as a conflict cluster.

k has no meaning other than uniquely identifying conflicts. The purpose of clustering in this way is to extract a single constraint for each cluster than address all of the potential conflicts therein. Figure 1 shows an example of two such conflict clusters. 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, N_c the number of conflicts.

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 $\mathbf{d} = (d_i)_{i=1}^{N_f}$ 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

$$D = \sum_{i=1}^{N_f} \left(d_i + \sum_{k \in K_i} d_{i,k} \right). \quad (5)$$

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

We focus on the case where conflicts will be avoided only by the introduction of extra delays, leaving for future work the introduction of maneuvering choices.

Let

$$D_{i,k} = d_i + \sum_{k' \in K_{i,k}} d_{i,k'} \quad (6)$$

be the accumulated delay of flight i by the time it reaches conflict k , where $K_{i,k} = \{k' \in K_i | k' < k\}$. 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 . For simplicity, we assume that no delay is introduced during a conflict, so that $\delta_{i,s} = D_{i,k}$ for all s associated with flight i in conflict k . The pairs of conflicting trajectory points associated with conflict k are given by

$$T_k = \{(s, t) | \{(i, s), (j, t)\} \in C_k, i < j\}. \quad (7)$$

Thus the potential conflict is avoided only if

$$D_k = D_{i,k} - D_{j,k} \notin B_k \quad (8)$$

where

$$B_k = \bigcup_{(s,t) \in T_k} (-\Delta_t + t - s, \Delta_t + t - s) = [\Delta_k^{\min}, \Delta_k^{\max}], \quad (9)$$

$$\Delta_k^{\min} = 1 - \Delta_t + \min_{(s,t) \in T_k} \{t - s\}, \quad (10)$$

$$\Delta_k^{\max} = \Delta_t - 1 + \max_{(s,t) \in T_k} \{t - s\}. \quad (11)$$

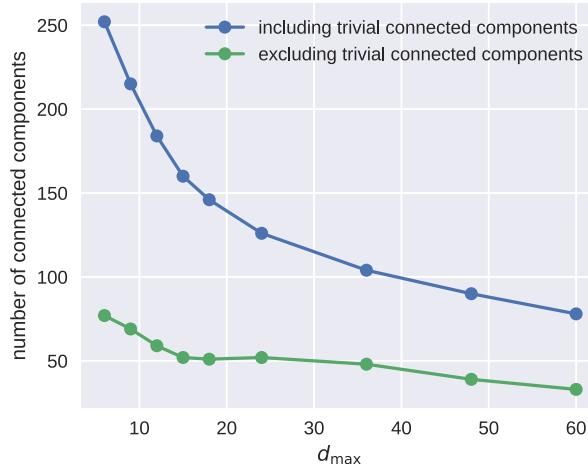
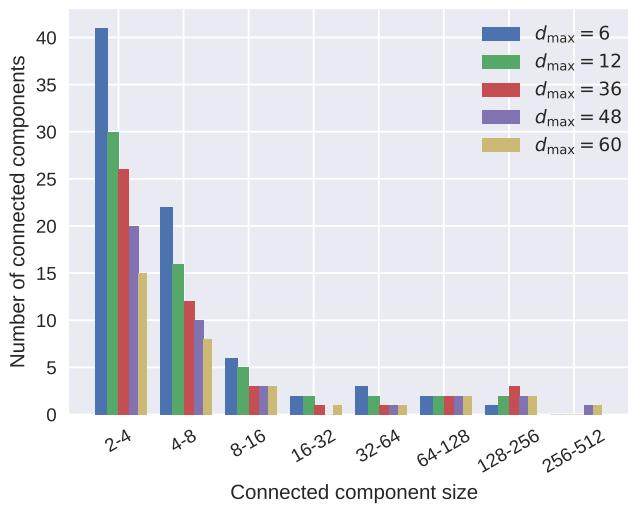
In the remainder of this paper, we focus on the simplified problem in which only departure delays are allowed. In this case, the configuration space is simply $\mathbf{d} = (d_i)_{i=1}^{N_f}$, the cost function (Eq. 5) transforms into $D = \sum_{i=1}^{N_f} d_i$, and the constraints become $d_i - d_j \notin B_k$ for all k .

III. INSTANCES

We test on realistic instances of the problem, using the precalculated wind-optimal trajectories for transatlantic flights on July 29, 2012 [2]. This data consists of 984 flights each of which has a constant cruising altitude and constant speed. However, our methods can be generalized to instances without these special properties.

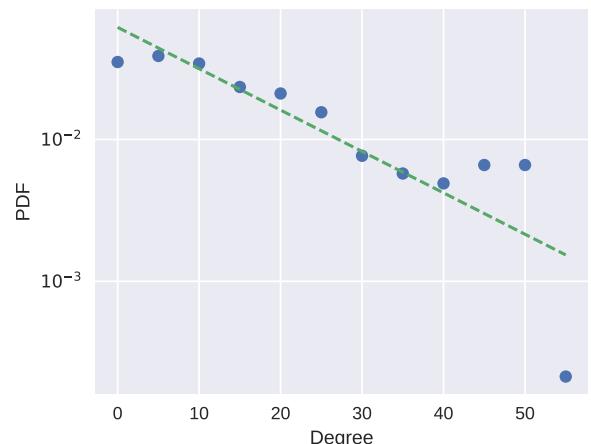
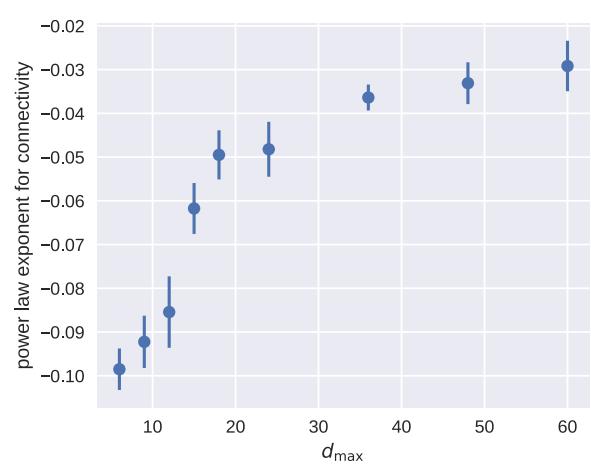
To identify the instances of the conflict-resolution problem we construct a *conflict graph*, whose vertices correspond to flights and which has an edge between a pair of vertices if there is at least one potential conflict between the corresponding flights. Note that the conflict graph for a given set of trajectories depends on the parameters of the problem. In the case of only departure delays, whether or not a potential conflict, and thus an edge in the conflict graph, exists between two flights is a function of the maximum allowable departure delay d_{\max} . For a certain value of d_{\max} , the conflict graph may contain several connected components, which can be considered as smaller, independent instances. Figure 2 shows this dependence of the number of connected components (both including and excluding trivial connected components, i.e. those containing a single vertex) on the maximum delay d_{\max} , and Figure 3 shows the distribution of the sizes of the connected components for various values of d_{\max} . As d_{\max} increases, the conflict graph becomes denser; at some point, the conflict graph saturates (though not necessarily as the complete graph), with every spatial conflict indicating a potential conflict. Interestingly, most of the connected components are very small; for example, with $d_{\max} = 60$ minutes, approximately 75% of the connected components contain no more than 10 flights.

In the remainder of this paper, we consider sets of smaller instances corresponding to the connected components of the

Fig. 2. Number of connected components versus d_{\max} .Fig. 3. Histogram of the connected component size for various values of the maximum delay time d_{\max} .

conflict graph from the larger single instance for various values of d_{\max} , for the given flight set. Let $\mathcal{I}_{d_{\max}}$ be the set of such instances for a particular value of d_{\max} , excluding trivial instances. We say that an instance is trivial if there are no conflicts when all flights therein depart without delay; in particular, this includes instances containing only a single flight.

