Combinatorial optimization and decision-focused learning for stochastic tail assignment

Léo Baty, Axel Parmentier

CERMICS, École des Ponts

January 23, 2025

École des Ponts ParisTech

The tail assignment problem

- **Instance** x:
  - available aircraft fleet
  - set of flight legs to operate
- **Decision** y: assign each leg to an aircraft
- **Objective**: minimize operational cost
- Operational **constraints** $\mathcal{Y}(x)$

$$\min_{\text{routes } y} \quad \text{operational cost} = \text{fuel cost} + \text{connection cost}$$

$$\text{subject to} \quad \begin{cases} \text{valid routes,} \\ \text{maintenance constraints,} \\ \text{mandatory connections.} \end{cases}$$

Example instance

Aircraft	Location	Fuel Factor
1	D	1.0
2	C	2.0
3	B	3.0

(a) Fleet

Flight Leg	Origin	Destination	Departure Time	Arrival Time
1	B	A	12:00	13:00
2	C	A	12:30	14:30
3	D	A	12:00	15:00
4	A	D	15:10	18:10
5	A	C	15:30	17:30
6	A	B	16:00	17:00

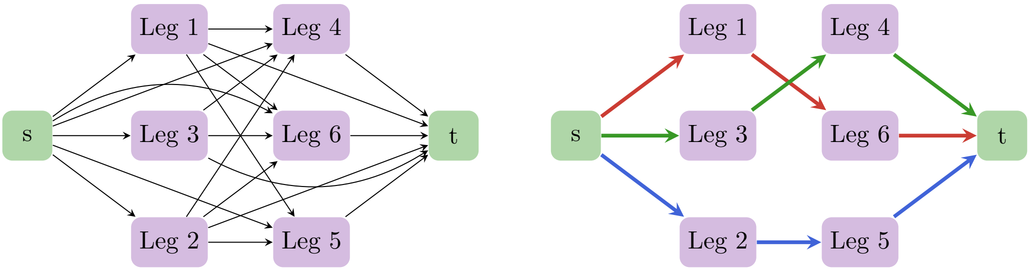
(b) Flight legs to operate

Deterministic tail assignment problem

Mixed-integer linear programming formulation:

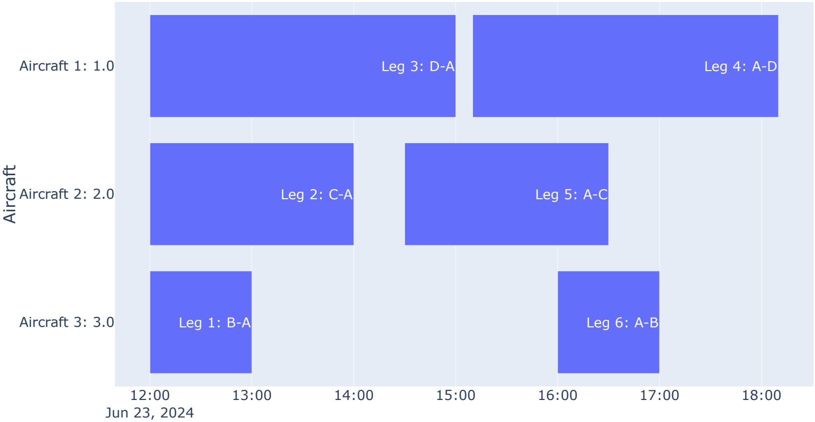
$$\min_{y \in \mathcal{Y}(x)} \theta^\top y$$

