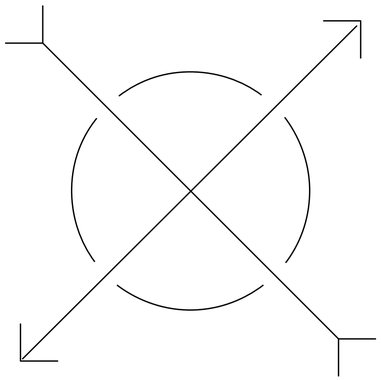


Optimization of Airport Surface Planning and Scheduling

Heron Yang - 2018 Spring Independent Study @ CMU SV

Problem Definition

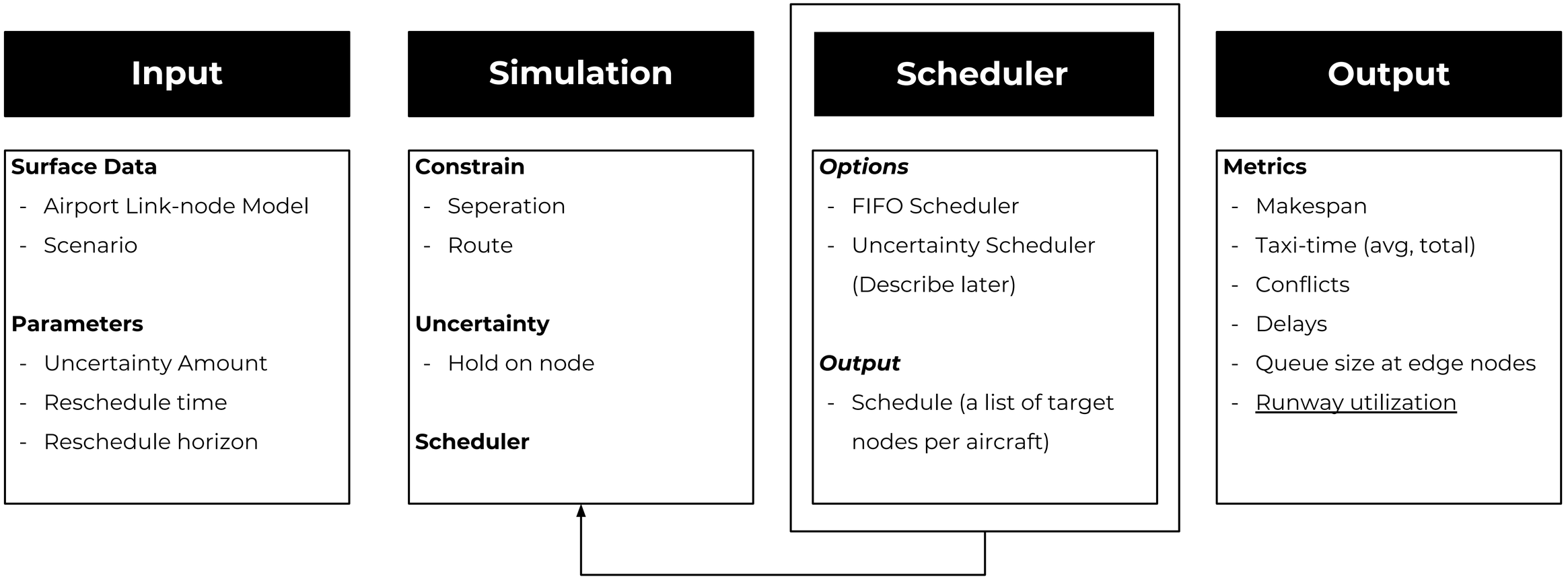
Can we develop a scheduler that gives less conflicts under real-world uncertainty?

Existing schedulers schedule based on a deterministic model and uncertainties are not put into concern. We argue that a more robust scheduler (less conflicts while there are uncertainties) can be implemented by scheduling on an non-deterministic model.

Can we reschedule less often?

In previous works, rescheduling happens within few seconds which is costly if it’s used by a real airport tower. Rescheduling at a high frequency is needed by that enables us to deal with uncertainties in real world. We argue that, by making the scheduler consider uncertainties while scheduling, we should be able to schedule less and obtain a similar performance.

