Formalizing the ATM problem

Bryan O'Gorman, Tobias Stollenwerk

April 22, 2016

These notes attempt to formalize the ATM problem: given a set of flights, schedule them in a way that minimizes cost.

1 Configuration space

At the highest level, the configuration space is a set of 4D trajectories, one for each flight. Even with relatively coarsely discretized spacetime, that space is infeasably large. Therefore we parameterize each trajectory by its deviations from the ideal, conflict-ignorant one. Such deviations are of two types: temporal and spatial. A temporal deviation is simply a delayed start (typically on the order of up to 30 minutes). We assume that within the time-scale of such delays, the ideal conflict-ignorant trajectory is time-independent. There are two types of spatial deviations: global and local. A global spatial deviation changes the entire trajectory. For example, one parameterization of global spatial deviations treats each trajectory as an arc in a plane perpendicular to the ground and considers smooth one-parameter curves to the projection of the trajectory onto the ground. A local deviation consists of a set of maneuvers that change relatively small parts of the spatial trajectory. Such maneuvers may introduce a delay, but, as with the delays due to temporal deviations, we assume that the subsequent trajectory is unaffected other than simply being delayed.

Henceforth, we focus on origination delays (temporal deviations) and maneuvers (local spatial deviations).

2 Cost function

In reality, the cost function results from factors such as fuel, labor, time, etc.; here we treat it as a unitless quantity that appropriately weights these factors. Because we are interested in the relative costs of different configurations, we treat the cost of the configuration in which all flights are routed along their independently optimal trajectories, ignoring conflicts, and are exactly on time, as zero. We further assume that the total cost is the sum of two different types of costs:

- for each flight i, the cost c_i^{late} of being late on arrival,
- for every conflict k, the cost c_k^{avoid} of avoiding it.

The total cost therefore is

$$c = \sum_i c_i^{\text{late}} + \sum_k c_k^{\text{avoid}},$$

where each contribution to the cost function depends (implicitly as written so far) on the configuration space. Because the lateness and avoidance costs are associated with flights and conflicts, respectively, we write simply $c_i = c_i^{\text{late}}$ and $c_k = c_k^{\text{avoid}}$ where the distinction is clear from context. Henceforth, we simplify things by ignoring the direct cost of the local maneuvers. However, they will still indirectly contribute to the cost function via their introduction of further delays. The quantity to minimized can then be written

$$c = \sum_i c_i(D_i),$$

where $D_i = d_i + \sum_k d_{ik}$ is the delay of flight i at its destination, which will be a function of its origination delay d_i and delays $\{d_{ik}\}$ introduced by local maneuvers. For notational simplicity we define d_{ik} for every flight i and every conflict k, but many of them will be trivially zero (i.e. when flight i is not involved or potentially involved in conflict k).

Lastly, we simplify things further by assuming that all destination delays are equally important, (which comes by necessity from the absence of information to the contrary), so that the quantity to minimize is simply the total delay of all flights:[1]

$$c = D = \sum_i D_i.$$

3 Classical subroutines

We further assume the availability of the following two subroutines, whose resource requirements we regard as negligible:

- given a flight (its origin and destination, and properties of the plane), return the optimal route
- given two flight paths, return the set of conflicts
- given two flight paths and start times, and one of the conflicts from above, return the "delay frontier" in the sparse case (described below) or the set of possible conflict-avoiding maneuvers in general

4 Assumptions made

- For each flight, the optimal route is time-independent (both spatially and in duration) over the timescale of possible delays. For example, if the flight has no potential collisions, its lateness on arrival will exactly equal its departure delay.
- Actively avoiding a collision (rather than avoiding via departure delays) does not change the cost of actively avoiding other collions on the same flight path. In reality, one could reason, e.g. that the little bit of fuel used in each active collision avoidance could add up to a disastrously empty tank, but in general we would like the effect of active collision avoidances to be as litle as possible. Similarly, one could argue that the plane burns fuel while idling.

5 Potentially helpful assumptions

- All conflicts are pairwise. That is, when two planes have a potential conflict, there are no other planes nearby. Furthermore, when this conflict is avoided, it does not create a conflict with another flight. (This may be the most unreasonable assumption, e.g. near major airports or on crowded jetstreams.)
- Collisions are symmetric. That is, the cost of avoiding them is a function only of the magnitude of the difference in delays of the two flights up until that point. For example, the cost of avoiding a collision between i and j is the same whether i is delayed by d or j is.
- The lateness cost functions are non-decreasing; we might as well allow it to be negative for earliness, though that could be compiled away by pushing the range of available departure times earlier. This is obviously satisfied for the linear cost functions with unit slope used here, and hard to imagine relaxing.

