SARDA Surface Schedulers

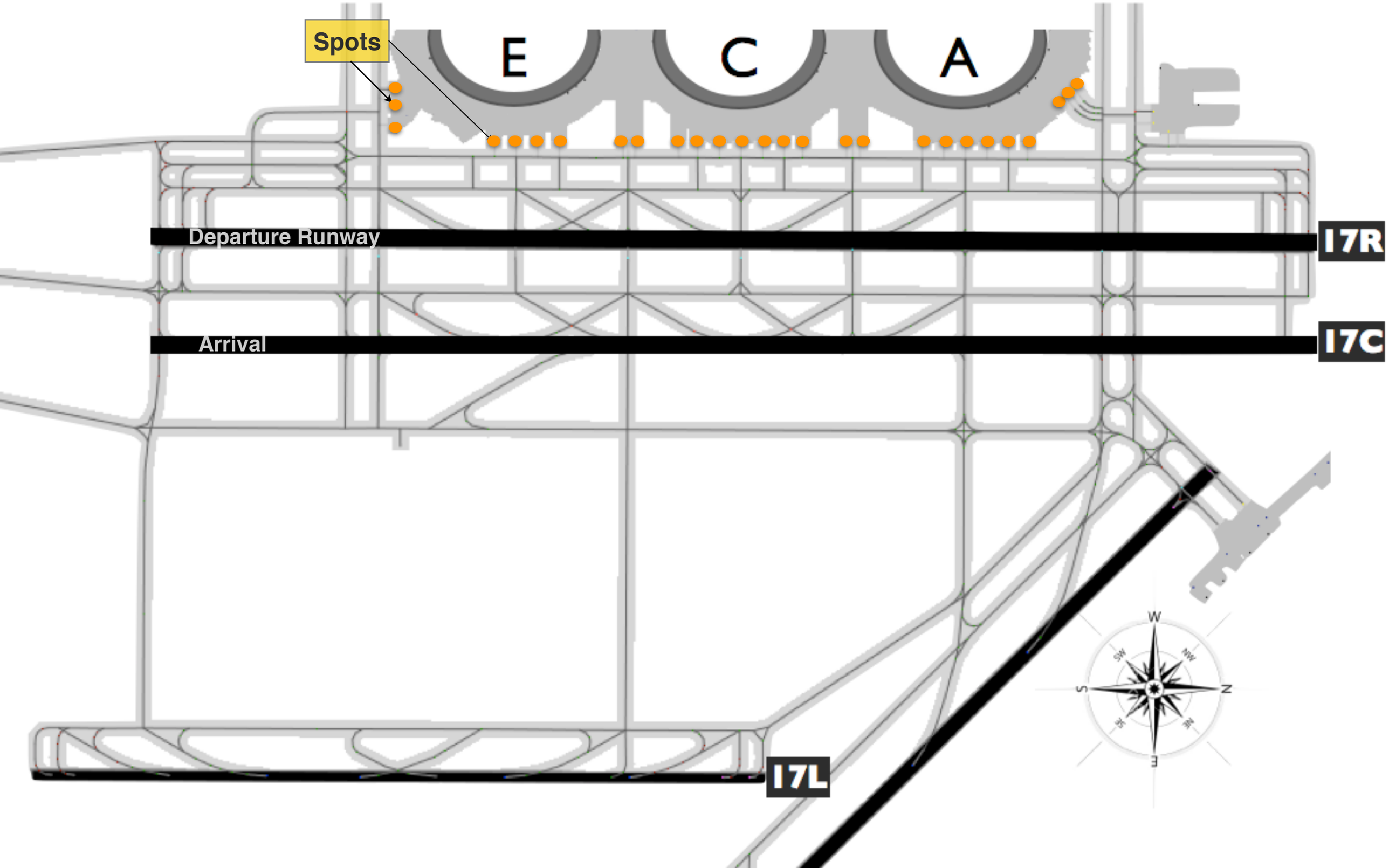
Waqar Malik
Universities Space Research Association

3rd Joint Workshop for
KAIA/KARI - NASA ATM Research Collaboration
October 24-26, 2016