Connection graph and example solution

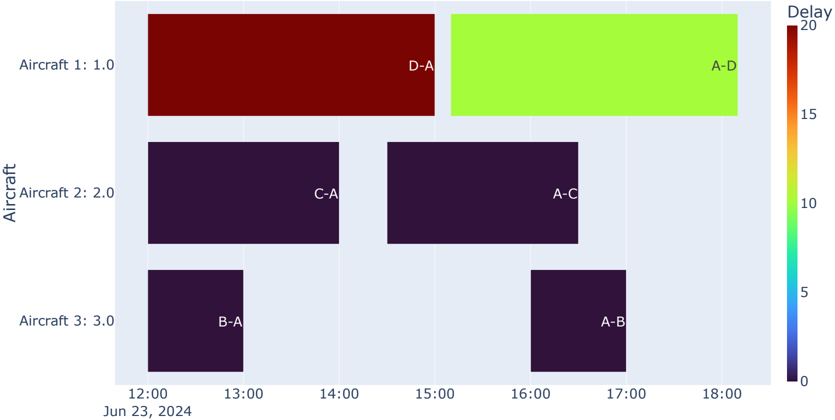


(a) Connection graph (b) Feasible solution

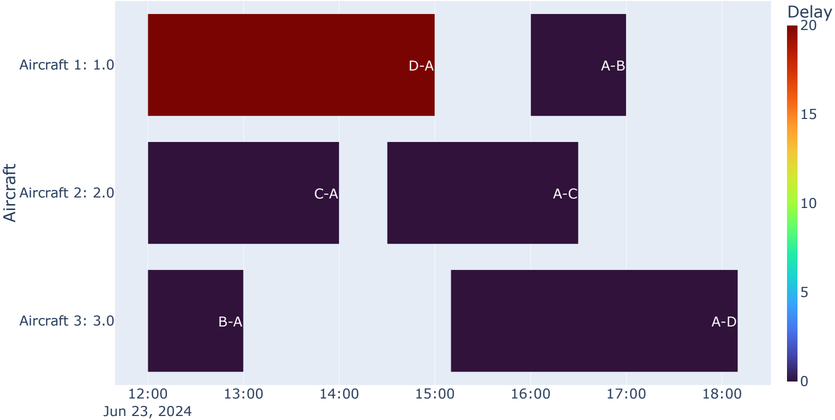
Connection graph and example solution



The **stochastic** tail assignment problem



The **stochastic** tail assignment problem

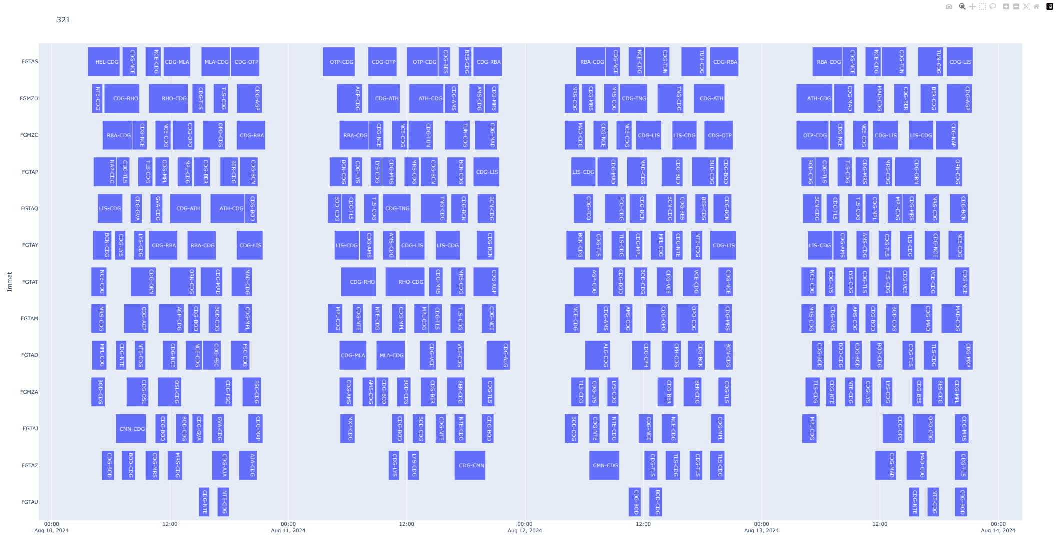


The **stochastic** tail assignment problem

We want to also have delay resilience:

$$\min_{\text{routes } y} \quad \text{operational cost} + \mathbb{E}_{\text{delays}}[\text{delays cost}]$$

$$\text{subject to} \quad \begin{cases} \text{valid routes,} \\ \text{maintenance constraints,} \\ \text{mandatory connections.} \end{cases}$$


