

Exploration of the Potential of Quantum Annealing for Hard Scheduling Problems in Air Traffic Management: Report, July 2016

Introduction

Given encouraging of early results in the planning domain [1, 2] and the expertise of our team in such sector, aim of the project is to use Quantum Annealing (QA), and in particular the state-of-art D-Wave 2XTM quantum annealer hosted at NASA Ames, as a metaheuristic for solving computational challenging problems in the context of Air Traffic Management (ATM). More precisely, the project has been focused on the problem to find distinct minimum-cost configuration scheduling advisories (CSAs). In this project, we limit our attention to the North Atlantic oceanic airspace (NAT) for which we have optimal-wind trajectories for two consecutive days (July 28th-29th 2012).

Since the D-Wave 2XTM quantum chip can only optimize binary optimization problems on a specific and fix-by-design architecture called Chimera, in the first part of the project we mainly focused on the formulation of CSAs in terms of discrete variables (Table 1 for tasks/milestones of the project). The discrete model is then expressed in terms of quadratic binary optimization (QUBO) problems that can be natively solved by the D-Wave 2XTM quantum chip. The next set of milestones will include the run of CSA instances on the D-Wave 2XTM quantum chip, as well as the comparison of the results with the best state-of-art classical QUBO optimizers.

Approach and technical details

NAT dataset consists in XXX¹ wind-optimal trajectories in (3+1)-dimensions. Since wind-optimal trajectories have been computed independently from each others, two or more trajectories can intersect in space creating a “conflict”. A conflict becomes “potential” if two or more flights can reach at the *same* time the spatial conflict. CSAs consist in finding minimal changes of the wind-optimal trajectories to avoid all the potential conflicts. Given the fixed architecture and the limited amount of physical resources of the D-Wave 2XTM quantum chip, it is impossible to directly CSAs problems on on the quantum hardware. For this reason, a series of transformation of the original CSA formulation must be performed before to use the quantum chip as optimizer. Figure 1 summarizes the transformation we used in this project:

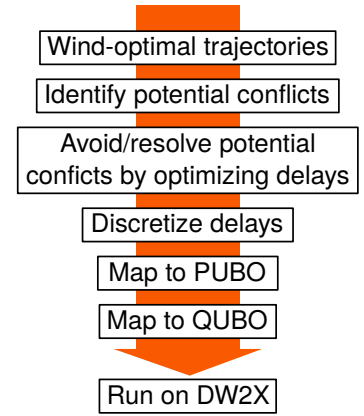


Figure 1: Flow diagram to map ATM problems onto the D-Wave 2XTM quantum chip.

- **Identify potential conflict.** In principle, any conflict is a potential conflict: indeed, flights can accumulate delay and therefore arrive at the same to a spatial conflict. Nevertheless, the number of spatial conflicts is prohibitive. To limit the number of potential conflicts, we pre-processed the wind-optimal trajectories assuming a *time* delay of at most 1 hour² and eliminating all the potential conflict that do not respect the bound. This process reduce the number of potential conflicts from XXX to XXX³.
- **Avoid/resolve potential conflicts by optimizing delays.** Even after the reduction of potential conflicts, it would be impossible to directly map the (3+1)-dimensional trajectories onto the D-Wave 2XTM quantum chip. Instead, we formulate the CSAs in terms of a “delay” problem. More precisely, let us define the time T_{ik} of a flight i to reach a potential conflict k , that is:

$$T_{ik}(d_i, \{d_{ip}\}) = t_{ik} + d_i + \sum_{p \in \mathcal{P}_i^{<k}} d_{ip}, \quad (1)$$

with t_{ik} the wind-optimal time to reach the conflict k , d_i the delay at the departure, $\mathcal{P}_i^{<k}$ the set of potential conflicts p that antecede the conflict k and d_{ip} the delay of flight i at the potential conflict p . In the above formulation, it is assumed that all the maneuvers to avoid potential conflicts consist in small and local changes of the wind-optimal trajectories that do not create a cascade of new potential conflicts and do not change the location of the existing potential conflict. The cost function

¹What is the number of trajectories?

²Check this

³Please add some numbers.