As a part of our analysis, we also studied the probability distribution of the degree of vertices in the conflict graph. In other words, the number of flights for which a given flight share a potential conflict with. Figure 4 shows the distribution of degrees of vertices in the conflict graph for $d_{\max} = 60$, which seem to approximately coincide with a power law, i.e. the number of vertices with degree d is proportional to d^{α} . This is consistent with a so-called “small-world” model believed to be typical of many real-world graphs [8], which are generated by preferential attachment and resultingly contain a few number of highly-connected hubs, as is the case with air traffic. Figure 5 shows the dependence of this empirical power-law exponent α as a function of d_{\max} . As d_{\max}

Fig. 4. Histogram of the degrees of vertices in the conflict graph for $d_{\max} = 60$. The distribution of the degrees approximately follows a power law, with the exponent depending on d_{\max} .Fig. 5. Empirical power-law exponent versus d_{\max} . The error bars indicate the error obtained from the linear regression.

increases, the exponent decreases. The larger the delay, the less the structure of the trajectories matters and the flatter the distribution of degrees in the conflict graph.

In many cases, generally hard problems are easy when restricted to tree-like instances [9], [10]. For example, if the conflict graph here is a tree, then the optimum could be easily found by propagating the delays along the tree; on the other hand, if the conflict graph is a complete graph, finding the optimum is much harder. The tree-width of a graph formalizes this notion of tree-likeness, ranging from 1 for a tree to $n - 1$ for fully connected graph. We examine the treewidth of the connected components as a proxy for the hardness of the instances they represent.

Figure 6 shows that the treewidth of a connected component scales approximately linearly with its size. This suggests that realistic instances of the deconflicting are indeed hard, and not restricted to easier (bounded tree-width) instances of the generally hard problem. Moreover, the correlation γ between the tree-width of a connected component and its size increases with d_{\max} , as shown in Figure 7. The larger d_{\max} , the more

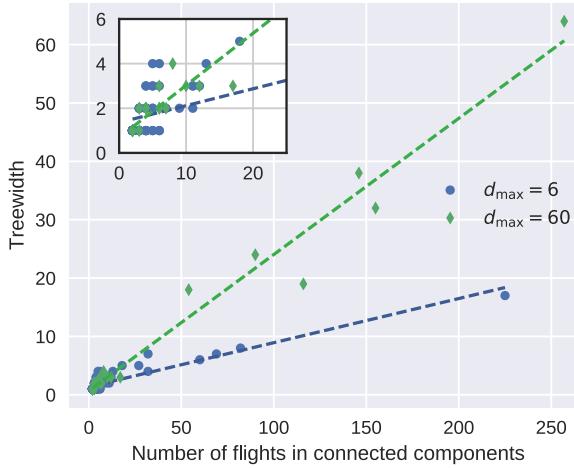


Fig. 6. The treewidths of connected components versus their sizes for various values of d_{\max} . The correlation is approximately linear, with a slope γ that depends on d_{\max} . The linear fit is representing the trend in the region with number of flights greater than 50.

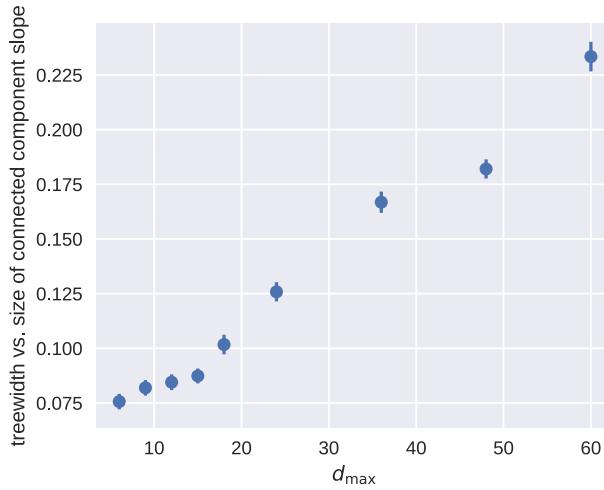


Fig. 7. Slope γ as a function of the maximum delay time. The error bars indicate the error obtained from the linear regression.

potential conflicts there are; restricting d_{\max} also restricts the number of conflicts.

IV. DISCRETIZING THE CONFIGURATION SPACE

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 $d_i \in \{\Delta_d l | l \in [0, 1, \dots, N_d]\}$, where d_i is the departure delay of flight i . The larger the configuration space is, the more qubits are needed to encode it, and so determining the effect of this discretization on solution quality is crucial to the effective use of quantum annealing. To do so, we solve the conflict-resolution problem with departure delays only for various delay resolutions and upper bounds and compare the various optima to the continuous problem without restrictions (other than non-negativity) on the delays.

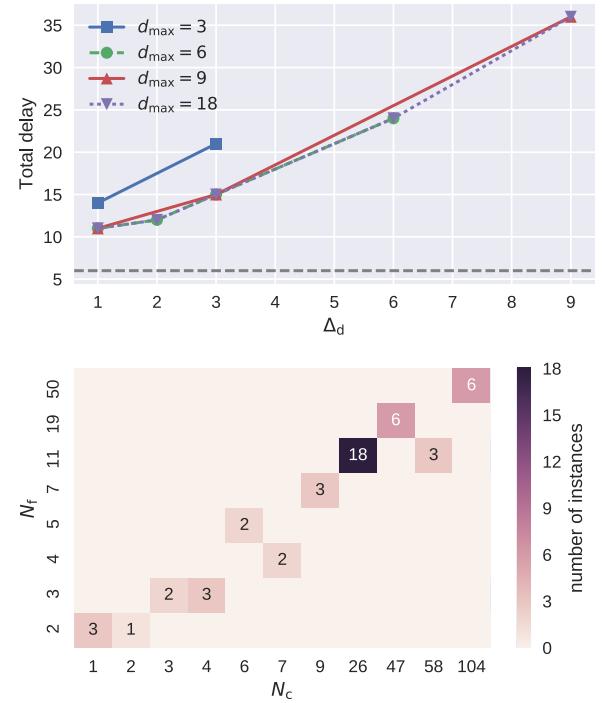


Fig. 8. Influence of discretization on the solution quality. Top: Minimum total delay of a problem instance from \mathcal{I}_{18} with 19 flights and 47 conflicts for various values of Δ_d and d_{\max} . Bottom: Results for continuous delay variables with upper bounds d_{\max} . We show the minimum upper bound d_{\max} necessary to obtain same result as that without bounding the delay. We used various instances in \mathcal{I}_{18} . The color code shows the number of instances with the same total delay.

We consider two sets of instances, \mathcal{I}_{18} and \mathcal{I}_{60} . For \mathcal{I}_{18} , the exact optima are found by modeling the problem as a constraint satisfaction problem [11]; the largest instance in \mathcal{I}_{18} has 50 flights and 104 potential conflicts.

The instances in \mathcal{I}_{60} are much larger and harder (the largest instance in \mathcal{I}_{60} has 257 flights and 4068 potential conflicts); we solved them by mapping to QUBO (as described in the next section) and then using the Isoenergetic Cluster Method (ICM) (a rejection-free cluster algorithm for spin glasses that greatly improves thermalization) [12], which has been shown to be one of the fastest classical heuristic to optimize QUBO problems [13]. Because ICM is a classical method, the penalty weights can be set arbitrarily large, ensuring that the desired constraints are satisfied. ICM is not guaranteed to return the global optimum in general. However, for the sizes of instances to which we applied ICM the results are sufficiently well converged to conclude that the solution found is indeed globally optimal with exceedingly high probability.