[1]The lack of information with respect to the relative importance of flights in reality only requires that all c_i be the same function. One could imagine minimizing the L_2 norm(which would help avoid having a few very late flights) rather than the L_1 norm as we do here, or indeed any other function. The simple sum here is the one currently used by Olga et al., and has the advantage of being most conducive to mapping to QUBO.

6 Towards a solution[2]

Let $D_{ik} = d_i + \sum_{k'<k} d_{ik}$ be the delay of flight i when it reaches potential conflict k, and $\mathbf{a}_k$ be some parameterization of the maneuvers chosen for conflict k. Then the general cost function is, with all dependencies explicit,

$$D(\mathbf{d}, \mathbf{a}) = \sum_i D_i(\mathbf{d}, \mathbf{a}) = \sum_i \left(d_i + \sum_k d_{ik}(\mathbf{D}_k, \mathbf{a}_k) \right),$$

where $\mathbf{d} = (d_i)_{i=1}^n$ and $\mathbf{a} = (a_k)_{k=1}^m$ represent all the origination delays and maneuver parameters, respectively; $\mathbf{D}_k = (D_{ik})_{i \in I_k}$, where I_k is the set of flights involved or potentially involved in conflict k; and $\mathbf{a}_k = (a_{k'})_{k'=1}^{k-1}$. This is the most general cost function we can write down using local maneuvers and the assumptions made in previous work by Olga et al.

6.1 Sparse case

First, we describe a method for solving the "sparse" version of the problem. By sparsity, we mean that all conflicts are between only two trajectories and that no other trajectories are nearby in spacetime, so that potential maneuvers for avoiding a conflict between two flights do not introduce additional conflicts with other flights.[3]

The delays (d_{ik}, d_{jk}) introduced by maneuvering to avoid conflict k between flights i and j are solely a function of the delays (D_{ik}, D_{jk}) of those flights up until the conflict and the maneuver a_k chosen to avoid the conflict. In particular, the delay depends only on the difference of the delays $\Delta_k = D_{ik} - D_{jk}$ (where the i and j for a given k are implictly specified and have a canonical ordering). It's plausible to assume that it furthermore depends only on the magnitude of this difference $|D_{ik} - D_{jk}|$, though it's not yet clear that that will help.

In general, the avoidance delays as a function of this difference Δ_k will be a function of the specific maneuvers chosen. The most general cost function in the sparse case is

$$D(\mathbf{d}, \mathbf{a}) = \sum_i \left(d_i + \sum_k d_{ik}(D_{ik} - D_{jk}, \mathbf{a}_k) \right).$$

Three special cases are as follows.

No delay In the (overly simplistic) case where maneuvers do not introduce any further delays, we have simply

$$D = \sum_i \left(d_i + \sum_k d_{ik}(d_i - d_j) \right),$$

where

$$d_{ik} = \begin{cases} 0, & \text{collision } k \text{ can be avoided given delays } d_i \text{ and } d_j, \\ \infty, & \text{otherwise.} \end{cases}$$

For the assumption that maneuvers do not introduce further delays to make any sense at all, we must bound their size, which is why some conflicts simply cannot be avoided within this approximation. Of course, we define d_{ik} as piece-wise infinite for consistency with the rest of these notes, but in practice this would be implemented as a hard constraint. (Interestingly, this notational device has the morose interpretation that if the flights do collide, the further delay introduced is indeed infinite.)

Independent delays There are no spatial maneuvers, but each flight i is allowed to slow down by some delay d_{ik} before conflict k (and after the its previous one). In the above formalism, this is equivalent to $\mathbf{a}_k = (d_{ik}, d_{jk})$. In this case, there is no need for $(d_i)_{i=1}^n$ because for each flight the origination delay d_i can just be subsumed into the delay d_{ik} of the first conflict. Each potential conflict k is an actual conflict when the difference between the total delays of the constituent flights are in some range $\left[\Delta_k^{\min}, \Delta_k^{\max}\right]$.

[2]It remains to be proven that this problem is hard, either in general or for realistic values of its parameters, e.g. cost functions.

[3]This could be extended to an m-sparse version, where each conflict can affect $m-1$ others.

Delay frontier For a given Δ_k that there is a minimal delay of each flight d^*_{ik} assuming that the other is unchanged, and there is (presumably) a concave "delay frontier" in the region $[0, d^*_{ik}] \times [0, d^*_{jk}]$ connecting $(d^*_{ik}, 0)$ and $(0, d^*_{jk})$ that gives the possible delays (d_{ik}, d_{jk}). We parameterize this curve by the scalar $\mathbf{a}_k = a_k \in [0, 1]$. Taking the various values of Δ_k into consideration, there is a "delay surface". Discretizing a_k into only two values $\{0, 1\}$ is equivalent to requiring that exactly one flight maneuvers around the other, whose trajectory remains unchanged. More generally, we could allow any combination of d_{ik} and d_{jk} not strictly within the delay frontier, combined with some upper bound to define an allowable delay volume.

