CIVIL AVIATION AUTHORITY

Presented by Dr David Timlin CAA Holborn

Problem

Stated briefly the problem presented by CAA was to predict or model aspects of the delay to flights from UK airports caused by Air Traffic Control restrictions.

Important points to note about the cause of such delays are as follows:

- Although all UK flights require a flight plan to be filed at least four hours in advance of departure, flights on routes subject to flow restrictions, in addition, need specific take-off slots to be granted, by flow regulators. Airlines must bid for these slots, but not earlier than three hours before requested departure.
- in theory, airlines are given a take-off slot in the order in which the requests are made;
- there are constraints on the number of flights leaving an airport each hour;
- there are constraints on the number of flights passing through key geographical points, called 'way points'.

We have throughout assumed that each airline attempts to put in its request for a take-off slot at exactly three hours before it is required since this maximises the chance of getting the slot. However, the order that the requests are displayed to the regulator may differ from the flight plan order. Such perturbations may have many causes, for instance

- different airlines entering requests for identical departure times
- single airlines entering requests for identical departure times (at different airports).

Let us give some examples of how small changes in ordering may affect delay. The examples are intentionally simplified but are illustrative of the type of situation which arises. In Figure 1 we see the flight route Heathrow (LHR) to Z. The point annotated 8/60 is a way point which can accept a maximum smoothed traffic of 8 flights per 60 minutes. If less than eight flights are requested per hour then there are no delays.

On the other hand, suppose that we have the more complicated situation shown in Figure 2. Depending on the order in which requests for take-off slots come, there may or may not be differing delays. This can be seen by comparing the two following situations in which one plane wants to go from GLA to X and another from GLA to Y. Suppose that there have already been three flights to X and 11 flights to Y already requested.

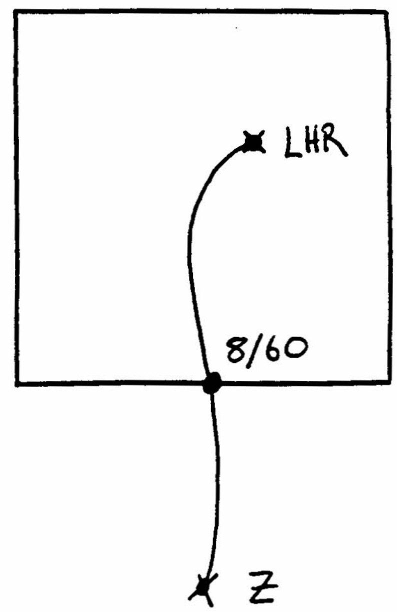

Figure 1: No interaction.

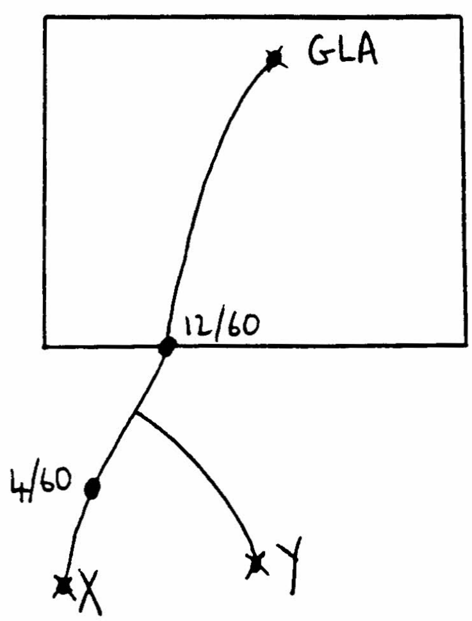

Figure 2: Interaction.

1. If the request for the flight to X comes in before that to Y there will be no delay to X but a five minute delay to Y.

2. If the request for Y comes before X then there will be no delay for Y but a 15 minute delay for X.

From this example it is clear that there is great scope for minimising total delays (perhaps suitably weighted) but this is not what the regulators are supposed to do. The true situation is more complicated in that different regulators are in competition for slots at common domestic way points.

The CAA have a computer program that takes as input the flight plan times and ordering and outputs delays:

$$\text{FLIGHT PLAN ORDER} \quad \rightarrow \quad \text{DELAY}$$

Unfortunately, the output is in poor agreement with reality.

We can represent the problem schematically as

FLIGHT PLAN ORDER	→	ACTUAL ORDER (various perturbations)	→	TAKE-OFF TIMES	→	DELAY

An interesting feature of the CAA computer program is that there are no significant trends in the error between predicted and actual delays. It does not consistently over- or under-estimate the delays.

Some of the CAA questions

1. Is it possible to produce realistic delays with this kind of model?

   It is unlikely that a totally deterministic model will predict the delays accurately.

2. What kind of modelling strategy is needed?

   Since real-world delays and alterations to the theoretical ordering are unpredictable and difficult to model it is probably best to seek a probabilistic model.

3. What level of agreement between modelling and actual delays is it feasible to aim for?

   If we accept a probabilistic model then we expect, with a sufficiently large sample, to predict the mean and the standard deviation of delays.

Simple paradigm models show that small scale reordering can easily lead to large scale changes in delays. It is important that any probabilistic model is based on changes to orderings of requests for take-off slots rather than the times at which these requests are made: the probability of a reordering of two flights is insensitive to the time between the requests, at least if these times are not large.

One strategy of obvious appeal is to use 'weather forecasting tactics', that is, to make small random alterations to the order of flight requests, then use the current computer code to calculate estimated delays. This will lead to a distribution of predicted delays. It will then be possible estimate the amount of random ordering needed to get theoretical results close to the actual results. Such an amount of reordering may or may not be realistic but will give useful information about why the current computer code is inaccurate. This idea is illustrated by the following simulation.