Figure 8 shows the minimum total delay of a problem instance with 19 flights and 47 potential conflicts from \mathcal{I}_{18} for various values of Δ_d and d_{\max} . With the exception of the small maximum delay $d_{\max} = 3$, the total delay of the solutions is nearly independent of the maximum delay. The total delay is non-decreasing with respect to the coarseness Δ_d of the discretization for a fixed maximum delay d_{\max} , and non-increasing with respect to d_{\max} for a fixed Δ_d . Since the original data is discretized in time in units of 1 minute,

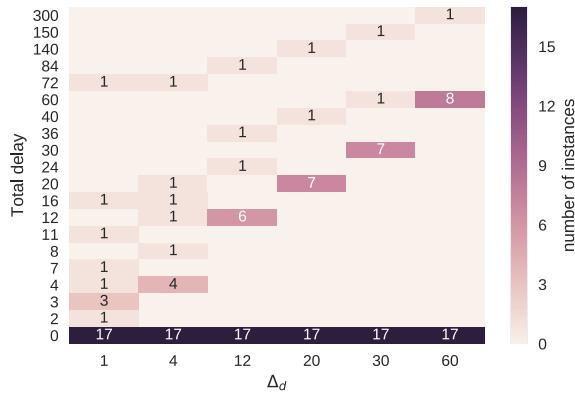


Fig. 9. Total delay in dependence of the discretization parameter Δ_d for 26 different problem instances from \mathcal{I}_{60} with up to in 12 flights and 25 conflicts. The color code shows the number of instances with the same total delay. 17 of these 26 instances had trivial solutions for all values of Δ_d , i.e the total delay vanishes.

$\Delta_d = 1$ yield the same result as a continuous variable with the same upper bound. Above some threshold value d_{\max}^0 , further increasing the maximum delay does not decrease the minimum total delay. With one exception, we found that for all the investigated problem instances $d_{\max}^0 \leq 6$ minutes (see figure 8). 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.

Figure 9 shows the dependence of the total delay time optimized by ICM on the delay discretization Δ_d for various problem instances extracted from the connected components of the conflict graph. Results are for maximum delay of 60 minutes. As expected, the total delay decreases by decreasing Δ_d . This is consistent with the idea that smaller Δ_d 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 its performance quickly decreases for fully-connected problems. Indeed, as shown in Section III, the underlying graph of connected components look more and more like a fully-connected graph rather than a tree graph as the number of flights inside the connected component increases.

V. MAPPING TO QUBO

In this section, we describe how to map to QUBO from the conflict-resolution problem limited to only departure delays; a more general mapping is given in the appendix.

A. Binary Encoding

Having suitably discretized the configuration space, we must then encode it into binary-valued variables. The value of d_i is encoded in $N_d + 1$ variables $d_{i,0}, \dots, d_{i,N_d} \in \{0, 1\}$ using a

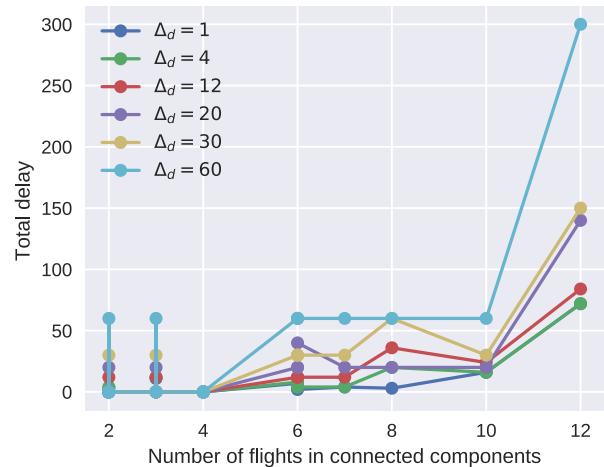


Fig. 10. Optimal total delay found by using the Isoenergetic Cluster Method (ICM) at fixed time step Δ_d as a function of numbers of flight within each connected component. ICM was unable to find solutions for connected component with more than 12 flights.

one-hot encoding:

$$d_{i,l} = \begin{cases} 1, & d_i = l, \\ 0, & d_i \neq l; \end{cases} \quad d_i = \Delta_d \sum_{l=0}^{N_d} l d_{i,l}. \quad (12)$$

To enforce this encoding, we add the penalty function

$$f_{\text{encoding}} = \lambda_{\text{encoding}} \sum_{i=1}^{N_f} \left(\sum_{l=0}^{N_d} d_{i,l} - 1 \right)^2, \quad (13)$$

where $\lambda_{\text{encoding}}$ is a penalty weight sufficiently large to ensure that any cost minimizing state satisfies $f_{\text{encoding}} = 0$. (Note that in practice, we could do away with the bit $d_{i,0}$ by removing it from (12) and substituting $\sum_{l=1}^{N_d-1} \sum_{l'=l+1}^{N_d} d_{i,l} d_{i,l'}$ into f_{encoding} .) In terms of these binary variables, the total delay contribution to the cost function is

$$f_{\text{delay}} = \Delta_d \sum_{i=1}^{N_f} \sum_{l=0}^{N_d} l d_{i,l}, \quad (14)$$

Lastly, actualized conflicts are penalized by

$$f_{\text{conflict}} = \lambda_{\text{conflict}} \sum_k \sum_{\substack{l, l' | \Delta_d(l-l') \in D_k \\ i, j \in I_k | i < j}} d_{i,l} d_{j,l}, \quad (15)$$

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

$$f = f_{\text{encoding}} + f_{\text{delay}} + f_{\text{conflict}}. \quad (16)$$

B. Softening the Constraints

In the QUBO formalism, there are no hard constraints; thus we use of penalty functions in the previous section. For sufficiently large penalty weights, the optimum will satisfy the desired constraints. However, precision is a limited resource in quantum annealing; therefore, we would like to determine the smallest sufficient penalty weights, at least at the level of the classical model [14].

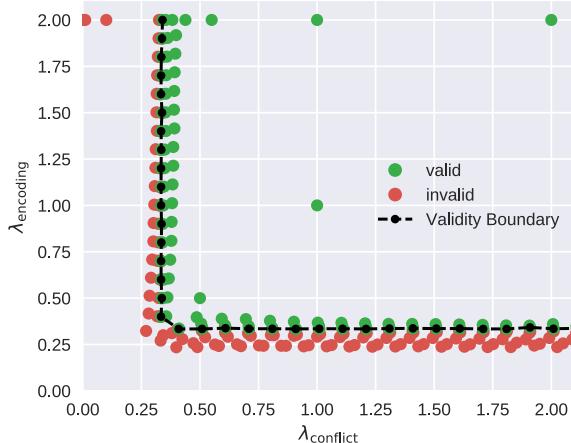


Fig. 11. Validity of exact solution to a QUBO extracted from a problem instance with $N_f = 7$ flights and $N_c = 9$ conflicts in dependence on the choice of the penalty weights, $\lambda_{\text{encoding}}$ and $\lambda_{\text{conflict}}$. Here, $\Delta_t = 6$ and $d_{\max} = 18$. In order to obtain the exact solutions, we used a Max-SAT solver [15] after we mapped the QUBO instances to Max-SAT instances.