321
Immat
FGTAS
FGMZD
FGMZC
FGTAP
FGTAQ
FGTAY
FGTAT
FGTAM
FGTAD
FGMZA
FGTAJ
FGTAZ
FGTAU
00:00
Aug 10, 2024
12:00
00:00
Aug 11, 2024
12:00
00:00
Aug 12, 2024
12:00
00:00
Aug 13, 2024
12:00
00:00
Aug 14, 2024

Delay propagation

Delay propagation equations:

$$\overbrace{\text{Arrival delay}}^{\xi_\ell^a} = \overbrace{\text{Departure delay}}^{\xi_\ell^d} + \overbrace{\text{Root arrival delay}}^{\varepsilon_\ell^a}$$

$$\underbrace{\text{Departure delay}}_{\xi_{\ell_j}^d} = \underbrace{\text{Propagated delay}}_{\max\left(\xi_{\ell_{j-1}}^a - \omega_{\ell_{j-1},\ell_j}, 0\right)} + \underbrace{\text{Root departure delay}}_{\varepsilon_{\ell_j}^d}$$

Root delay prediction model:

- Neural network: learn delay distribution from historical data
- Used to evaluate delay cost of solutions
- Used to generate scenarios for optimization

Sample average approximation

We can sample i.i.d. scenarios with our delay model:

$$\min_{y \in \mathcal{Y}(x)} \mathbb{E}_{\xi}[c^0(y; x, \xi)] \approx \min_{y \in \mathcal{Y}(x)} \frac{1}{S} \sum_{s=1}^{S} c^0(y; x, \xi_s)$$

Compact MIP

c^0 is non-linear:

- non-linear delay propagation
- piecewise linear delay cost

$\implies$ linearization leads to poor scaling with instance size and number of scenarios

$$
\begin{aligned}
\min_y \quad & \frac{1}{S}\sum_{s=1}^{S} c^0(y; x, \xi_s) \\
\text{s.t.} \quad & \sum_{a \in \delta^-(v)\cap\mathcal{A}^i} y_a^i = \sum_{a \in \delta^+(v)\cap\mathcal{A}^i} y_a^i, && \forall v \in \mathcal{V}, \forall i \in \mathcal{I}, \\
& \sum_{a \in \delta^+(s)\cap\mathcal{A}^i} y_a^i = 1, && \forall i \in \mathcal{I}, \\
& \sum_{a \in \delta^-(t)\cap\mathcal{A}^i} y_a^i = 1, && \forall i \in \mathcal{I}, \\
& \sum_{i\in\mathcal{I}} \sum_{a \in \delta^-(\ell)\cap\mathcal{A}^i} y_a^i = 1, && \forall \ell \in \mathcal{L}, \\
& y_a^i \in \{0,1\}, && \forall i \in \mathcal{I}, \forall a \in \mathcal{A}^i.
\end{aligned}
$$

Dantzig-Wolfe decomposition

- non-linearity is hidden in route costs c_r^i
- relaxation can be solved with column generation $\implies$ good quality lower bound
  - subproblem: constrained shortest path
  - can scale to large instances and more scenarios
- restricted master heuristic to generate integer solutions $\implies$ still does not scale well