All of the above forms invite straightforward QUBO formulations. Discretize and introduce bits for each discrete value for the variables d_i, d_{ik}, $\Delta_k = D_{ik} - D_{jk}$, and a_k:

$$C = D + xC^{\text{one}} + yC^{\text{equal}},$$

where

$$D = \sum_i \sum_\alpha \alpha d_{i\alpha} + \sum_{ik} \sum_\beta \beta d_{ik\beta}$$

is the delay to be minimized,

$$C^{\text{one}} = \sum_i \left(\sum_\alpha d_{i\alpha} - 1 \right)^2 + \sum_{ik} \left(\sum_\beta d_{ik\beta} - 1 \right)^2 + \sum_k \left(\sum_\gamma \Delta_{k\gamma} - 1 \right)^2 + \sum_k \left(\sum_\delta a_{k\delta} - 1 \right)^2$$

is the penalty function ensuring that each variable has a well defined value encoded by a single bit,

$$C^{\text{equal}} = \sum_k \left[\left(\sum_\alpha d_{i\alpha} + \sum_{k'<k} \sum_\beta \beta d_{ik\beta} \right) - \left(\sum_\alpha d_{j\alpha} + \sum_{k'<k} \sum_\beta \beta d_{jk\beta} \right) - \left(\sum_\gamma \gamma \Delta_{k\gamma} \right) \right]^2 + \sum_{ik} \left[\left(\sum_\beta \beta d_{ik\beta} \right) - \left(\sum_{\gamma\delta} d_{ik}(\gamma, \delta) \Delta_{k\gamma} a_{k\delta} \right) \right]^2$$

is the penalty function ensuring that $\Delta_k = D_{ik} - D_{jk}$ and $d_{ik} = d_{ik}(\Delta_k, a_k)$, the Greek indices represent the discrete values that the corresponding quantities can assume, and x and y are penalty weights.[4]

We could make a further approximation and for each conflict pick some optimal-in-expectation a_k, thereby removing that variable from the formulation. Alternatively, we could allow the delays (d_{ik}, d_{jk}) to take on any value at or beyond the delay frontier (i.e. somewhere in between the independent delays and delay frontier cases). We could also suitably parameterize the "delay surface" so that the cost function could be written in a functional form.

For the specific case in which we allow independent delays (d_{ik}, d_{jk}) subject to their difference being outside a certain window $[\Delta^{\min}_k, \Delta^{\max}_k]$, we have some can write a more efficient QUBO in several ways. In the simplest formulation, we need only bits $(D_{ik\alpha})_{ik\alpha}$ defining the values of the variables (D_{ik}). Let $d^{\max}_i$, $d^{\max}_{ik}$, and $D^{\max}_{ik}$ be the maximum values, respectively, that d_i, d_{ik}, and D_{ik} are allowed to take on. The value of $d^{\max}_i$ and $d^{\max}_{ik}$ are part of the instance specification (dictated by realistic constraints on how much certain flights can be delayed at origination and conflicts). With introducing any further approximations, the value of $D^{\max}_{ik}$ then is simply $d^{\max}_i + \sum_{k'<k} d^{\max}_{ik}$. We introduce the penalty function

$$C^{\text{one}}(\mathbf{D}) = \sum_{ik} \left(\sum_\alpha D_{ik\alpha} - 1 \right)^2$$

to ensure that each D_{ik} has a well-defined value. There are two further types of constraints. First, that subsequent delays $(D_{ik}, D_{i\partial_{ik}})$ are consistent with a plausible value of $d_{ik} \in [0, d^{\max}_{ik}]$, where ∂_{ik} is the next potential conflict that flight i is involved in after potential conflict k. We enforce this via the penalty function

$$C^{\text{trajectory}}(\mathbf{D}_i) = \sum_i \sum_{k \in K_i} \sum_{(\alpha-\beta) \notin [0, d^{\max}_{ik}]} D_{ik\alpha} D_{i\partial_{ik}\beta}$$

[4]In practice we could have varying penalty weights for the various penalties.

6.2 Dense case

The sparsity condition assumed above is in general false, even when ignoring near-airport trajectories. The other extreme is the "dense" case, in which all flights are restricted to a "highway" as in Olga's thesis. An appropriate QUBO for that problem is relatively straightforward, and can be written explicitly if needed.

6.3 General case

General instances fall into neither of the above extremes. One potential solution is to use some sort of master-slave subproblem decomposition to combine solvers for each extremal case into a general solution.