For a given instance, we say that a pair of penalty weights $(\lambda_{\text{conflict}}, \lambda_{\text{encoding}})$ is valid if the minimum of the total cost function satisfies both the conflict and encoding constraints when using those weights. Figure 11 shows the phase space of these penalty weights for a single instance with 7 flights and 9 conflicts. The box-like boundary between valid and invalid penalty weights suggests that the validity of the two penalty weights is independent; this box-like boundary is found for all of our instances with up to 7 flights and 9 conflicts.

VI. QUANTUM ANNEALING

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

A. Background

Quantum annealing (QA) is a heuristic for minimizing pseudo-Boolean functions $f : \{1, -1\}^n \rightarrow \mathbb{R}$ using specially-designed quantum hardware. In physical terms, the goal is to find a ground state (i.e. lowest-energy state) of the “problem Hamiltonian”

$$\hat{H}_f = \sum_i h_i \hat{Z}_i + \sum_{i,j} J_{i,j} \hat{Z}_i \hat{Z}_j, \quad (17)$$

where \hat{Z}_i is the Pauli Z operator acting on the i th qubit and

$$f(\mathbf{s}) = \sum_i h_i s_i + \sum_{i,j} J_{i,j} s_i s_j \quad (18)$$

is the unique multilinear form of f . This is done by starting in a uniform superposition of the computational basis states, which is the ground state of the initial Hamiltonian

$$H_0 = \sum_i \hat{X}_i, \quad (19)$$

where \hat{X}_i is the Pauli X operator acting on the i th qubit. The adiabatic theorem implies that if we change the system’s

TABLE I
PARAMETERS OF THE LARGEST EMBEDDABLE
INSTANCES FOR THE D-WAVE 2X

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

Hamiltonian from the initial one to the final one slowly enough, then at all times the system will remain in its ground state, including at the end, which yields the ground state of the final Hamiltonian that we want. The essential principle is that excitations to higher-energy states are suppressed to a degree related by their difference in energy from the ground state. In practice, system noise and other factors mean that this ideal is not achieved, but often a low-energy state is obtained, i.e. practical QA only *approximately* minimizes the function f .

One obstruction to applying a particular quantum annealer to a given function is that the pairs of qubits $\{i, j\}$ for which a term $J_{i,j} \hat{Z}_i \hat{Z}_j$ can be included in the problem Hamiltonian H_f are restricted. These restrictions are captured in what we’ll call the “hardware graph”, whose vertices correspond to qubits and whose edges indicate for which pairs of qubits a term in the Hamiltonian can be included. Similarly, the vertices of the “problem graph” correspond to the Boolean variables s and the edges to the quadratic terms in the multilinear expansion. Because the hardware graph is inherently fixed and bounded-degree, the problem graph is usually not a subgraph thereof, meaning that we cannot directly assign each variable s_i to a single qubit. This is addressed by “graph-minor embedding”, in which each variable is mapped to a set of qubits. This is done in such a way that when the vertices of the hardware graph corresponding to each vertex of the problem graph are contracted into one, the resulting graph is isomorphic to the problem graph. Lastly, additional terms, called “intra-logical couplings” are added between the qubits that each variable is mapped to in order to ensure that they act as one, by penalizing states in which the state of those qubits are not the same.

B. 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 [5]. However, it is possible to map every QUBO to another QUBO which obeys the constraints of the Chimera architecture while increasing the number of binary variables used by a so called minor-embedding technique [16]

Of course the QUBO graph structure of the instances is not suitable for direct calculation on the D-Wave machine. Therefore we used D-Wave’s heuristic embedding algorithm [17] to embed instances with up to $N_f = 50$ and $N_c = 104$ on the D-Wave 2X as well as up to $N_f = 64$ and $N_c = 261$ on the D-Wave 2000Q depending on discretization (cf. Tables I and II). We generated up to 5 different

TABLE II
PARAMETERS OF THE LARGEST EMBEDDABLE
INSTANCES FOR THE D-WAVE 2000Q

Δ_d	3	6	9
Number of flights N_f	19	50	64
Number of conflicts N_c	47	104	261
Number of logical qubits	133	200	192
Average number of physical qubits	1235	1080	1121

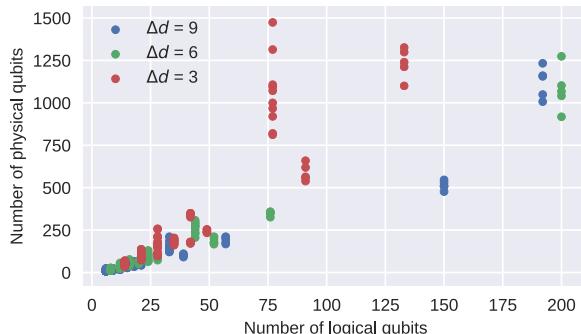


Fig. 12. Number of physical qubits versus the number of logical qubits for embeddings into a D-Wave 2000Q of the QUBO instances corresponding to \mathcal{I}_{18} .

embeddings for each QUBO instance, and selected the one that used the smallest number of physical qubits. In figure 12 one can see the dependence of the number of physical qubits on the number of logical qubits.

C. Success Probability

In order to investigate the performance of the D-Wave machines 2X and 2000Q, we compared the annealing results to the ones of an exact solver. We used an exact Max-SAT solver [15] after we mapped the QUBOs to Max-SAT. For each QUBO instance, we ran the annealing process in between 10^4 and 10^6 times. The success probability p is then given by the ratio of the number of annealing solutions which are equal to the exact solution and the number of total annealing runs. As a measure of the runtime of the machine, we used the time-to-solution with probability 99%.

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

where T_{Anneal} is the annealing time which was set to $20\mu\text{s}$.

In figure 13 the dependence of the time to solution T_{99} on the number of flights and the number of conflicts is shown. One can see, that the success probability decreases for larger problem instances as well as for finer discretizations. We conjecture, that this is mainly due to the limited precision in the specification of a QUBO on the D-Wave machines. In order to investigate the influence of limited precision, we need a measure for the precision needed to represent a given QUBO instance. If the embedded QUBO instance is given by $H = \sum_{ij} Q_{ij} x_i x_j$ with $x_i \in \{0, 1\}$ the corresponding Ising model

$$H = \sum_i h_i s_i + \sum_{ij} J_{ij} s_i s_j, \quad s_i \in \{-1, 1\},$$

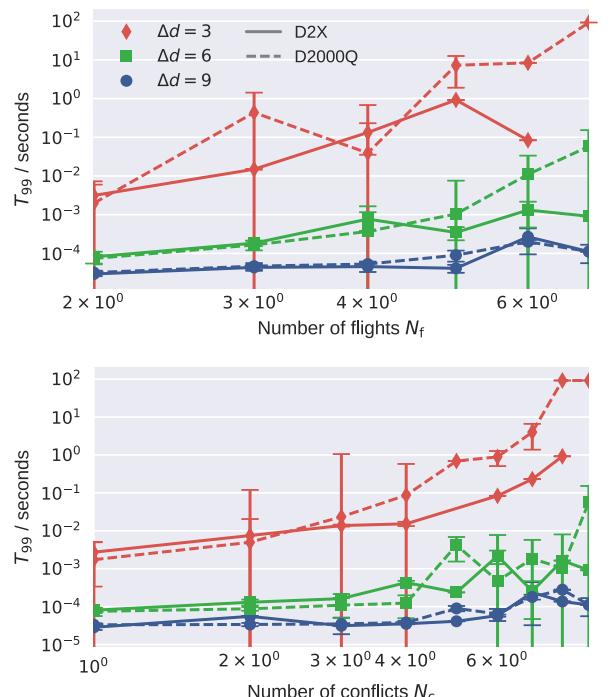


Fig. 13. Median of the time to solution with 99 % with probability T_{99} for QUBO instances in dependence of the number of flights N_f and the number of conflicts N_c . The error bars indicate the 35% and 65% percentiles. We used 10000 annealing runs for each instance, penalty weights $\lambda_{\text{conflict}} = \lambda_{\text{encoding}} = 1$ and 5 different embeddings. The ferromagnetic coupling between all physical qubits of the same logical qubit was set to $J_F = -1$ in absolute units for all the instances. The solid lines are results from the D-Wave 2X whereas the dashed lines are results from the D-Wave 2000Q. For these results, we did not use gauges and used energy minimization to deal with broken qubit chains.

can be obtained by the transformation $s_i = 2x_i - 1$. A measure for the precision needed is then given by the *maximum coefficient ratio*

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

The larger this number is, the finer precision is needed for correctly representing the QUBO on a D-Wave machine.