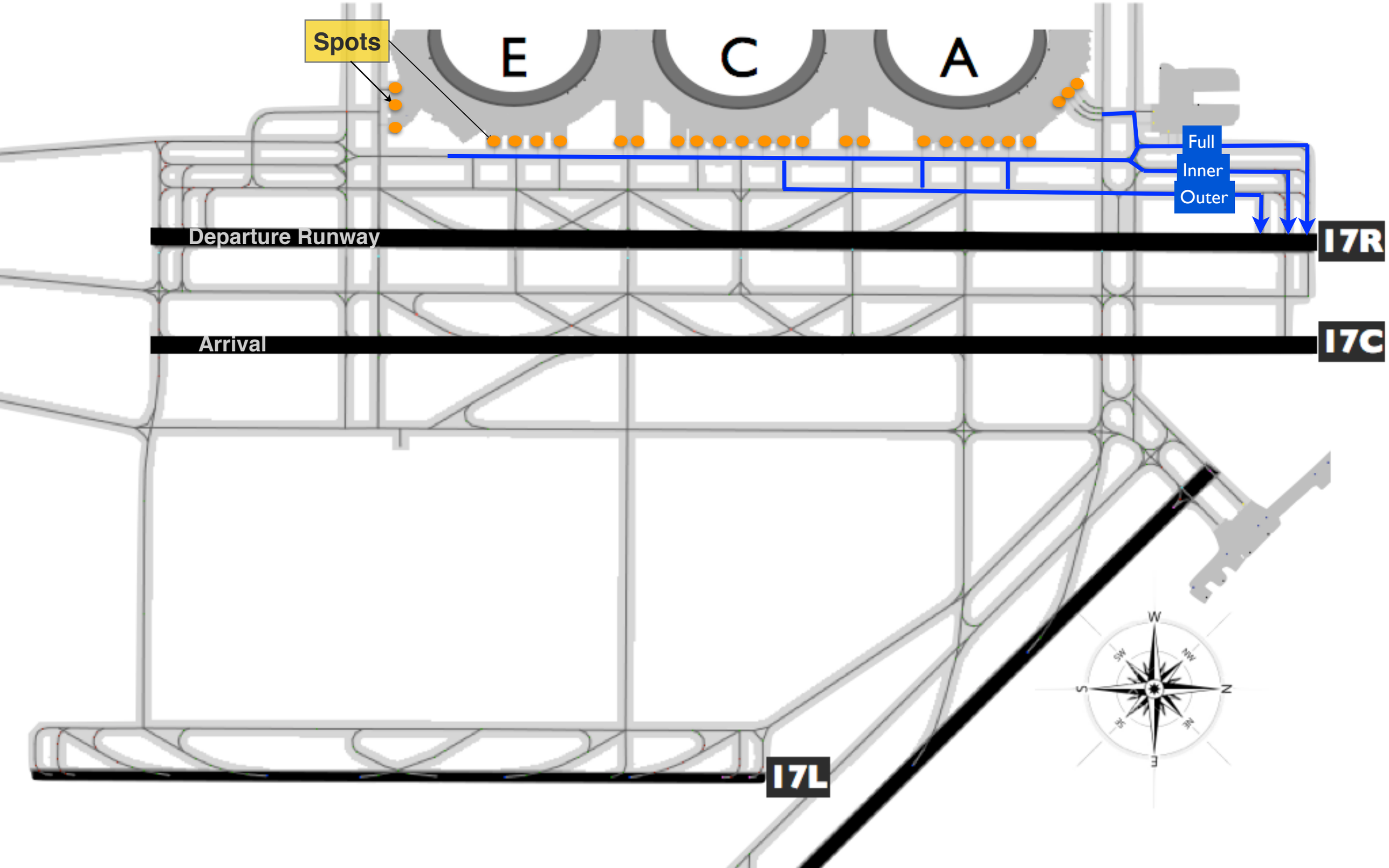
Outline

- Overview of airport surface management
- Single runway scheduling
  - One exact algorithm
  - Two heuristics-based algorithms
  - Simulation setup and results
- Multiple runway scheduling
  - CLT airport layout
  - Mixed Integer Linear Program Formulation
  - Simulation setup and results

