Minimizing Flight Delay

Tanujit Dey • David Phillips • Patrick Steele*

The College of William & Mary

Introduction

Southwest Airlines 1987–2008

1987

1997

2002

2008

Motivations

• Over time, flight networks have grown in size and complexity, delays on flight legs have similarly grown.
• How can individuals and airlines make better decisions regarding flight travel?

GOAL: Design a visual decision support tool that can find a flight plan with smallest predicted delay.

Data

• Airlines: American, American Eagle, Continental, Delta, Skywest, Southwest, United
• Variables: Year, Month, DayofMonth, DayOfWeek, DepTime, ArrTime, UniqueCarrier, ArrDelay, DepDelay, Origin, Dest, Distance, TaxiIn, TaxiOut, CarrierDelay, WeatherDelay, NASDelay, SecurityDelay, LateAircraftDelay
• Omitted data: All cancelled flights

Stochastic IP

We use techniques from integer programming and stochastic optimization. A (linear) integer program (IP) is an optimization problem with form

$$\min \left\{ \sum_{j=1}^{J} C_j \gamma_j : \forall i, \sum_{j=1}^{J} a_{ij} \gamma_j = b_i, \gamma_j \in \{0, 1\} \right\},$$

where $C_j, a_{ij} \in \mathbb{R}$ are given and $\gamma_j$ represent yes/no decisions.

• Examples include finding the minimum cost assignment of airplanes to flights, routing service delivery vehicles and scheduling sports teams.
• A stochastic IP (SIP) has some or all of $C_j$ and $a_{ij}$ random. E.g., if $C_j$ were random variables, then the IP would be a SIP.
• Every SIP has an associated deterministic IP where the random variables are replaced by non-random parameters.
• The solution and associated objective value of an SIP are random variables so solutions found are usually in expectation or probability.

Flight Graphs

• A graph for the airline $\mathcal{F}$ is $\mathcal{N}_F = (V, E)$ where $V$ are nodes representing airports and $E \subseteq V \times V$ are edges representing flight legs, i.e.,

$$V = \{i : i = \text{an airport from our data}\},$$

$$E = \{(i, j) : \exists \text{flight of } \mathcal{F} \text{ from } i \text{ to } j \text{ in 2005-2008}\}.$$ 

E.g., if there is a flight from LAX to IAD of 60 minutes, then $(\text{LAX, IAD}) \in E$. Edges are directed $(i, j) \neq (j, i)$.

• A path in $\mathcal{N}_F$ is an ordered set of edges,

$$P = ((i_1, j_1), (i_2, j_2), \ldots, (i_k, j_k))$$

so that $i_j \neq i_i$ for all $j \neq \ell$. We define $|P| = k$.

Cascading dependencies

Model

In order to predict delay, we performed a multiple linear regression on the response DelayLevel, a categorical variable defined as follows.

<table>
<thead>
<tr>
<th>DelayLevel</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay (minutes)</td>
<td>0 – 15</td>
<td>15 – 30</td>
<td>30 – 60</td>
<td>60 – 120</td>
<td>120+</td>
</tr>
</tbody>
</table>

Due to the volume of the yearly data sets, we randomly sampled (without replacement) 70% of the data to perform the multiple regression, and averaged the estimated coefficients of significant variables over 500 runs. These Bagged estimates were then used to predict DelayLevel which were linearly extrapolated by our sampling methods to predict delay for any origin-destination pair.

Sampling methods

Given an origin-destination pair, $s, t \in V$, max. flight legs, $k$, month, $m$, weekday, $w$ and airline, we must find distributions $C_{ij}$ for flight legs $(i, j)$ in one of the following sets. Let $P(s, t)$ denote a path from $s$ to $t$ and

$$E(s, t, k) = \{(i, j) \in E : (i, j) \in P(s, t), |P(s, t)| = k\}.$$ 

For $(i, j)$, let $S(i, j, m, w, \tau) = \{(i, j)_{(\delta, \tau)} : (i, j) \text{ has flight on date } d, \text{ time } t \geq \tau, \text{ month}=m, \text{ weekday}=w\}.$

• Sample in one of the following two ways:
  1. Naïve:
     - For each arc $(i, j) \in E(s, t, k)$, independently generate $(i, j)_{(\delta, \tau)} \sim \text{Unif}(S(i, j, m, w, 0))$
  2. Cascade:
     - Generate a specific date, $d \sim \text{Unif}(1/1/05, \ldots, 12/31/08: \text{month}=m, \text{weekday}=w)$.
     - (a) Set $\tau = 0$. For $d$, generate
       $$(s, j)_{(\delta, \tau)} \sim \text{Unif}(S(s, j, m, w, \delta), \delta = d),$$
       for every $j$ with $(s, j) \in E(s, t, k)$.
     - (b) For each $j$, set $\tau = t_j$ and repeat.