Problem Domain



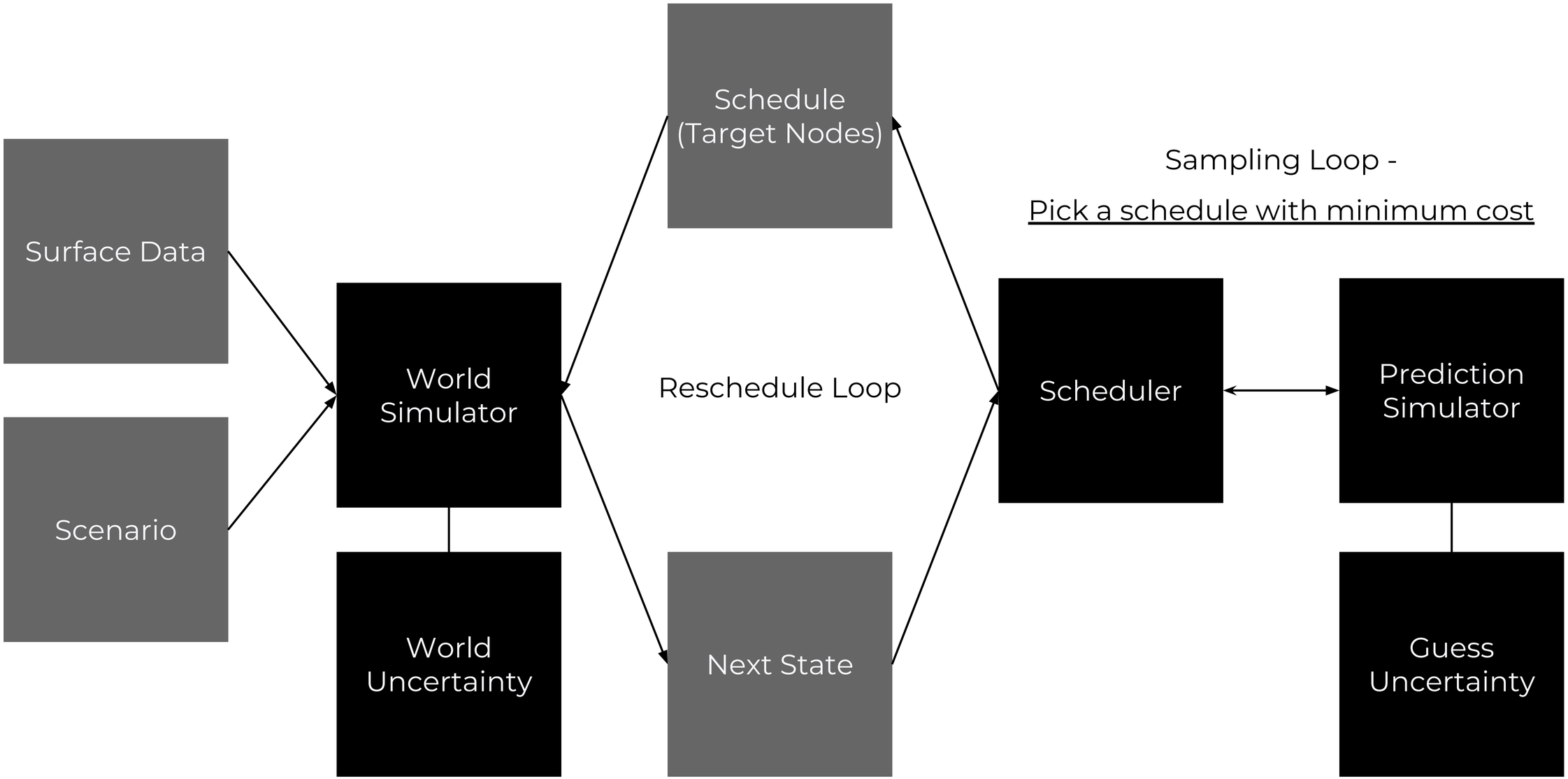
Cost function

$$
\begin{aligned}
Cost(s,u) &= w_0 \times makespan \\
&\quad + w_1 \times \ total\ taxi\ time \\
&\quad + w_2 \times expected\ number\ of\ conflicts \\
&= w_0 \times t_{end} + w_1 \times \Sigma_{a \in A(t_i)} t_D(a) + w_2 \times \frac{1}{n_u} \times \Sigma_j^{n_u} p_{u,cb_j} \times c_{s,cb_j}
\end{aligned}
$$