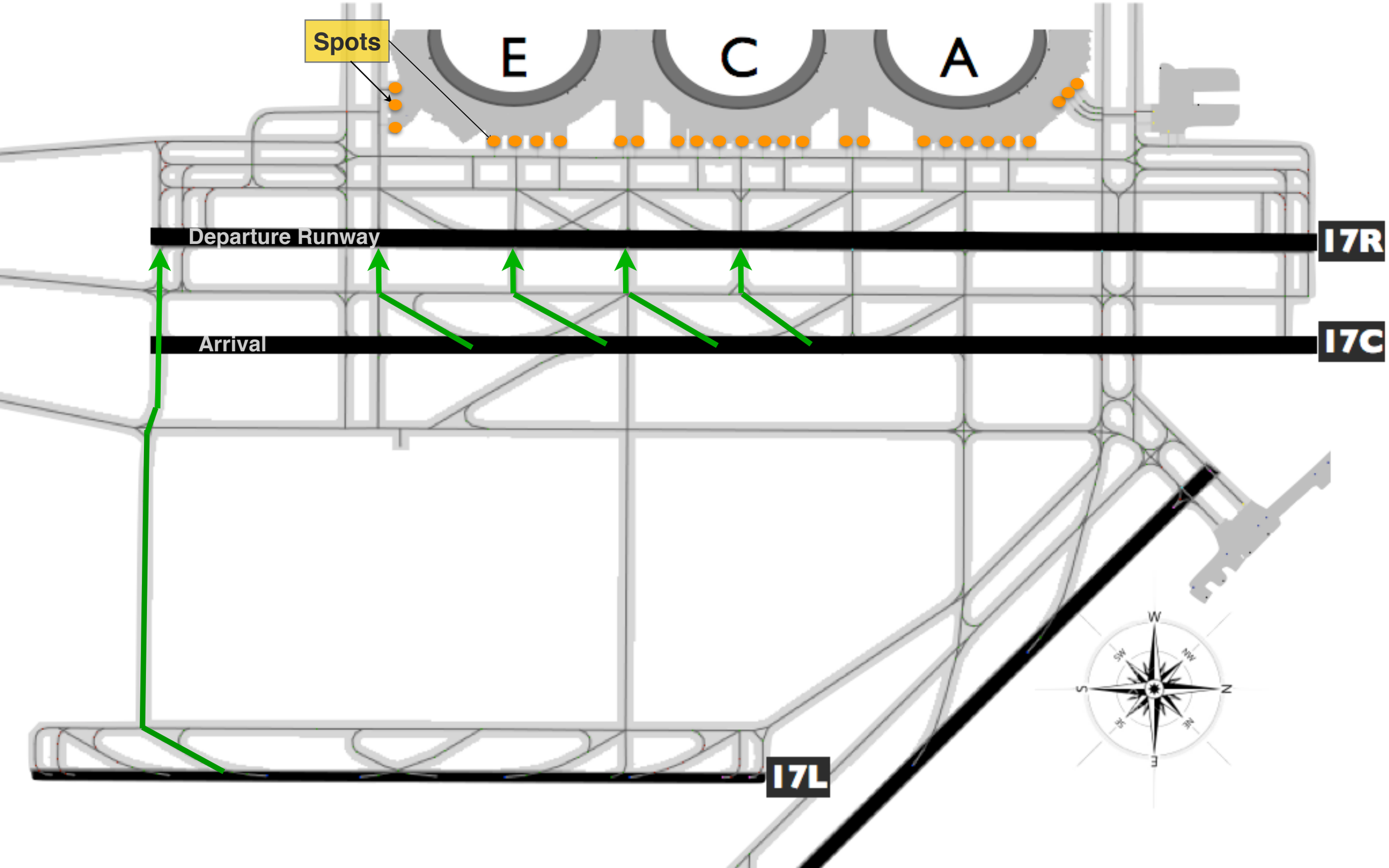
Current Operations



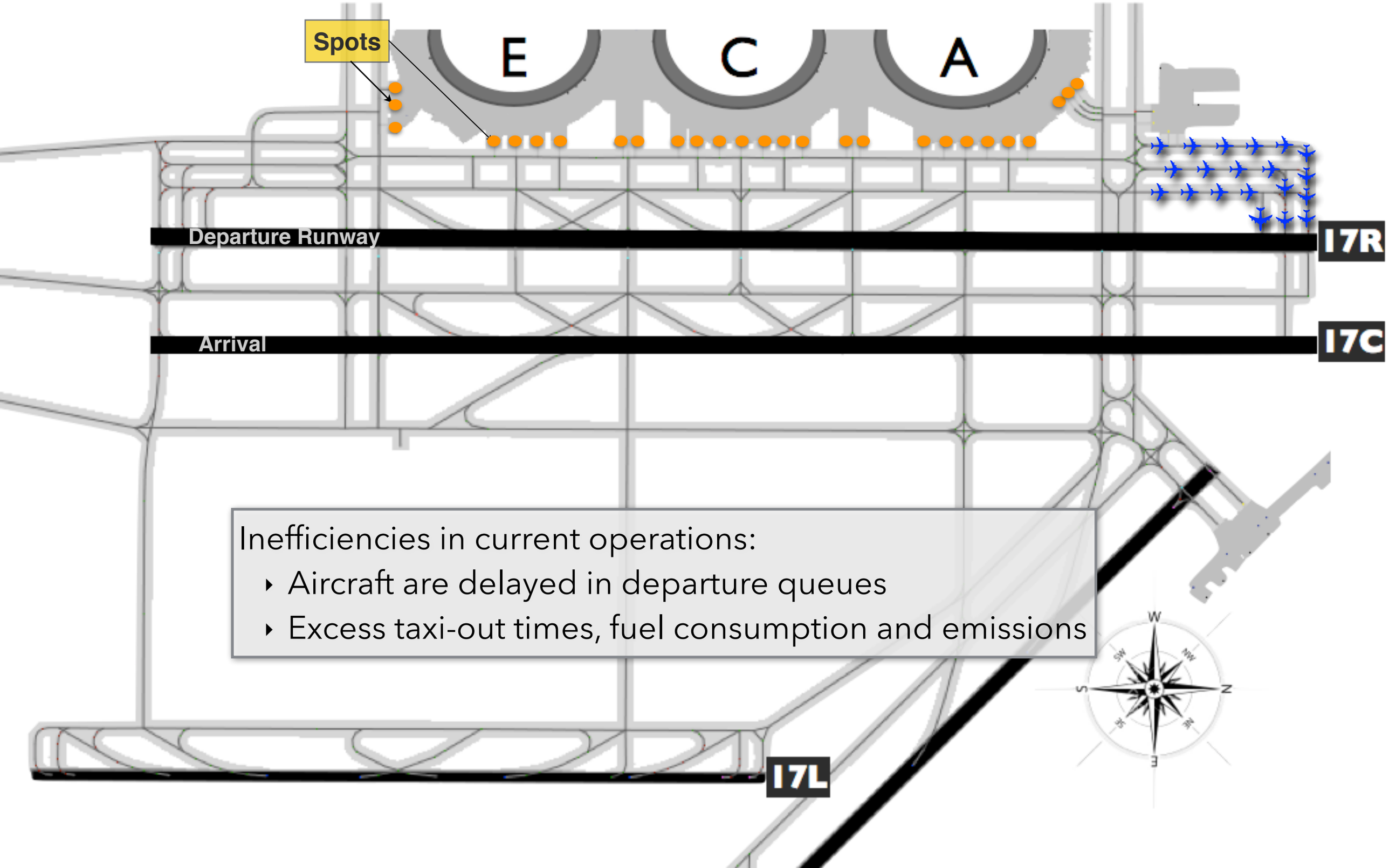
Current Operations



Current Operations



Current Operations



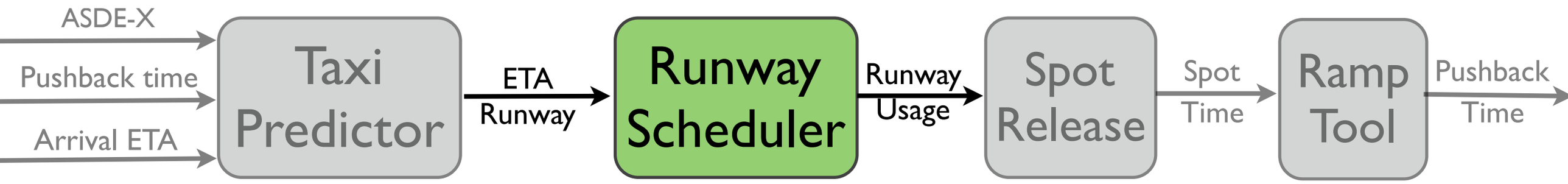
Inefficiencies in current operations:

- Aircraft are delayed in departure queues
- Excess taxi-out times, fuel consumption and emissions

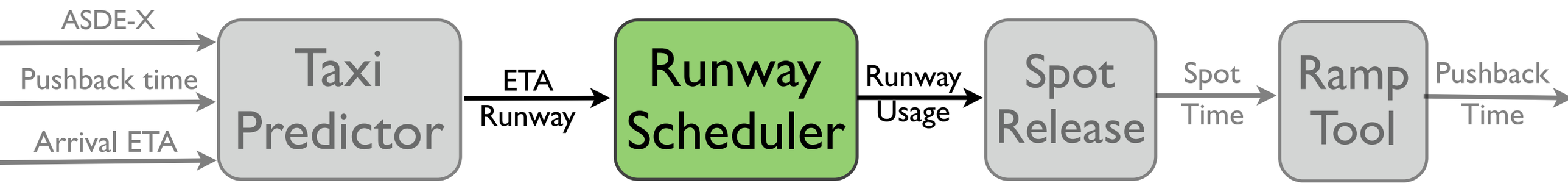
Surface Optimization at NASA

- Spot and Runway Departure Advisor (SARDA) provides a departure metering capability by efficiently scheduling aircraft on airport surface
- Human-in-the-loop simulations (2010, 2012)
  - Dallas/Fort Worth Airport (DFW), East Tower
  - Advisories provided to tower controllers
- Human-in-the-loop simulations (2014)
  - Charlotte Douglas International Airport (CLT)
  - Advisories provided to ramp controllers

SARDA Concept

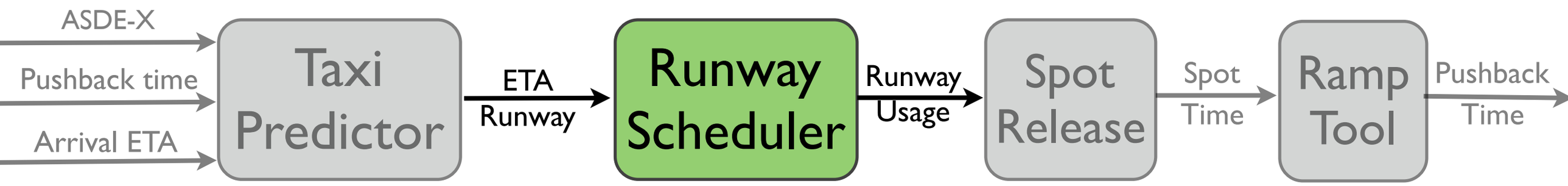


SARDA Concept



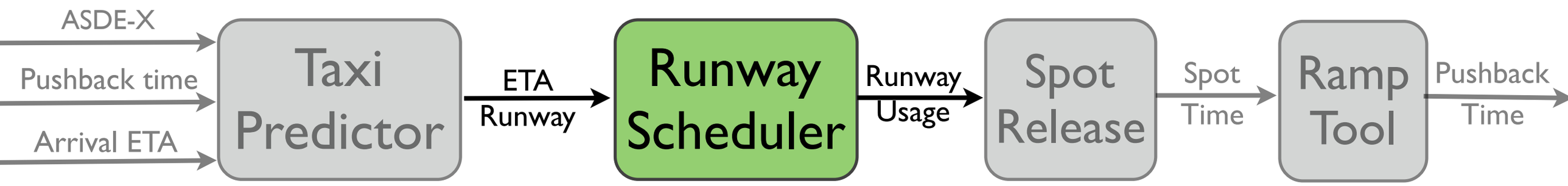
- A collaborative decision support tool for airlines and tower controllers to enhance the efficiency of surface traffic