• For each arc $(i, j)$, apply estimated delay formula to sampled $(i, j)_{(\delta, \tau)}$ to obtain $\hat{\epsilon}_{ij}$.

Shortest paths with random distances (SPRD)

For a given origin, destination, month/weekday of travel and the maximum number of legs allowed, we solve the following SIP. For all $(i, j) \in E$, we define $\gamma_{ij}$ as indicators that $(i, j)$ are on the shortest path and $\gamma$ as the vector of $\gamma_{ij}$. We define the Shortest Paths Problem with Random Distances (SPRD) as the SIP, min

$$\min \left\{ \sum_{(i, j) \in E} C_{ij} \gamma_{ij} : \gamma \in \Omega \right\},$$

where $C_{ij}$ are random variables representing delay and $\Omega$ is the set of arc indicators corresponding to paths from the origin to the destination. Thus,

$$\Omega = \left\{ \gamma : \sum_{(i, j) \in E} \gamma_{ij} - \sum_{(i, j) \in E} \gamma_{ij} = b_i, \forall i \in V \right\},$$

where for all $i$ not equal to the origin or destination, $b_i = 0$. At the origin, $b_i = 1$ and at the destination, $b_i = -1$.

Our solution method

1. For each $(i, j) \in E$, estimate the distribution for $C_{ij}$
2. Repeat 500 times
   (a) Randomly generate a realization, $\bar{\epsilon}_{ij} \sim C_{ij}, \forall (i, j) \in E$.
   (b) Solve the (deterministic) shortest path problem.
   (c) Save the shortest path found.
Finding shortest paths

A deterministic problem

Given a set of predicted delay times \( C_{ij} \) on the arcs, and an origin-destination pair \((s,t)\), find the path from \( s \) to \( t \) of \( k \) flight legs or less with minimum delay.

Our algorithm

1. Find \( E(s,t,k) \) via Breadth First Search (BFS), finds \( s \)-reachable nodes with a FIFO queue.

   - BFS starting from IAD
   - Mark IAD found (yellow).
   - FIFO queue = [IAD].

   - Mark IAD done (orange).
   - Mark IAD found (yellow).
   - Set queue = [BOS, MIA].

2. Find the shortest path from \( s \in V \) via Dijkstra’s algorithm (requires \( C_{ij} \geq 0 \)).
   - Dijkstra’s algorithm uses \( d(i) = \text{estimate of shortest path from } s \text{ to } i \)
   - Relax(i): \( \forall (i,j) \in E \), if \( d(j) > d(i) + C_{ij} \), set \( d(j) = d(i) + C_{ij} \)
   - At each step, finds node \( p \) where \( d(p) = \min\{d(i) : i \text{ not relaxed} \} \), then calls Relax(\( p \)).

Conclusions

- Cascade sampling predicts cascade effects of delay better than Naive sampling.
- Cascade effects and delay patterns on any flight route within airlines, times/dates and airports can be visually compared.
- Runtimes are modest, with sampling as the computational bottleneck.
- Use \( \alpha_i C_{ij} + c((i,j)) \) for costs to find objectives such as total travel time, weighted delay with flight costs, etc.

Runtimes

<table>
<thead>
<tr>
<th>Algorithm step</th>
<th>For two flight legs and less</th>
<th>For three flight legs and less</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFS computations</td>
<td>( \approx 1 \text{ second} )</td>
<td>( \approx 1 \text{ second} )</td>
</tr>
<tr>
<td>Cascade sampling</td>
<td>( \approx 30 \text{ seconds} )</td>
<td>( \approx 400 \text{ seconds} )</td>
</tr>
<tr>
<td>Dijkstra’s algorithm</td>
<td>( \approx 1 \text{ second} )</td>
<td>( \approx 3 \text{ seconds} )</td>
</tr>
<tr>
<td>Total runtime</td>
<td>( \approx 32 \text{ seconds} )</td>
<td>( \approx 7 \text{ minutes} )</td>
</tr>
</tbody>
</table>