Project Proposal: Simulation Engine for Evaluating GDP Planning Methods

Alex Estes
Advisor: Dr. Michael Ball

September 30, 2014
Bad weather can decrease the capacity of airport
This could cause a queue of airplanes to form in the air
To prevent this, Federal Aviation Administration (FAA) issues a Ground Delay Program (GDP)
Delay is taken on the ground instead of the air
Airlines may react to GDP
Background

- Many methods have been proposed for planning GDPs
- Not easy to determine effectiveness of proposed method
- Airline response not include in evaluation of GDP planning methods\(^\text{12345}\)

---


Goal

• More accurately and consistently evaluate effectiveness of GDP planning method
Approach

- Build a simulation
- Include GDP planning method and airline response
Inputs

- For each flight:
  - Flight number
  - Airline
  - Origin
  - Original scheduled departure time
  - Current scheduled departure time
  - Original scheduled estimated time of arrival (ETA)
  - Current scheduled ETA

- A set of destination capacity scenarios. For each one:
  - GDP start time
  - GDP end time
  - Time resolution of capacity scenarios
  - Capacity in each time interval
  - Probability
Information for GDP planning module:
- Which GDP planning method to use
- When to use the GDP planning method
- Tunable parameters for method if applicable

Other parameters:
- The time at which the simulation starts
- The time at which the simulation ends
- Time resolution of simulation
- Parameters describing distribution of actual departure times
- Parameters describing distribution of actual arrival times
- Parameters describing distribution of critical times
State Object

- The simulation will keep track of the state of the system
- This will include all of the input data, but also other fields
- Additional Fields:
  - Actual departure time (randomly generated)
  - Actual arrival time (randomly generated)
  - Critical time (randomly generated)
  - Delay Metrics
Program Organization

Program components:
1. Input parser
2. State update module
3. GDP planning module
4. Airline response module
Program Organization

- Input
  - Input Parser
- State Update Module
  - State object
  - Airline response module
  - GDP planning module
  - Metrics
  - State object
  - State object
  - State object
- State object
  - State object
  - State object
  - State object
  - State object 10
Algorithm 1 State Update

INPUT: a state object, as described in appendix one

1: for all flights $f$ whose actual departure time is the current time period do
  ▷ Handle flights trying to depart
  2: Let $a$ be the origin airport of flight $f$.
  3: Flight $f$ departs.
  4: Set an actual time of arrival with some randomness
  5: Update metrics
2: end for

7: for all flights $f$ whose actual time of arrival is the current time period do
  ▷ Handle flights trying to land
  8: if enough time has passed since last arrival at destination then
  9:   Flight $f$ lands.
10: Set a time with some randomness until next flight may land
11: Calculate total delay and update delay metrics
12: else
13:   Delay $f$’s current departure ETA by one time period
14:   Update delay metrics
15: end if
16: end for
GDP Planning Modules

- Rationing Heuristics
- Integer Programs (IPs)
Algorithm 2 RBS\textsuperscript{6}

**INPUT:** A list of time slots and a list of flights.
1. Sort flights by increasing original ETA
2. \textbf{for all} $f$, flights not airborne and non-exempt \textbf{do}
3. \hspace{1em} Set the scheduled ETA of $f$ to the earliest unoccupied slot which is later than its original ETA
4. \textbf{end for}

---

Algorithm 3 RBD

**INPUT:** A list of time slots and a list of flights.

1. Sort flights by decreasing distance and then by increasing original ETA
2. **for all** $f$, flights not airborne and non-exempt **do**
3. Set the scheduled ETA of $f$ to the earliest unoccupied slot which is later than its original ETA
4. **end for**
Algorithm 4 E-RBD\(^8\)

**INPUT:** A list of time slots and a list of flights and a maximum allowed delay \(D\)
1. Sort flights by increasing original ETA
2. **for all** \(f\), remaining flight **do**
3. the scheduled ETA of \(f\) to the earliest unoccupied slot which is later than its original ETA, mark this slot as temporarily occupied
4. **end for**
5. Sort flights by decreasing distance and then by increasing original ETA
6. **for all** \(f\), not airborne, non-exempt **do**
7. Find the earliest temporarily occupied slot that is later than the original ETA of \(f\) such that swapping time slots with the occupant would not delay the occupant more than the maximum allowed delay \(D\). Swap \(f\) into this slot, and mark the slot as permanently occupied.
8. **end for**

---

Parameters:

\[\Theta = \text{the set of capacity scenarios}\]
\[M^t_q = \text{the capacity in scenario } q \text{ at time } t \text{ for } q \in \Theta\]
\[P\{q\} = \text{the probability of scenario } q \in \Theta\]
\[\Phi = \text{the set of flights}\]
\[T = \text{the number of discretized time intervals}\]
\[\text{Arr}_f = \text{the earliest time period in which } f \text{ may arrive for } f \in \Phi\]
\[\text{Dep}_f = \text{the earliest time period in which } f \text{ may depart for } f \in \Phi\]
\[\lambda = \text{the ratio of cost of air delay to ground delay}\]
\[W_{max} = \text{the maximum number of flights allowed to queue at the destination}\]