SARDA Concept



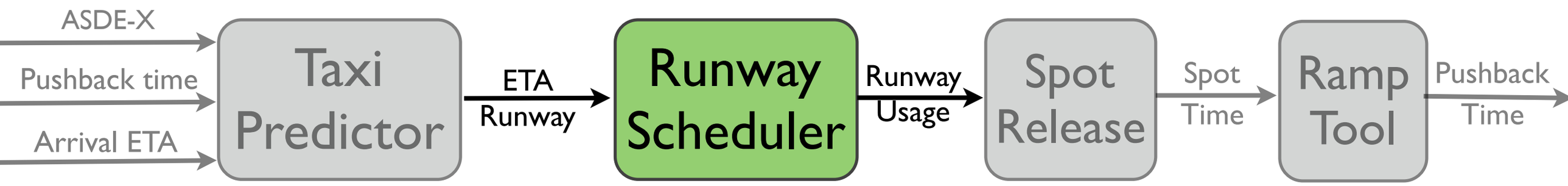
- A collaborative decision support tool for airlines and tower controllers to enhance the efficiency of surface traffic
- Provides advisories to Air Traffic Control Tower controllers and airline operators

SARDA Concept



- A collaborative decision support tool for airlines and tower controllers to enhance the efficiency of surface traffic
- Provides advisories to Air Traffic Control Tower controllers and airline operators
- Efficient Runway Scheduler that incorporates aircraft specific constraints, as well as arrivals

SARDA Concept

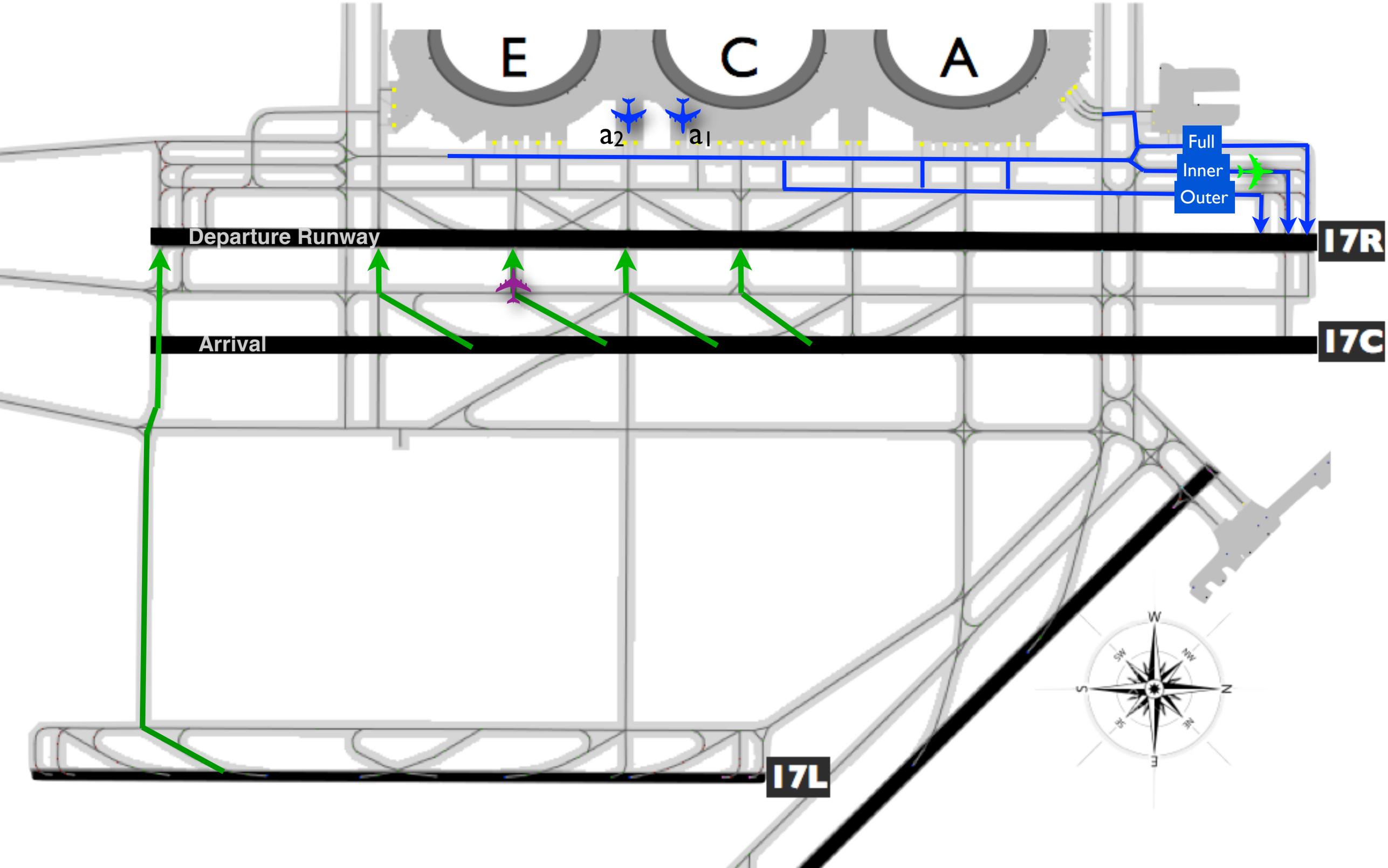


- A collaborative decision support tool for airlines and tower controllers to enhance the efficiency of surface traffic
- Provides advisories to Air Traffic Control Tower controllers and airline operators
- Efficient Runway Scheduler that incorporates aircraft specific constraints, as well as arrivals
- Both computation time and solution quality are critical factors in deciding a solution technique for Runway Scheduler