The success probability also depends on the choice of the ferromagnetic intra-logical qubit coupling J_F . Figure 14 shows the dependence of the success probability on this coupling for one particular problem instance. However, the general behavior of this curve is instance independent. For very small J_F , the qubit strings which represent logical qubits might be broken by coupling to outer logical qubits and the success probability is suppressed. On the other hand, if J_F is very large, the precision needed C_{\max} will eventually surpass the machine precision and the success probability will decrease. In between these two extrema, there will be a sweet spot with an optimal J_F^{opt} which yield maximum success probability. We determined the optimal coupling J_F^{opt} for the problem instances by sweeping over values in between $J_F = -1.25$ to $J_F = -0.125$ in units of the largest coefficient of the embedded Ising model. The inset in figure 14 shows the distribution of the J_F^{opt} . Using the optimal couplings, the performance is increased with

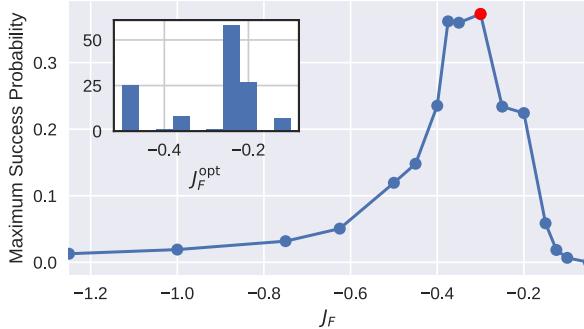


Fig. 14. Maximum success probability on the D-Wave 2000Q for a QUBO instance with $N_f = 5$, $N_c = 5$ and $\Delta_d = 6$ in dependence of J_F , where J_F is given in units of the largest coefficient of the embedded Ising model. We used 5 different embeddings, 100000 annealing runs and penalty weights $\lambda = \lambda_{\text{conflict}} = \lambda_{\text{unique}} = 1$ for each of the data points. The red data point indicates the optimal value J_F^{opt} . The inset shows the distribution of J_F^{opt} for all solvable instances.

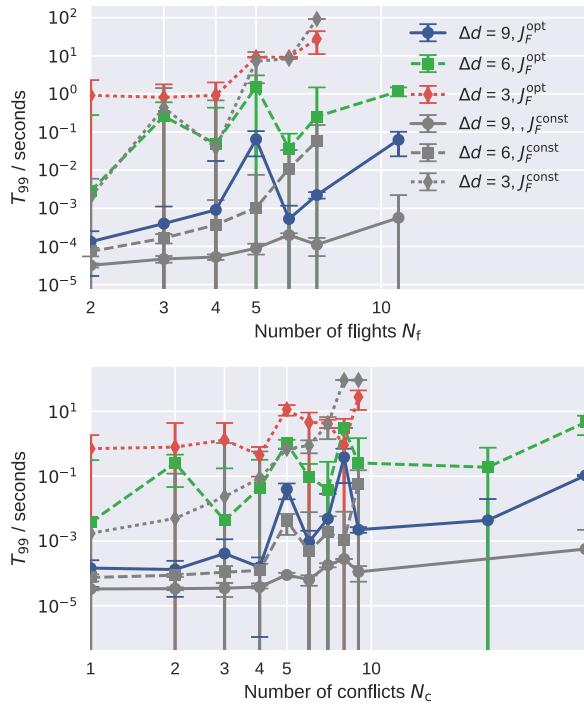


Fig. 15. Median of the time to solution with 99 % with probability T_{99} for fixed and optimal J_F . The error bars indicate the 35% and 65% percentiles. We used up to 1000000 annealing runs on the D-Wave 2000Q for each instance, penalty weights $\lambda_{\text{conflict}} = \lambda_{\text{encoding}} = 1$ and up to 5 different embeddings. The colored lines indicate results obtained with an optimal $J_F = J_F^{\text{opt}}$ which is instance dependent. The gray lines indicate results obtained with a fixed $J_F = J_F^{\text{const}} = -1$ in absolute units.

respect to a constant value $J_F^{\text{const}} = -1$ in absolute units as one can see in figure 15.

Using the optimal coupling J_F^{opt} we can study the influence of the limited machine precision on the success probability. Figure 16 shows the maximum success probability with optimal J_F for all embeddable instances in \mathcal{I}_{18} . The influence of the limited machine precision can be seen in the decrease of the success probability with increasing precision C_{\max} . The success probability vanishes around $C_{\max} \approx 30$ which corresponds to the machine precision of the D-Wave 2000Q

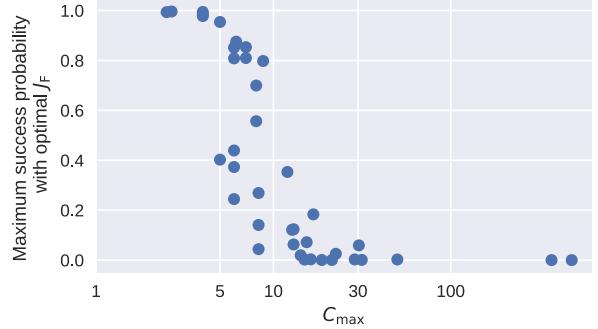


Fig. 16. Maximum success probability on the D-Wave 2000Q for optimal J_F for all embeddable instances in \mathcal{I}_{18} against the coefficient ratio C_{\max} .

of around $\sim 1/30$. Since C_{\max} in general increases with the problem size as well as with finer discretizations, this explains the long time-to-solutions for large instances and fine discretizations in figures 13 and 15.

VII. CONCLUSIONS

Quantum annealing is a relatively new heuristic that offers hope of solving classical optimization problems better in several ways compared to classical alternatives: higher-quality solutions, faster time to solution, many approximate solutions, and qualitatively different solutions. Whether this hope will likely have to be answered empirically, by running problems on actual hardware. This work is a first step towards that end, but not the last. In particular, the restriction to time delays only is obviously unrealistic. Nevertheless, this simplification both serves a proof of principle and enables runs on extant quantum annealers at a reasonable problem scale. To be practically relevant, this line of work must be extended to include in the QUBO all practically relevant aspects of the problem. The full problem can then be solved using other promising QUBO solvers such as ICM, which may improve on the state of the art even if quantum annealing does not.

In this paper, we propose a novel QUBO mapping for a simplified version of the Air Traffic Management (ATM) conflict-resolution problem for wind-optimal trajectories involving minimum trajectory modifications. Although these efforts are driven by making the problem amenable to quantum annealers, the techniques used may be beneficial also for the classical solution of the problem. In our study, we considered the actual wind-optimal trajectories for transatlantic flights (NAT) on July 29, 2012. Given the large number of flights, the wind-optimal trajectories cannot be directly mapped in a QUBO model. To overcome this limitation, our modified version of the conflict-resolution problem assumes that the flight maneuvers applied to avoid conflicts modify the wind-optimal trajectories only locally, resulting in assigning “delays” to the flights. Therefore, wind-optimal trajectories can be “hard encoded” in our QUBO formulation of the conflict-resolution problem with the flights delays being the only variables to optimize. Nevertheless, as explained in Appendix 2, our method is general enough to potentially include the effect of other maneuvers as well.

