1 Cost Function

Given a schedule and a fixed amount of uncertainty, our cost function is definied as:

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

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.

---

2 Scheduling

In scheduling, we pick the schedule with minimum cost among limited amount of samples. The pseudocode code of schedule function is as below:

```
uncertainty ← PRE_DEFINED_VALUE

func schedule(airport_state, scenario) {

    # Initial schedule (deterministic, may contain conflicts)
    schedule ← fifo(airport_state, scenario)

    # Resolve conflicts with uncertainty considered
    min_cost ← MAX
    min_cost_schedule ← schedule

    for (1 ... iteration) {
        state ← simulator.predict(airport_state, schedule, uncertainty)
        cost ← cost_func(schedule)
        if (cost < min_cost) {
            min_cost ← cost
            min_cost_schedule ← schedule
        }
    }

    return schedule

}
```