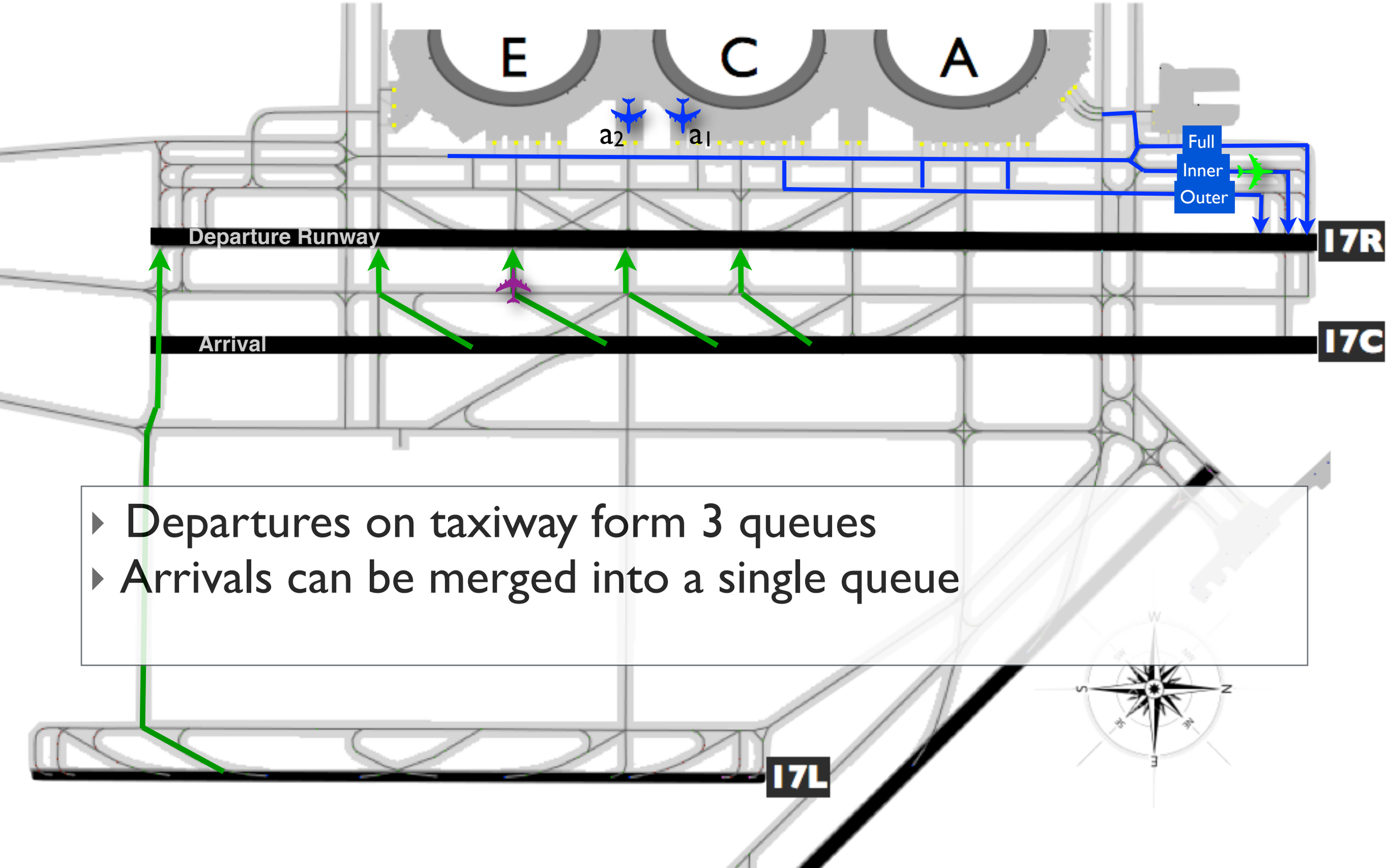
Outline

- Overview of airport surface management
- **Single runway scheduling**
  - One exact algorithm
  - Two heuristics-based algorithms
  - Simulation setup and results
- Multiple runway scheduling
  - CLT airport layout
  - Mixed Integer Linear Program Formulation
  - Simulation setup and results