As part of our study, we also introduce a novel “pre-processing” algorithm to eliminate potential conflicts that,

given a maximum delay, can never occur and clustering adjacent conflicts. This novel approach does not only reduce the number of potential conflicts, but is also gives an important indication of the underlying topology the conflict graph. Indeed, we have discovered that most of the flights have very few conflicts while there are few flights that have conflicts in a non trivial way. The latter sets of flights represent the hardest part of the conflict-resolution problem to optimize. We want to emphasize that the proposed pre-processing algorithm is general and can be successfully applied to the existing conflict-resolution methods to improve both the speed and quality of solutions.

We also present several different QUBO mapping including local and global trajectory deviations as well as including and excluding maneuvers. Due to the hardware limitations of the D-Wave machine we focus on a model excluding maneuvers in order to keep the number of variables small. Using D-Wave's embedding algorithm, several smaller problem instances were embeddable onto the D-Wave 2X as well as onto the D-Wave 2000Q. However, the limiting factor for the success probability is not the sizes of the chips but its limited precision. Therefore the success probability is suppressed for finer discretizations and larger problem sizes.

Finally, we have analyzed the performance of both classical and quantum heuristics in solving the QUBO model where only delays at the departure are allowed. Results show that it is already hard to find conflict-free solution for a flight set that involve more than 12 flights.

This work represents the foundation for future work, including:

- Embed and solve QUBO instances for models that benefits from variable simplification (see section V), also including maneuvers on a quantum annealer.
- Improve performance of quantum annealing by alternative embedding strategies and advanced annealing schedules available on newer D-Wave devices.
- Use best-available classical solvers that exploit the conflict graph description for classical solutions to the problem. The studied problem is related to several works in multi-agent path planning [18], [19] and no-wait job-shop-scheduling problems [20] and the quantum annealing results could be benchmarked with the domain-specific solvers of these related problems, after careful mapping.

APPENDIX

In this section we describe a mapping to QUBO of a more general version of the deconflicting problem than that covered in the main text.

A. Alternative Encodings

In the mappings describe both in the main text and the appendix, we use a one-hot encoding to encode a variable. This is best for the specific mappings we described, but in variants an alternative may be better. Say we have a variable x that we want to allow to have variables from finite set $W = \{w_1, w_2, \dots, w_m\}$. The one-hot encoding has m bits

$(x_i)_{i=1}^m$ such that $x = \sum_{i=1}^m w_i x_i$ and $\sum_{i=1}^m x_i = 1$. While we focus on the case in which $W = \{0, 1, \dots, m - 1\}$, our methods are not dependent on that being case, and in particular can address non-uniform sets of values, say if via clever preprocessing it can be determined that such a set would be sufficient. An alternative encoding would remove the requirement that exactly one of the bits is one. The variable x would still be encoded as $x = \sum_{i=1}^m w_i x_i$, but without the one-hot constraint can take on values in $\{\sum_i b_i w_i | b_i \in \{0, 1\}\}$. In particular, this encompasses the unary encoding in which $w_i = 1$ for all i and thus $x \in [0, m]$, as well as the binary encoding $w_i = 2^{i-1}$ for which $x \in [0, 2^m - 1]$. The latter has the advantage of requiring much fewer qubits, but at the cost of similarly increased precision. The former requires the same number of qubits as the one-hot encoding we use, and even has the benefit of minimal precision, but does not allow for quadratic constraints that penalize certain pairs of values of variables, e.g. $d_i - d_j \neq B_k$, without the use of ancillary bits. In models in which the bits x_i only appear in the sum $\sum_i x_i$, it is actually preferable to use the unary encoding to improve the precision requirements. We stick to the one-hot encoding for simplicity, but in practice the unary encoding should be used when possible.

To make the expressions more concise, we define the generalized encoding penalty function

$$f_{\text{encoding}}(\{X_i\}_i) = \lambda_{\text{encoding}} \sum_i \left(\sum_{x \in X_i} x - 1 \right)^2 \quad (20)$$

that enforces the constraint that exactly one bit x is one for each set of bits X_i .

B. Global Trajectory Modifications

Consider the case in which each trajectory can be modified by a departure delay and some parameterized spatial transformation, i.e. for each flight i there is a variable d_i and some parameter θ_i . For example, Rodionova *et al.* [2] consider a single angle θ_i that determines a sinusoidal transformation of the trajectory. For the QUBO mapping, we require that these variables be allowed to take on values from some finite set, so that are QUBO variables are $\{d_{i,\alpha}\}$ and $\{\theta_{i,\phi}\}$, where $d_{i,\alpha} = 1$ ($d_{i,\alpha}$) indicates that $d_i = \alpha$ ($d_i \neq \alpha$) and similarly for $\theta_{i,\phi}$. For every pair of flights $i < j$, we can efficiently (in time and space polynomial in the size of the input) compute whether the corresponding trajectories conflict when modified according to d_i, d_j, θ_i and θ_j . Let $B_{i,j}$ be the set of values of $(d_i, \theta_i, d_j, \theta_j)$ such that the the modified trajectories conflict. Lastly, let $d_{(i,\alpha),(j,\beta)} = 1$ indicate that $d_i = \alpha$ and $d_j = \beta$, and similarly for $\theta_{(i,\phi),(j,\psi)}$. The overall cost function is

$$\begin{aligned} f_{\text{global}} & \left((d_{i,\alpha})_{i,\alpha} (d_{(i,\alpha),(j,\beta)})_{i,j,\alpha,\beta} (\theta_{(i,\phi),(j,\psi)})_{i,j,\phi,\psi} \right) \\ & = f_{\text{encoding}} + f_{\text{consistency}} + f_{\text{delay}} + f_{\text{conflict}}, \end{aligned} \quad (21)$$

where

$$f_{\text{encoding}} \left(\left(\{d_{i,\alpha}\}_{\alpha} \cup \{\theta_{i,\phi}\}_{\phi} \right)_i \right) \quad (22)$$

ensures that the values of d_i and θ_i are uniquely encoded;

$$\begin{aligned} f_{\text{consistency}} = \lambda_{\text{consistency}} & \left[\sum_{i < j, \alpha, \beta} s(d_{i,\alpha}, d_{j,\beta}, d_{(i,\alpha),(j,\beta)}) \right. \\ & \left. + \sum_{i < j, \phi, \psi} s(\theta_{i,\phi}, \theta_{j,\psi}, \theta_{(i,\phi),(j,\psi)}) \right] \end{aligned} \quad (23)$$

ensures consistency between the values of $d_{i,\alpha}$, $d_{j,\beta}$, and $d_{(i,\alpha),(j,\beta)}$;

$$s(x, y, z) = 3z + xy - 2xz - 2yz \quad (24)$$

is a non-negative penalty function that is zero if and only if $z = xy$;

$$f_{\text{delay}} = \sum_{i,\alpha} \alpha d_{i,\alpha} \quad (25)$$

is the cost function to be minimized; and

$$f_{\text{conflict}} = \lambda_{\text{conflict}} \sum_{i < j} \sum_{(\alpha, \phi, \beta, \psi) \in B_{i,j}} d_{(i,\alpha),(j,\beta)} \theta_{(i,\phi),(j,\psi)} \quad (26)$$

penalize conflicts.