A simple simulation

To investigate how flight departure delays can develop, it is illustrative to consider a simple case. Suppose, for example, that a sequence of flights are designated to leave London (L) and fly either through one constrained way point P, or through two constrained way points, P then R. Let the total number of flights be $n = n_P + n_R$, with n_P flights through P only, and n_R through P and R. Flights are designated to leave at equal time intervals. For simplicity, we will assume that the time between successive flights, the journey time from L to P and the journey time from P to R are all integer multiples of some unit of time. Furthermore, we adopt the times as follows:

$$\begin{aligned} \text{time between designated departures} &= T, \\ \text{journey time from L to P} &= 1, \\ \text{journey time from P to R} &= K. \end{aligned}$$

The choice of T and K is discussed later.

We suppose that initially airspace is empty. Each flight takes off at its designated departure time, or as soon as possible afterwards, according to the following rules:

- Only one aircraft can occupy each of the LP and PR routes during a single time interval.
- In general, a 'first-come, first-served' system applies. When a flight is designated to depart, time corresponding to the soonest possible journey is reserved on the LP and PR routes. Reservations cannot be overwritten (even for the sake of global optimisation).
- A flight through R cannot pause at P, and so cannot leave London if the PR route is not free when necessary, i.e. 1 time unit after departure.

One feature of this system is that a limited form of 'queue-jumping' can occur. Specifically, a flight through P only can depart as soon as a time interval of 1 unit is available on the LP route — even though several future bookings for this route may already exist. Following this system, we may compute the departure time of each flight, and the delay d_i of the ith flight: d_i = (actual departure time) − (designated departure time).

As an example of the way we calculate delays, consider the following trivial example. Here $n = 4, n_P = 2, n_R = 2, K = 4, T = 3$, and we take a sequence of flights $\{P, R, R, P\}$. The table below shows the reservations made for each flight on the two routes, and any delays.

time interval	designated departure flight no.	destination	delay	L-P	P-R
1	1	P	0	#1	
2					
3					
4	2	R	0	#2	
5					#2
6					#2
7	3	R	1		#2
8				#3	#2
9					#3
10	4	P	0	#4	#3
11					#3
12					#3
13					
		total delay = 1			

The choice of the time interval T is obviously important. If T is large, then delays may be few and far between; if T is very small, then significant delays may be inevitable. The minimum value of T where a sequence can exist that gives rise to <u>no</u> delays is $T = (n_r/n)K$. Since with this value of T delays may generally be large, T is taken to be the smallest integer $\leq 2(n_r/n)K$.

Given a sequence of flights, we may thus compute the average delay $\bar{d} = \sum_{i=1}^{n} d_i \,/n$. Suppose that the original sequence is now perturbed by successively swapping adjacent pairs with probability p. To investigate the effect of such a perturbation on $\bar{d}$, a short FORTRAN program was written. The program produces an original sequence, calculates the (average) delay, and then repeatedly perturbs the original sequence as above, and calculates the new delay in each case. Random perturbations were simulated by using a pseudo-random number created by a NAG routine.

Figure 3 below shows computed distributions of average delays for the case $n = 1000$, $n_P = 750$, $n_R = 250$, $K = 4$, $T = 2$, with the number of perturbations $m = 1000$ and p taking values 0.03 and 0.01. The same original sequence was used, and the average delay for this unperturbed sequence is 0.237 units. Both distributions clearly exhibit multiple local peaks.

p = 0.01

```
Each dot represents 15 points
                             .
                             :
                             :
                          .  :  .
                          :  :  :
                          :  :  :  .
                       :  : .:. :. :  .
      .    .... ....:.::.::::::.:::::.:......... ..  .. .
  +---------+---------+---------+---------+---------+---------+-------
0.210     0.220     0.230     0.240     0.250     0.260
```

p=0.03

```
Each dot represents 5 points
                       .
                      ::.
                  .   ::: .
                  : ::::: :
                : ::::::::: :
                : ::::::::: :::
            . : :::::::::::::::.:
            : :.:::::::::::::::::
  . .... .:.:::::::::::::::::::::::.:.:. .. .. . .   .
---+---------+---------+---------+---------+---------+---------+---
 0.210     0.225     0.240     0.255     0.270     0.285
```

Figure 3: Distribution of average delays

Let's examine a sequence of P's and R's to discover the patterns that gives rise to delays. To do that it is useful to look at clumps. A clump is a subsequence of maximal length with an equal number of R and P's beginning with two R's. For example, in the following sequence the underlined parts are the clumps:

PPRPRPRRPRPRPPRPPPRPPPRRPPPRRPRPRPPP .

When perturbing a sequence of R and P's as above, it is most important what happens to the P's immediately in front of a clump. Consider a clump C, with n_C elements in it. Note that a clump is preceded by prefixes of type RP or PP. If the prefix of C is of type RP, and the P in front of C is swapped forward (with R), then depending on the distribution of P's in C a little consideration shows that the total delay increases with between $n_C - n_C/2$ and $n_C + n_C/2$ units. If on the other hand the P is swapped backwards (possible in both cases) then the total delay decreases by between $(n_C - 2) - (n_C/2 - 1)$ and $(n_C - 2) + (n_C/2 - 1)$ units. All other swaps, inside or outside clumps, change the total delay by either 0, 1 or 3 units.

The distribution of average delays when perturbing a sequence depends crucially on the size and number of clumps in the original sequence. Because of the large effect described above that a swap in front of a clump has on the average delay, one expects a rather large variance in the distribution. This is also observed on the shown figures with a multiple number of peaks. We conclude that the size of average delay is rather unpredictable when there is a random element in the perturbation method.

CJA,UTC,JND,ADF,HH,ELT,AEW,PW,SW