Single Runway Scheduling

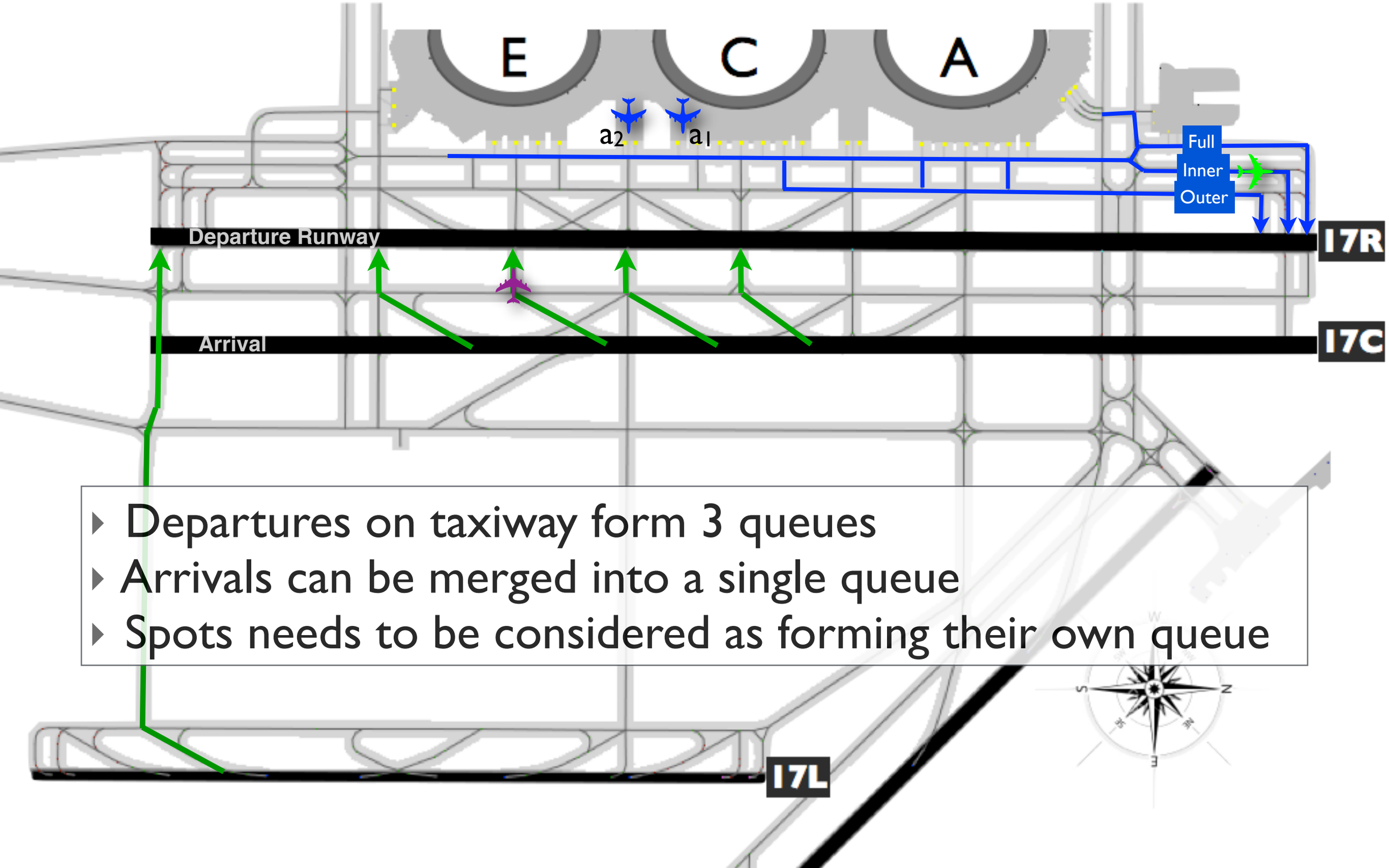


Single Runway Scheduling



- Departures on taxiway form 3 queues
- Arrivals can be merged into a single queue

Single Runway Scheduling



- Departures on taxiway form 3 queues
- Arrivals can be merged into a single queue
- Spots needs to be considered as forming their own queue

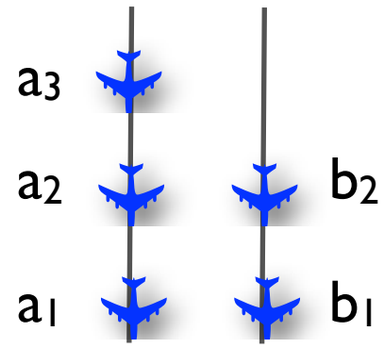
Single Runway Scheduling

- Exact Dynamic Program (EDP)
- Restricted Dynamic Program (RDP)
- Insertion and Local Search (ILS)

Runway Scheduler: Inputs

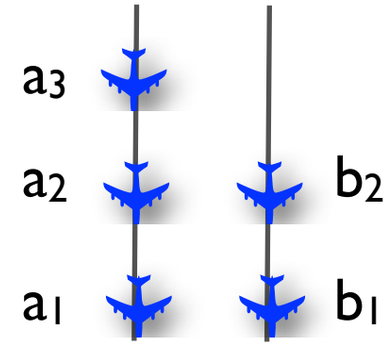
- For each aircraft:
  - Earliest available time
  - Spot, surface route, position, and fix/exit
  - Weight class and operation type
- Wake-vortex separation criteria and RNAV separation
- Separation between arrivals and departures for mixed use runway
- Separation between arrivals and departures for runway crossings
- Individual time-windows of intended take-off times for departing aircraft — Expect Departure Clearance Time (EDCT) and Call For Release (CFR)

Exact Dynamic Program



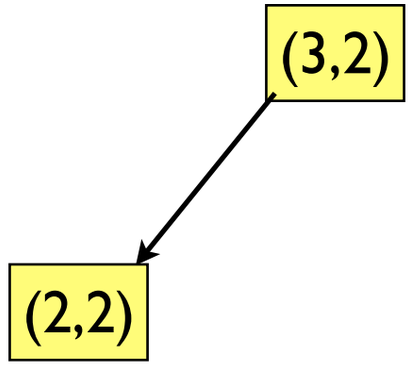
Exact Dynamic Program

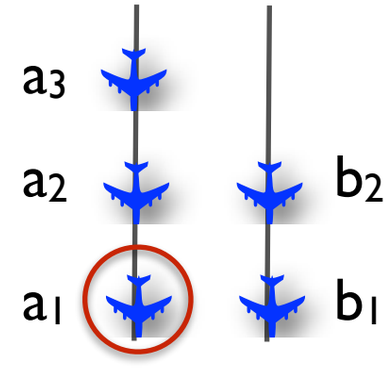
(3,2)



Partial solution:

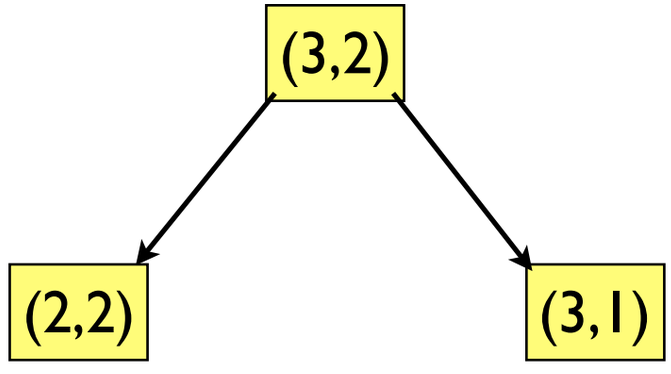
Exact Dynamic Program

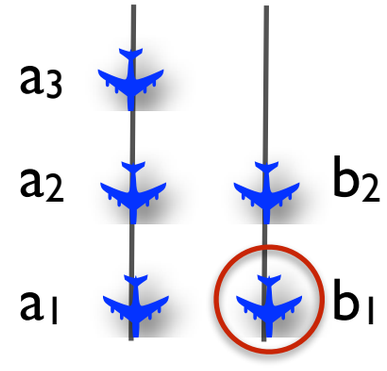




Partial solution: $\{a_1\}$

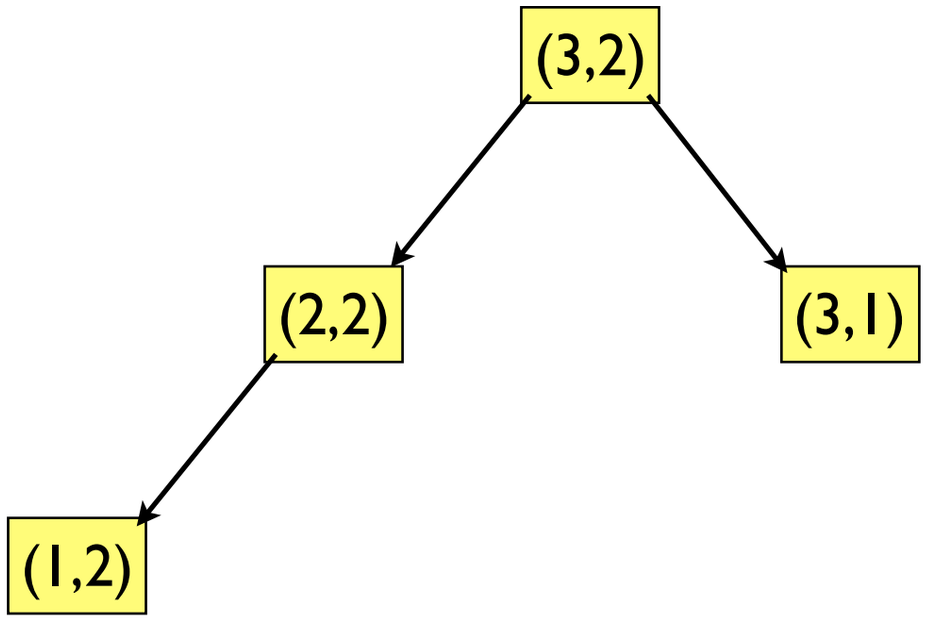
Exact Dynamic Program

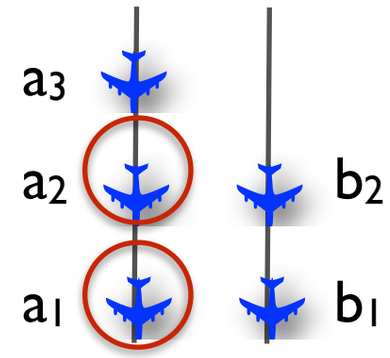




Partial solution: $\{b_1\}$

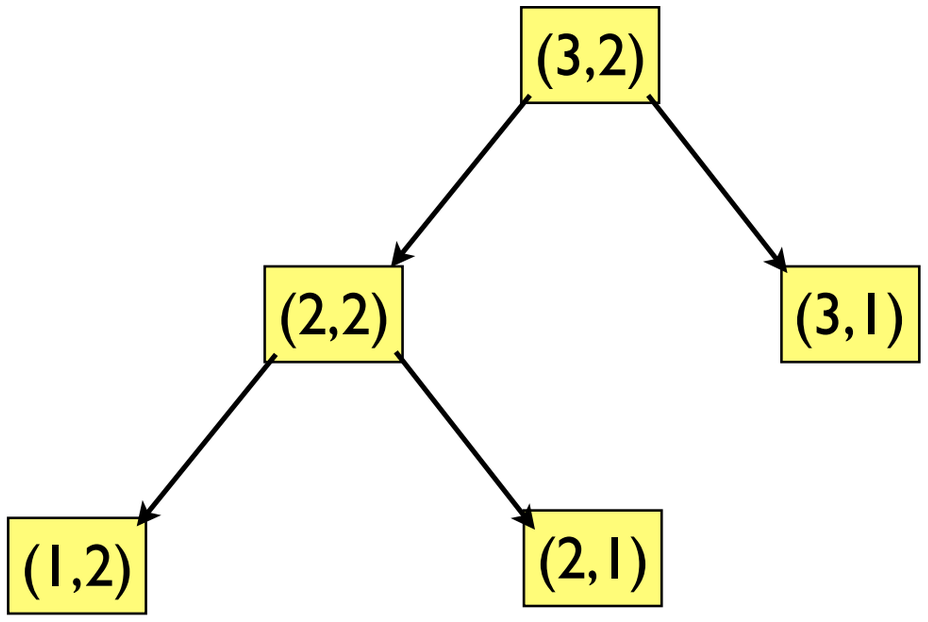
Exact Dynamic Program

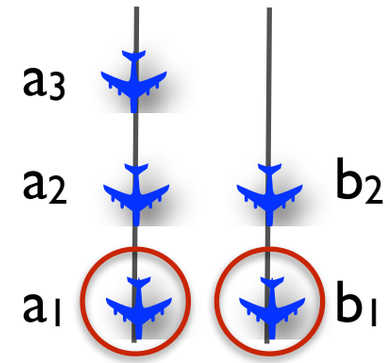




Partial solution: $\{a_1, a_2\}$