C. Local Trajectory Modifications

Alternatively, we can consider modifications to the trajectory only near conflicts. We describe a few special models and their mapping to QUBO, though many more such ways of doing so, and we leave a full accounting for future work.

1) *Exclusive Avoidance*: Suppose for every conflict k and associated pair of flights $i < j$, there is a way for either flight to go around the trajectory of the other, introducing some delay $d_{i,k}$ to flight i or $d_{j,k}$ to flight j depending on which trajectory is changed. Let $a_k = a_{i,k} = 1$ ($a_{i,k} = 0$) indicate that flight i 's trajectory is changed (unchanged), and for convenience let $a_{j,k} = 1 - a_{i,k}$, though only one (qu)bit will be used per conflict. Adding in the departure delay, we have the total cost function

$$f_{\text{exclusive}} \left((d_{i,\alpha})_{i,\alpha}, (a_k)_k \right) = f_{\text{delay}} + f_{\text{encoding}}, \quad (27)$$

where

$$f_{\text{delay}} = \sum_i \left[\sum_{\alpha} \alpha d_{i,\alpha} + \sum_{k \in K_i} d_{i,k} a_{i,k} \right] \quad (28)$$

and f_{encoding} is as in (13). This assumes that the trajectory modifications don't introduce potential conflicts with other flights; this assumption can be partially relaxed by adding penalty terms of the form $a_{i,k} a_{j,k'}$ or $d_{i,\alpha} a_{j,k}$ as appropriate.

2) *Flexible Avoidance*: In the exclusive avoidance model, it is *required* that one or the other flight is delayed at each conflict. We can relax this by accounting for the fact that if the flights arriving at a potential conflict are already relatively delayed, the conflict could be passively avoided (i.e. with no active maneuver). Let $D_{k,\gamma} = 1$ ($D_{k,\gamma} = 0$) indicate that $D_k = \gamma$ ($D_k \neq \gamma$), where D_k is the difference in the accumulated delays at conflict k as defined in (8).

The total cost function is

$$\begin{aligned} f_{\text{flexible}} \left((d_{i,\alpha})_{i,\alpha}, (a_k)_k, (D_{k,\gamma})_{k,\gamma} \right) \\ = f_{\text{encoding}} + f_{\text{delay}} + f_{\text{consistency}} + f_{\text{conflict}}, \end{aligned} \quad (29)$$

where the first term is

$$f_{\text{encoding}} \left(\left\{ \{d_{i,\alpha}\}_\alpha \right\}_i \cup \left\{ \{D_{k,\gamma}\}_\gamma \right\}_k \right); \quad (30)$$

the consistency term is

$$f_{\text{consistency}} = \lambda_{\text{consistency}} \sum_k \left(D_{i,k} - D_{j,k} - \sum_{\gamma} \gamma D_{k,\gamma} \right)^2 \quad (31)$$

using the notational variables

$$D_{i,k} = \sum_{\alpha} \alpha d_{i,\alpha} + \sum_{k' \in K_{i,k}} d_{i,k'} a_{i,k'}; \quad (32)$$

f_{delay} is as in (28) but where $a_{i,k}$ and $a_{j,k}$ are separate bits; and

$$f_{\text{conflict}} = \lambda_{\text{conflict}} \sum_k \sum_{\gamma \in B_k} [D_{k,\gamma} (1 - a_{i,k} - a_{j,k}) + 2a_{i,k} a_{j,k}] \quad (33)$$

If we want to allow both flights to be delayed at conflict $a_{i,k} = a_{j,k} = 1$, we must introduce an ancillary bit a_k that indicates whether at least one flight is delayed at conflict k , adding

$$\lambda_{\text{consistency}} \sum_k [(a_{i,k} + a_{j,k}) (1 - 2a_k) + a_{i,k} a_{j,k}] \quad (34)$$

to $f_{\text{consistency}}$, and replacing f_{conflict} with

$$\sum_k \sum_{\gamma \in B_k} D_{k,\gamma} (1 - a_k). \quad (35)$$

3) *Interstitial Delays*: In the interstitial-delay model, the local modifications are not made *at* conflicts but *between* them, and conflicts are only avoided via accumulated delays. That is, the delay $d_{i,k}$ introduced to flight i before reaching conflict k but after leaving the previous conflict $\kappa_{i,k} = \max_{k' \in K_{i,k}} k'$. Unlike in the flexible avoidance model, $d_{i,k}$ is now a variable rather than a parameter, and we encode it using bits $d_{i,k,\delta}$.

$$\begin{aligned} f_{\text{interstitial}} \left((d_{i,\alpha})_{i,\alpha}, (D_{i,k,\gamma})_{i,k \in K_i, \gamma} \right) \\ = f_{\text{encoding}} + f_{\text{consistency}} + f_{\text{conflict}} + f_{\text{delay}}, \end{aligned} \quad (36)$$

where

$$f_{\text{encoding}} \left(\left\{ \{d_{i,\alpha}\}_\alpha \right\}_i \cup \bigcup_i \left\{ \{D_{i,k,\gamma}\}_\gamma \right\}_{k \in K_i} \right), \quad (37)$$

$$f_{\text{consistency}} = \sum_i \sum_{k \in K_i} \sum_{(\gamma, \gamma') \in B_{i,k}} D_{i,k,\gamma} D_{i,\kappa_{i,k}, \gamma'}, \quad (38)$$

$$f_{\text{conflict}} = \lambda_{\text{conflict}} \sum_{k=1}^{N_c} \sum_{(\gamma, \gamma') \in B_k} D_{i,k,\gamma} D_{j,k,\gamma'}, \quad (39)$$

and

$$f_{\text{delay}} \sum_i \sum_{\gamma} D_{i,\max K_i, \gamma}. \quad (40)$$

ACKNOWLEDGEMENTS

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright annotation thereon.