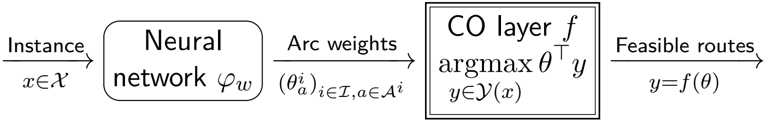
$$\begin{aligned}
\min_{y} \quad & \sum_{i \in \mathcal{I}} \sum_{r \in \mathcal{R}^i} c_r^i y_r^i \\
\text{s.t.} \quad & \sum_{i \in \mathcal{I}} \sum_{r \ni \ell, r \in \mathcal{R}^i} y_r^i = 1, && \forall \ell \in \mathcal{L}, \\
& \sum_{r \in \mathcal{R}^i} y_r^i \leq 1, && \forall i \in \mathcal{I}, \\
& y_r^i \in \{0, 1\}, && \forall i \in \mathcal{I}, \forall r \in \mathcal{R}^i.
\end{aligned}$$

1. Mathematical programming formulations

2. Decision-focused learning pipeline

3. Results

- Mathematical programming formulations have difficulty to scale on large instances
- They can be used to generate training data, $(x, \bar{y})$ pairs on small instances.
- We can then learn a parametrized policy as a decision-focused learning pipeline:



Learning problem: find w such that $\pi = f \circ \varphi_w$ is a good policy.

Learning algorithm 1: imitating integer solutions

Fenchel-Young loss over integer solutions:

$$\mathcal{L}_{\mathcal{Y}}^{\mathrm{FYL}}(\theta, \bar{y}) = \mathbb{E}_Z \left[\max_{y \in \mathcal{Y}(x)} (\theta + \varepsilon Z)^\top y \right] - \theta^\top \bar{y},$$

$\varepsilon > 0$, $Z \sim \mathcal{N}(0, \mathrm{I})$.

- $\mathcal{L}_{\mathcal{Y}}^{\mathrm{FYL}}(\theta, \bar{y})$ is convex and differentiable in θ.
- Subgradient:

$$\mathbb{E}_Z \left[\underset{y \in \mathcal{Y}(x)}{\mathrm{argmax}} (\theta + \varepsilon Z)^\top y \right] - \bar{y} \in \partial_\theta \mathcal{L}_{\mathcal{Y}}^{\mathrm{FYL}}(\theta, \bar{y})$$

This loss is not well-defined on relaxation solutions $y \in \tilde{\mathcal{Y}}(x)$.

Learning algorithm 2: imitating column generation relaxation solutions

Fenchel-Young loss over column generation relaxation solution:

$$\mathcal{L}^{\mathrm{FYL}}_{\tilde{\mathcal{Y}}}(\theta, \bar{y}) = \mathbb{E}_Z\left[\max_{y \in \tilde{\mathcal{Y}}(x)} (\theta + \varepsilon Z)^\top y\right] - \theta^\top \bar{y}.$$

1. Mathematical programming formulations
2. Decision-focused learning pipeline
3. Results

Experiment setup

Instances:

- Train/validation: small instances solvable with mathematical programming formulations.
- Test: larger instances

Features: distributional information about connection slacks

Solutions:

- $\mathcal{D}^{\tilde{\mathcal{Y}}}$: dataset with solutions of the column generation relaxation with 100 scenarios
- $\mathcal{D}_1^{\mathcal{Y}}$: dataset with integer solutions with 1 scenario
- $\mathcal{D}_{10}^{\mathcal{Y}}$: dataset with integer solutions with 10 scenarios

Performance on small instances

Training dataset	$\mathcal{D}^{\tilde{\mathcal{Y}}}$		$\mathcal{D}_1^{\mathcal{Y}}$		$\mathcal{D}_{10}^{\mathcal{Y}}$	
Evaluation dataset	Train	Val	Train	Val	Train	Val
Gap to lower bound	3.3%	4%	5.9%	4.6%	5.9%	3.5%
Total cost improvement	-0.7%	-0.8%	+0.8%	+0.04%	+0.6%	-0.02%

Performance scaling on large instances

Training dataset	$\mathcal{D}^{\tilde{\mathcal{Y}}}$	$\mathcal{D}_1^{\mathcal{Y}}$	$\mathcal{D}_{10}^{\mathcal{Y}}$
Gap to lower bound	4.1%	6.2%	6.5%
Total cost improvement	-27%	-25%	-25%

Thank you !