\footnotesize{A. Mukherjee and M. Hansen. “A dynamic stochastic model for the single airport ground holding problem.” In: Transportation Science 41 (4 2007), pp. 444–456.}
Mukherjee-Hansen Dynamic IP Model

Decision Variables:

\( X_{q,f,t} \): Binary variable, 1 if flight \( f \) arrives at time \( t \) in scenario \( q \).

\( Y_{q,f,y} \): Binary variable, 1 if flight \( f \) departs at time \( t \) in scenario \( q \)

\( W_{q,t} \): the number of flights at in the arrival queue the end of time period \( t \)

Objective Function:

\[
\min \sum_{q \in \Theta} \left( \sum_{f \in \Phi} \sum_{t=\text{Arr}_f}^{T+1} (t - \text{Arr}_f) X_{q,f,t} + \lambda \sum_{t=1}^{T} W_{q,t} \right)
\]
Mukherjee-Hansen Dynamic IP Model

Constraints:

\[ Y_{f,t}^q = X_{f,t}^q \]

\[ \sum_{t=Arr_f}^T X_{f,t}^q = 1 \]

\[ W_{t-1}^q - W_t^1 + \sum_{f \in \Phi} X_{f,t}^1 \leq M_t^q \]

\[ W_t^q \leq W_{max} \]

\[ W_0^q = W_{t+1}^q = 0 \]

\[ Y_{f,t}^{S_{1,t}} = Y_{f,t}^{S_{2,t}} = \ldots = Y_{f,t}^{S_{n,t}} \]

\[ X_{f,t}^q \in \{0, 1\}, \; Y_{f,t}^1 \in \{0, 1\}, \; W_t^q \geq 0 \]
Parameters:

- $T$: number of time periods in the discretization of the GDP time period
- $S_t$: The number of flights with original ETA in time period $t$
- $Q$: Set of capacity scenarios
- $D_{t,q}$: The arrival capacity of the airport in time period $t$ in scenario $q$
- $p_q$: the probability that capacity scenario $q$ occurs
- $a_t$: cost of air delay at time $t$ (usually constant over $t$)
- $g_t$: cost of ground delay at time $t$ (usually constant over $t$)

---

Hofkin Model

Variables:

\( x_t \): number of flights with arrival time in period \( t \)
which are assigned to arrive in time period \( t \)

\( y_t \): number of flights with arrival time in period \( t \)
who are ground delayed at least until period \( t + 1 \)

\( z_{t,q} \): number of flights which arrival in time period \( t \)
who are air delayed at least until time period \( t + 1 \) in scenario \( q \)

Objective function:

\[
\min \sum_{t=1}^{T+1} \left( g_t y_t + \sum_{q=1}^{Q} p_q a_t z_{tq} \right)
\]
Constraints:

\[ x_t + y_t - y_{t-1} = S_t \]
\[ z_{t-1,q} + x_t - z_{t,q} \leq D_{t,q} \]
\[ y_0 = y_{T+1} = 0 \]
\[ z_{0,q} = z_{T+1,q} = 0 \]
\[ x_t, y_t \geq 0 \]
\[ z_{t,q} \geq 0 \]
\[ \forall t \in \{1, \ldots, T + 1\} \]
\[ \forall t \in \{1, \ldots, T + 1\}, q \in Q \]
\[ \forall q \in Q \]
\[ \forall t \in \{1, \ldots, T + 1\} \]
\[ \forall t \in \{1, \ldots, T + 1\}, q \in Q \]
Richetta-Odoni model\textsuperscript{11,12}

Parameters:

\[ g(n) : \] the cost of ground delaying a flight \( n \) time periods
all others: as in Hofkin model

Variables:

\( x_{t,s} : \) number of flights with scheduled arrival time in period \( t \)
which are assigned to arrive in time period \( s \)
\( z_{t,q} : \) number of flights which held in air at destination at end of period \( t \)
in scenario \( q \)


\textsuperscript{12} O. Richetta and A. R. Odoni. “Solving optimally the static ground-holding policy problem in air traffic control.” In: \textit{Transportation Science} 27 (3 1993), pp. 228–238.
Richetta-Odoni Model

Objective function:

\[
\min \sum_{t=1}^{T} \sum_{s=t}^{T+1} G(s-t) x_{t,s} + \sum_{q=1}^{Q} p_{q} \sum_{t=1}^{T} a_{t} z_{t,q}
\]