REFERENCES

- [1] H. K. Ng, B. Sridhar, and S. Grabbe, "Optimizing aircraft trajectories with multiple cruise altitudes in the presence of winds," *J. Aerosp. Inf. Syst.*, vol. 11, no. 1, pp. 35–47, Jan. 2014, doi: [10.2514/1.I010084](https://doi.org/10.2514/1.I010084).
- [2] O. Rodionova, D. Delahaye, B. Sridhar, and H. Ng, "Deconflicting wind-optimal aircraft trajectories in north atlantic oceanic airspace," in *Proc. Adv. Aircr. Efficiency Global Air Transport Syst. Conf. (AEGATS)*, 2016. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01304633/>
- [3] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 4, pp. 179–189, Dec. 2000.
- [4] E. Hojjat and N. Kamyar, "A comparison framework for conflict detection and resolution multi agent modeling methods in air traffic management," *Int. J. Inf. Technol., Control Autom.*, vol. 2, no. 4, pp. 51–64, 2012.
- [5] 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 Inf. Process.*, vol. 14, no. 1, pp. 1–36, 2015, doi: [10.1007/s11128-014-0892-x](https://doi.org/10.1007/s11128-014-0892-x).
- [6] E. G. Rieffel, D. Venturelli, M. Do, I. Hen, and J. Frank, "Parametrized families of hard planning problems from phase transitions," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2337–2343. [Online]. Available: <https://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8614/8697>
- [7] D. Venturelli, D. J. J. Marchand, and G. Rojo. (2015). "Quantum annealing implementation of job-shop scheduling." [Online]. Available: <https://arxiv.org/abs/1506.08479>
- [8] R. Albert, H. Jeong, and A.-L. Barabási, "Diameter of the World-Wide Web," *Nature*, vol. 401, no. 6749, pp. 130–131, 1999.
- [9] U. Bertele, *Nonserial Dynamic Programming*. New York, NY, USA: Academic, 1972.
- [10] R. Halin, "S-functions for graphs," *J. Geometry*, vol. 8, nos. 1–2, pp. 171–186, 1976.
- [11] E. Hebrard, E. O'Mahony, and B. O'Sullivan, "Constraint programming and combinatorial optimisation in numberjack," in *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Berlin, Germany: Springer, 2010, pp. 181–185, doi: [10.1007/978-3-642-13520-0_22](https://doi.org/10.1007/978-3-642-13520-0_22).
- [12] Z. Zhu, A. J. Ochoa, and H. G. Katzgraber, "Efficient cluster algorithm for spin glasses in any space dimension," *Phys. Rev. Lett.*, vol. 115, no. 7, p. 077201, 2015.
- [13] 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," *Phys. Rev. A, Gen. Phys.*, vol. 94, no. 2, p. 022337, 2016.
- [14] Z. Zhu, A. J. Ochoa, S. Schnabel, F. Hamze, and H. G. Katzgraber, "Best-case performance of quantum annealers on native spin-glass benchmarks: How chaos can affect success probabilities?" *Phys. Rev. A, Gen. Phys.*, vol. 93, no. 1, p. 012317, 2016.
- [15] A. Kügel, "Improved exact solver for the weighted max-SAT problem," *POS-10. Pragmatics SAT*, vol. 8, pp. 15–27, May 2013. doi: [10.29007/38lm](https://doi.org/10.29007/38lm)
- [16] V. Choi. (2008). "Minor-embedding in adiabatic quantum computation: I. The parameter setting problem." [Online]. Available: <https://arxiv.org/abs/0804.4884>
- [17] J. Cai, W. G. Macready, and A. Roy. (2014). "A practical heuristic for finding graph minors." [Online]. Available: <https://arxiv.org/abs/1406.2741>
- [18] S. Akella and S. Hutchinson, "Coordinating the motions of multiple robots with specified trajectories," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 1, May 2002, pp. 624–631.
- [19] X. Wang, M. Kloetzer, C. Mahulea, and M. Silva, "Collision avoidance of mobile robots by using initial time delays," in *Proc. IEEE 54th Annu. Conf. Decis. Control (CDC)*, Dec. 2015, pp. 324–329.
- [20] S. Sahni and Y. Cho, "Complexity of scheduling shops with no wait in process," *Math. Oper. Res.*, vol. 4, no. 4, pp. 448–457, 1979.



Tobias Stollenwerk received the German Diploma degree in physics from the University of Bonn in 2007. He was a Research Assistant in computational photonics in Bonn for two years. From 2009 to 2013, he did his Ph.D. research in theoretical condensed matter physics with the Group of J. Kroha, University of Bonn. In 2016 and 2017, he was a Visiting Researcher with the Quantum Artificial Intelligence Laboratory, NASA Ames. Since 2013, he has been with the German Aerospace Center, focusing on quantum computing.



Bryan O'Gorman received the A.B. degree in physics from the Harvard College in 2013. He is currently pursuing the Ph.D. degree in computer science with the University of California, Berkeley. Since 2013, he has been with the Quantum Artificial Intelligence Laboratory, NASA Ames Research Center. He is currently a NASA Space Technology Fellow with the University of California, Berkeley.



Davide Venturelli received the degree from the Ecole Normale Supérieure de Lyon, the Ph.D. degree in numerical simulations of the condensed matter from the International School for Advanced Studies (SISSA), Trieste, and the Ph.D. degree in nanophysics from the Université de Grenoble (CNRS/UJF). From 2010 to 2011, he was a Post-Doctoral Researcher with the Condensed Matter and Information Theory Group, Scuola Normale Superiore, Pisa, Italy. In 2012, he joined the NASA Intelligent System Division (TI) as one of the founding members of the Quantum Artificial Intelligence Laboratory, where he is under the NASA Academic Mission Service Contract. He is also the Quantum Computing Team Lead and the Science Operations Manager of the Research Institute of Advanced Computer Science, Universities Space Research Association. He currently invested in research projects dealing with quantum optimization applications and their near-term implementation in the real hardware.



Salvatore Mandrà received the Ph.D. degree in theoretical physics from the University of Milan, Italy, in 2013. He was a Post-Doctoral Researcher with Harvard University, where he focused on quantum annealing and quantum computation. In 2016, he joined the Quantum Artificial Intelligence Lab, NASA Ames. His expertise ranges from the theoretical development of new classical/quantum algorithms as well as the numerical optimization of classical/quantum simulations (including high-level programming in C/C++ and distributed programming in MPI/OpenMP).



Olga Rodionova received the master's degree in applied mathematics and computer science from Saint-Petersburg State Polytechnical University, Saint Petersburg, Russia, in 2011, and the Ph.D. degree in applied mathematics from French Civil Aviation University, Toulouse, France, in 2015. From 2015 to 2017, she continued her research in air traffic management as a Post-Doctoral Fellow at the NASA Ames Research Center, CA, USA, focusing on aircraft trajectory and airspace load optimization. She is currently a Research and Development Engineer with Innov'ATM, Cugnaux, France.



Hokkwan Ng received the Ph.D. degree in mechanical and aerospace engineering from the University of California, Los Angeles. He was in aviation related researches and applications for the past ten years. He is currently a Research Aerospace Engineer with NASA Ames Research Center. He specializes in the algorithm development for the air traffic management system. His current research interests include air traffic management and disruptions management for urban air mobility and unmanned aircraft systems.



Banavar Sridhar is currently a Research Associate with the NASA Ames Research Center. Earlier, he served as the NASA Senior Scientist for air transportation systems. His research interests are in the applications of modeling and optimization techniques to aerospace systems. He has received the 2004 IEEE Control System Technology Award for his contributions to the development of modelling and simulation techniques for multi-vehicle traffic networks. He led the development of traffic flow management software, Future ATM Concepts Evaluation Tool, which received the AIAA Engineering Software Award in 2009. He has also received the NASA Invention of the Year Award in 2010 and the FAA Award for the Excellence in Aviation Research in 2010. He is a fellow of the AIAA.



Eleanor Gilbert Rieffel received the Ph.D. degree in mathematics from the University of California, Los Angeles. She was with FXPAL, where she performed research in diverse fields including quantum computation, applied cryptography, image-based geometric reconstruction of 3-D scenes, bioinformatics, video surveillance, and automated control code generation for modular robotics. In 2012, she joined the NASA Ames Research Center, to work on their expanding quantum computing effort. She leads the Quantum Artificial Intelligence Laboratory, NASA Ames Research Center. She has co-authored the book *Quantum Computing: A Gentle Introduction* (MIT press, 2011), along with Wolfgang Polak. Her research interests include quantum heuristics, evaluation and utilization of near-term quantum hardware, fundamental resources for quantum computation, quantum error suppression, and applications for quantum computing.



Rupak Biswas received the Ph.D. degree in computer science from Rensselaer in 1991. Since 1991, he has been with NASA. He is currently the Director of Exploration Technology with the NASA Ames Research Center, CA, USA. In this role, he is in charge of planning, directing, and coordinating the technology development and operational activities of the organization that comprises advanced supercomputing, human systems integration, intelligent systems, and entry systems technology. He is also the Manager of the NASA's High End Computing Project that provides a full range of advanced computational resources and services to numerous agency programs. He is an internationally recognized expert in high-performance computing and has published more than 160 technical papers, has edited several journal special issues, and has given numerous lectures around the world. He received many best paper awards.