where:

a	= Identify aircraft.
A	= Set of all aircraft.
$A(t)$	= Set of all aircraft active at time t.
c_{s,cb_j}	= Number of conflicts given a schedule s and a node-holding combination cb_j.
cb_j	= jth node-holding combination.
n_u	= Total number of node-holding combinations caused given an uncertainty plan u.
s	= Identify schedule.
$t_D(a)$	= Planned time for aircraft a to reach its destination.
t_i	= Time at which the ith plan is made.
t_{end}	= The longest taxi time of any aircraft under consideration.
p_{u,cb_j}	= Probability of getting jth node-holding combination given an uncertainty plan u.
u	= Uncertainty plan (amount of uncertainty injected).
w_0, w_1, w_2	= Weightings in the cost function.

Flow


Surface Data
Scenario
World Simulator
World Uncertainty
Schedule (Target Nodes)
Reschedule Loop
Next State
Sampling Loop -
Pick a schedule with minimum cost
Scheduler
Prediction Simulator
Guess Uncertainty

Implementation - SFO

Picked Terminal 2

To start modeling SFO surface data, I picked Terminal 2 as a starting point since it only contains one boarding area while other terminals contains at least two. This terminal is also close to runways comparing to others.

Data Source

The geodata of the runway, taxiways, and gates are fetched from Open-Street dataset. Pushback ways data are drawn manually on Google Map. Scenario data is randomly generated by a script.

75

Links

We model 14 taxiways and 1 runway. After breaking on intersections, we have 75 links.

16

Nodes

In terminal 2, there are 14 gates, and I manually plotted 2 spot positions.

Implementation - More

Conflict Detection and Handling

One conflict is detected if two aircrafts break the separation constraint. Although we don’t model movements of an aircraft, we can still calculate the exact location of an aircraft for conflict detection. The scheduler will handle **“some conflicts”** in order to minimize the cost calculated with the horizon. (Under discussion)

Uncertainty

We let the aircraft to hold on each node with N% of possibility. This leads to a delay of the schedule assigned to this aircraft.