Task/Milestone	Performance Metric	Expected completion
Map the ATM problem to a suitable polynomial binary optimization problem (PUBO).	Number of required qubits. Largest degree of the polynomial. Connectivity of the underlying coupling graph.	✓
Map the ATM problem to a suitable quadratic binary optimization problem (QUBO).	Number of required qubits. Connectivity of the underlying coupling graph.	✓
Identify a set of benchmark ATM problems.	Hardness as a function of size.	1 month
Analyze the results/performance of classical QUBO solvers on ATM problems.	Assess the quality/variety of the solutions after discretization. Find the bottom line for classical computation.	1.5 month
Find an embedding to map the ATM problem onto the Chimera architecture.	Overhead of the embedding. Largest ATM problem embeddable on the current D-Wave 2X™ chip.	2 month
Compile the ATM benchmark ensemble for the D-Wave 2X™ chip.	Scaling of expected time to solution vs. size compared to classical code. Variety of different acceptable solutions.	3 month
Outlook on different architectures and annealing strategies, hardware changes	Potential quantum enhancement.	4 month

Table 1: Breakdown of the project effort into milestones, including suggested performance metric and completion dates (check marks indicate completed tasks).

to minimize can be then expressed as

$$f(\{d_i\}, \{d_{ip}\}) = \sum_i d_i + \sum_{ip} d_{ip}, \quad (2)$$

where all the $\{d_i\}$ and $\{d_{ip}\}$ are constrained so that two flights i (j) have a different arrival time T_{ik} (T_{jk}) to the same potential conflict k , namely

$$\Theta_{a_{ijk}}(|T_{ik} - T_{jk}|) = 0, \quad (3)$$

with Θ a certain function which ensure that the two flights are far apart to complete their maneuvers. Here, a_{ijk} represent some extra-parameters that can be added to the model to have more control on the maneuvers.

• **Discretize delays.** The delay model is completely defined by the Equations (1)-(3). The successive step consists in the discretization of the variables involved in the delay model. The discretization is achieved by expressing each variable x in terms of a discrete set of values $\{x_\alpha\}$

$$x = \sum_{\alpha} x_{\alpha} \sigma_{\alpha}, \quad (4)$$

with $\sigma_{\alpha} = \{0, 1\}$, and then enforcing that only one among all the $\{\sigma_{\alpha}\}$ is one, namely

$$\sum_{\alpha} \sigma_{\alpha} = 1. \quad (5)$$

• **Map to PUBO.** Using the discretization in Equation (4), with the constraint given by Equation (5), the cost function f in Equation (2) of the delay model can be therefore expressed as a polynomial unconstrained binary optimization (PUBO).

• **Map to QUBO.** The D-Wave 2X™ quantum chip can only natively solve quadratic unconstrained binary optimization (QUBO) problems. Therefore, a reduction from a PUBO, which involves higher order polynomials, to QUBO is necessary. The reduction can be done using two different approaches. On one hand, it is possible to use gadgets to reduce higher order polynomials to quadratic terms [3]. On the other hand, one can use Lagrange multipliers [4] to enforce constraints as in Equation (5) and then reduce the number of higher order polynomials.

Assessment and conclusions

We have devised the delay model to reduce ATM problems to QUBO problems, which are natively solved by the D-Wave 2X™ quantum chip. Despite its simplicity, the model can still capture many of the properties of ATM problems. Next step of the project will be to analyze the quality/variety of the solutions of the delay model using state-of-art classical QUBO optimizers. This will give us a bottom line for the performance of quantum annealers. Finally, we will run the ATM problems on the current D-Wave 2X™ quantum chip and compare the quality of results with the classical counterpart. Given the limited amount of resources for the D-Wave 2X™ quantum chip, we will benchmark the quantum annealer using small problems. The potential quantum enhancement will be then extrapolate from the data. See Table 1 for a complete overview of tasks/milestones.

References

- [1] Eleanor G Rieffel, Davide Venturelli, Bryan O’Gorman, Minh B Do, Elicia M Prystay, and Vadim N Smelyanskiy. A case study in programming a quantum annealer for hard operational planning problems. *Quantum Information Processing*, 14(1):1–36, 2015.
- [2] Davide Venturelli, Dominic JJ Marchand, and Galo Rojo. Quantum annealing implementation of job-shop scheduling. *arXiv preprint arXiv:1506.08479*, 2015.
- [3] Ryan Babbush, Bryan O’Gorman, and Alán Aspuru-Guzik. Resource efficient gadgets for compiling adiabatic quantum optimization problems. *Annalen der Physik*, 525(10-11):877–888, 2013.
- [4] Pooya Ronagh, Brad Woods, and Ehsan Iranmanesh. Solving constrained quadratic binary problems via quantum adiabatic evolution. *arXiv preprint arXiv:1509.05001*, 2015.