Constraints:

\[
\sum_{s=t}^{T+1} x_{t,s} = S_{t} \quad \forall s = \{1, ..., T\}
\]

\[
\sum_{t=1}^{T+1} x_{t,s} + z_{t-1,q} - z_{t,q} \leq D_{t,q} \quad \forall s \in \{1, ..., T+1\}, q \in Q
\]

\[
z_{0,q} = z_{T+1,q} = 0 \quad \forall q \in Q
\]

\[
x_{t,s}, z_{t,q} \geq 0 \quad \forall t \in \{1, ..., T+1\}, q \in Q
\]
Parameters:

\[ \Phi = \text{the set of non-airborne flights for the airline} \]
\[ S = \text{the set of arrival times that are available for the airline} \]
\[ \text{Arr}_f = \text{the earliest arrival time slot of } f \text{ may arrive for } f \in \Phi \]
\[ \text{Crit}_f = \text{the critical time slot of } f \]
\[ \lambda = \text{the cost of normal ground delay} \]
\[ \mu = \text{the cost of delay past critical time} \]
Luo-Yu Airline Response model

Decision Variables:

\[ X_{f,s} \]: Binary variable, 1 if flight \( f \) is assigned to arrive in slot \( s \).

Objective Function:

\[
\min \sum_{f \in \Phi} \sum_{\substack{s \in S \atop \text{Arr}_f \leq s \leq \text{Crit}_f}} \lambda (s - \text{Arr}_f) X_{f,s} + \mu \sum_{s \in S \atop \text{Crit}_f \leq s} X_{f,s}
\]
Luo-Yu Airline Response model\textsuperscript{13}

Constraints:

\[
\sum_{s=\text{Arr}_f}^S X_{f,s} = 1 \quad \forall f \in \Phi
\]

\[
\sum_{f \in \Phi} X_{f,s} = 1 \quad \forall s \in \{1, \ldots, S\}
\]

\[
X_{f,s} \geq 0 \quad \forall f \in \Phi, s \in \{1, \ldots, S\}
\]

Implementation

- Java with Gurobi API
- IPs: NP-Hard, computationally tractable
- Other algorithms: less computationally intensive
- Running simulation once should not be overly time-consuming
- If many runs of simulation are needed, could take a long time
- Potentially could be parallelized
- Will run on personal laptop or office desktop
Databases:

- FAA Advisories database:
  http://www.fly.faa.gov/adv/advAdvisoryForm.jsp
- Bureau of Transportation Statistics on-time performance database:
  http://www.transtats.bts.gov/Fields.asp?Table_ID=236
- FAA Aviation System Performance Metrics database:
Validation

- For state update model and rationing heuristics can create small test cases which can be compared manually.
- For Richetta-Odoni and Hofkin model: there exist test results with complete descriptions of data used\(^\text{14}\)\(^\text{15}\)
- For Luo-Yu and Mukherjee-Hansen models: tests were run, incomplete descriptions of parameters used.
- For these, make reasonable estimates of test problems, make rough comparisons.
- Integrated testing: can make test cases which can be compared manually to ensure that modules are interacting properly.


Testing realism of simulation:
- We know metrics of real days and have historical flight schedules on these days
- FAA uses RBS
- Run simulation with RBS and airline response module, compare output metrics to real ones
Expected Results

- More realistic evaluation of performance of GDP planning methods
- Estimated metrics for a variety of days
- A framework which can easily accommodate other GDP planning methods and airline response models
Concluding Remarks

Summary:
- Build a simulation for evaluating GDP planning methods
- Include input parser, state update module, GDP planning module, and airline response module
Timeline

- **October:** Complete input parser and state update module. Create test cases to validate.
- **November:** Implement and validate RBS, RBD and E-RBD GDP planning modules.
- **December:** Prepare end of fall presentation, begin work on airline recovery module.
- **January:** Finish implementing and validating airline recovery module.
- **February:** Implement and validate IP-based planning modules.
- **March:** Integrated testing.
- **April:** Parallelization if necessary, otherwise develop GUI.
- **May:** Prepare report
Milestones

- Milestone 1: Simulation framework with input parser and state update module
- Milestone 2: Simulation framework with input parser, state update module, and RBS/RBD/E-RBD GDP planning modules
- Milestone 3: Simulation framework with input parser, state update module, RBS/RBD/E-RBD GDP planning modules and airline response
- Milestone 4: Completed simulation framework
- Milestone 5a: Parallelized simulation framework if necessary.
- Milestone 5b: Graphic user interface if time allows.
Deliverables

The deliverables of the project are as follows:

- Weekly reports
- Project proposal
- Mid-semester report
- Mid-year status report
- Final report
- Complete simulation framework including input parser, state update module, GDP planning modules and airline response module
- Test cases for validation of each module
- Integrated test cases for validation of framework