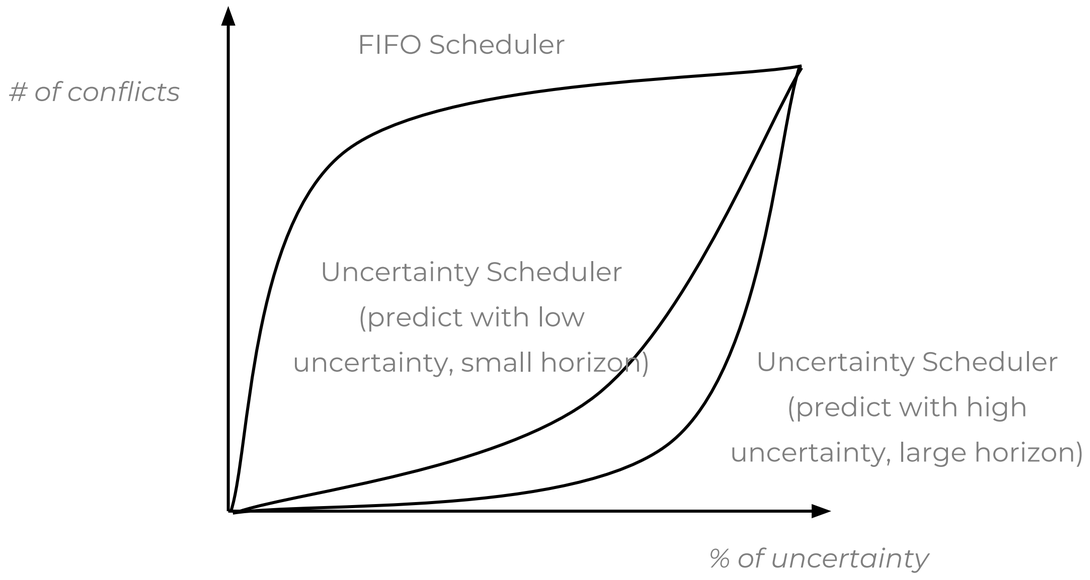
Queue at Edge Nodes

We will add queues at edge nodes, which are the gates and one end of the runway. This avoids conflicts at edge nodes which the scheduler have no control of.

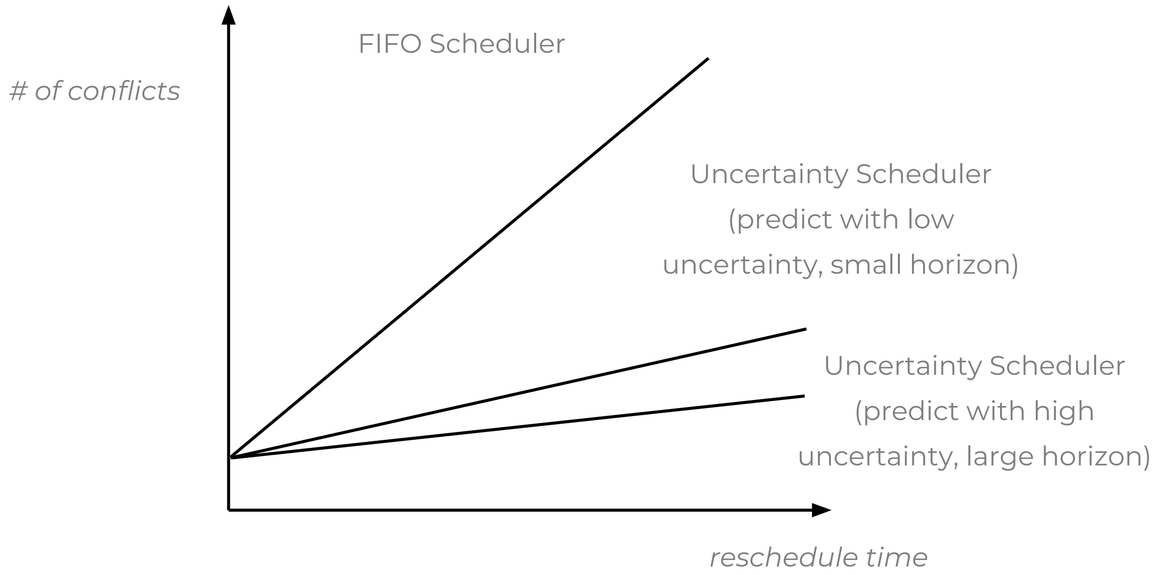
Experiment Designs

I will run following experiment like below, and the figures I expect to see.

In each experiment, our Uncertainty Scheduler is compared with the FIFO scheduler.



Handling Uncertainty



Reschedule Time

Timeline

2/26 - 3/7 SFO Terminal 2 Model

- Load and parse SFO terminal 2 data into the simulation. [DONE]
- Run the break-link-at-node algorithm. [DONE]
- Verify the data is parsed and can be routed successfully. [ALMOST]

3/7 - 3/12 Detect Conflicts

- Virtually model the real location of each aircraft while moving
- Find a list of conflicts at any time point

3/7 - 4/1 Uncertainty Scheduler (Cost Function)

- Add uncertainty (delay holds) into the the schedule [3/7 - 3/18]
- Calculate the cost function [3/19 - 3/24]
- Pick the best schedule [3/24 - 4/1]

4/2 - 4/14 Experiments

- Write scripts to enable multiple runs of simulation
- Run experiments that we planned
- Plot figures

4/2 - 4/14 Paper Writing

- Compose the paper

4/14 - 5/6 Paper Issue Fixing

- Fixes issues that we didn’t expect
- Fixes bugs in simulation
- Refine code for future usages

Thank you.