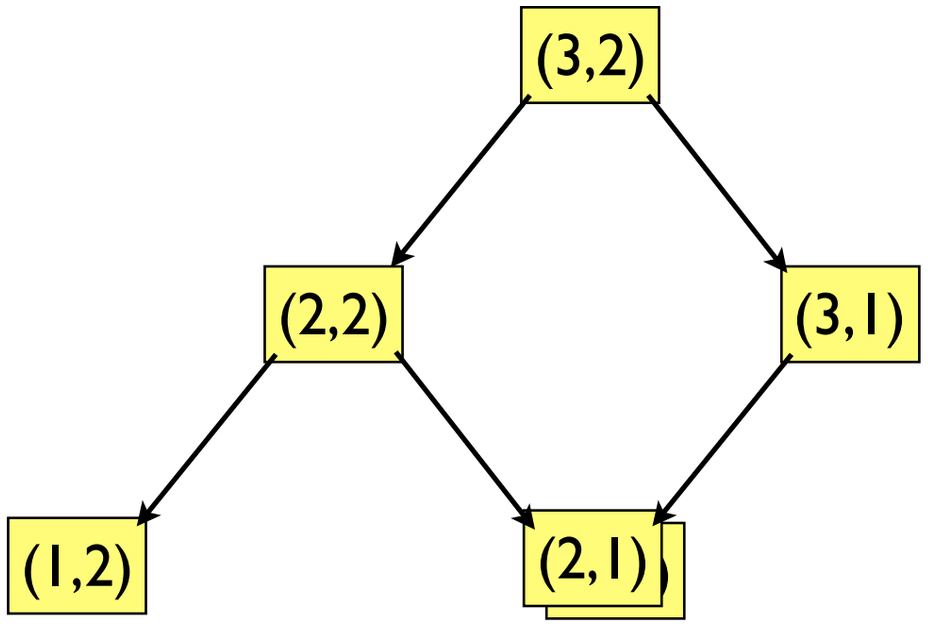
Exact Dynamic Program

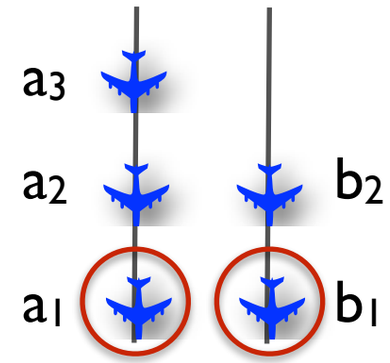




Partial solution: $\{a_1, b_1\}$

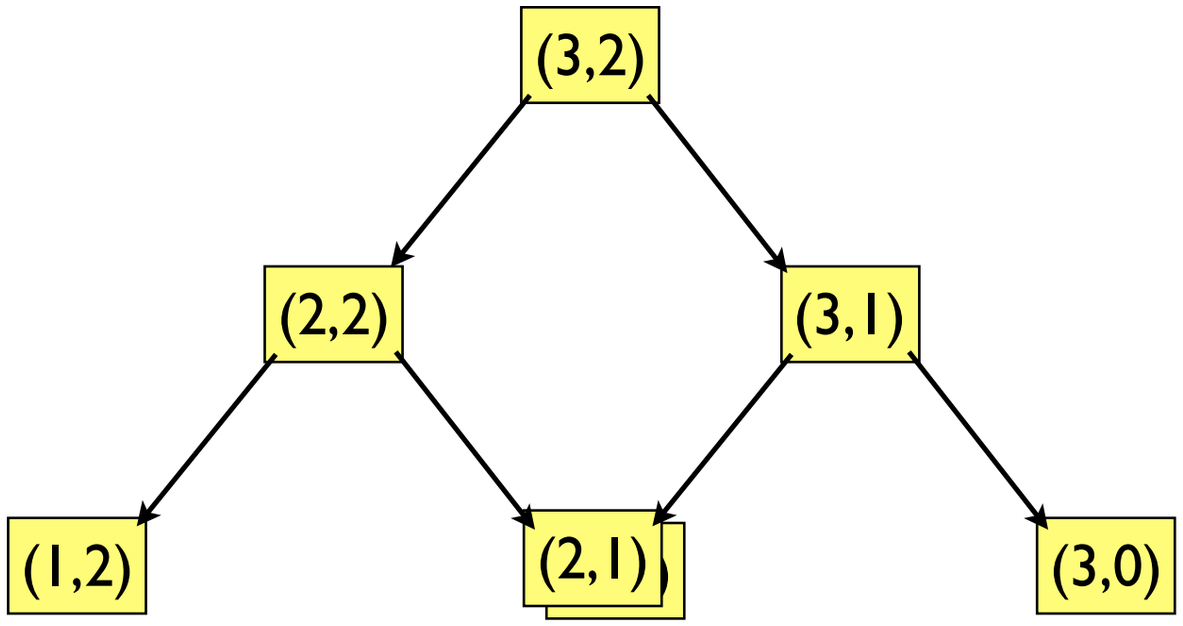
Exact Dynamic Program

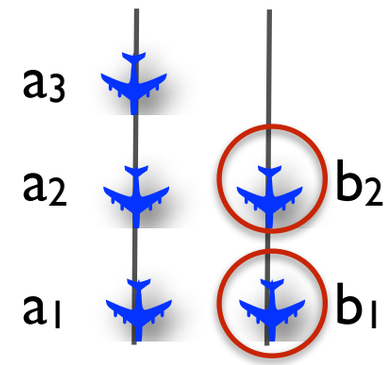




Partial solution: $\{a_1, b_1\}, \{b_1, a_1\}$

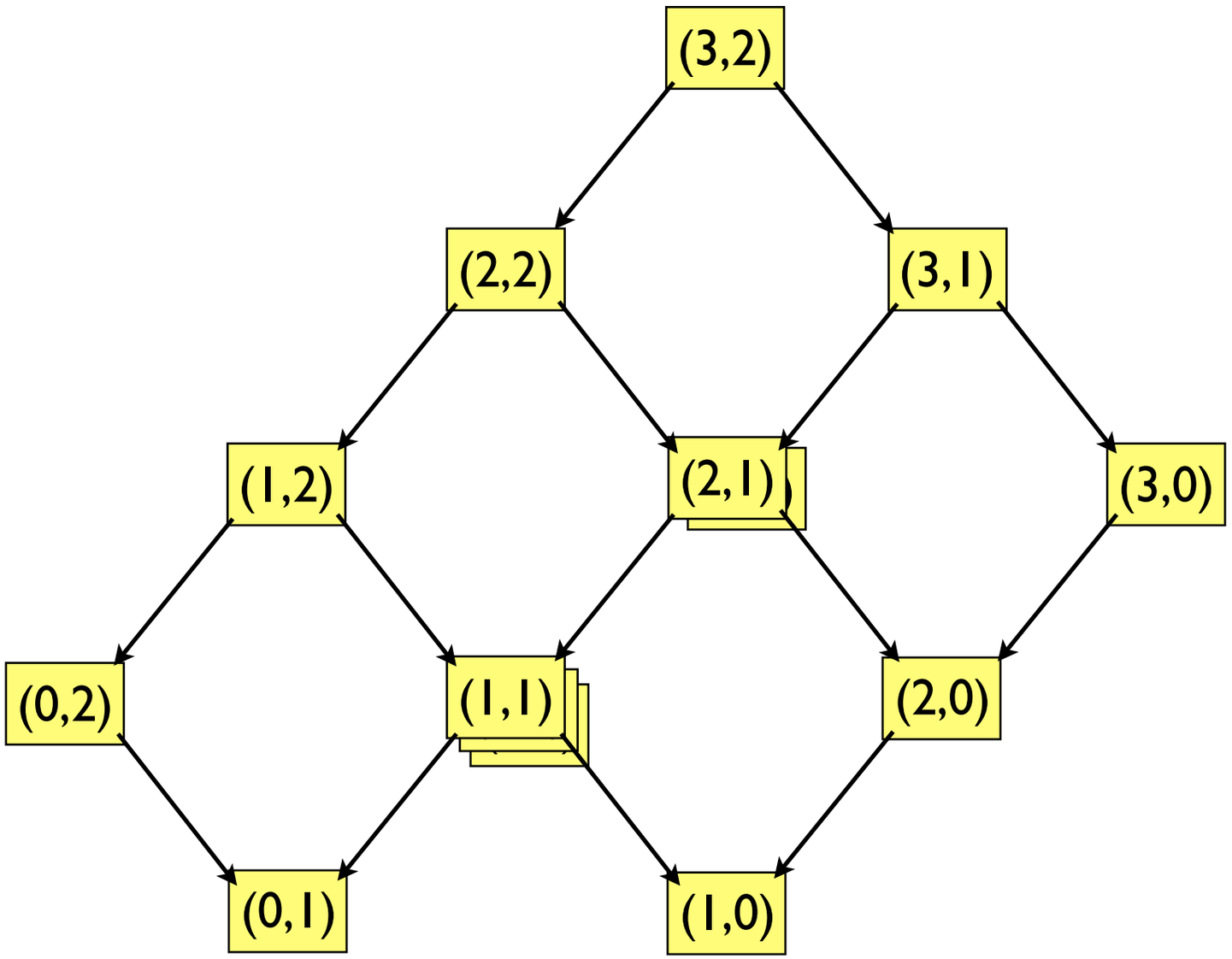
Exact Dynamic Program

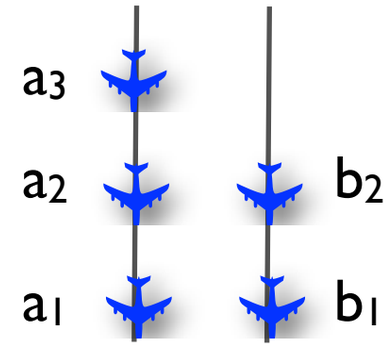




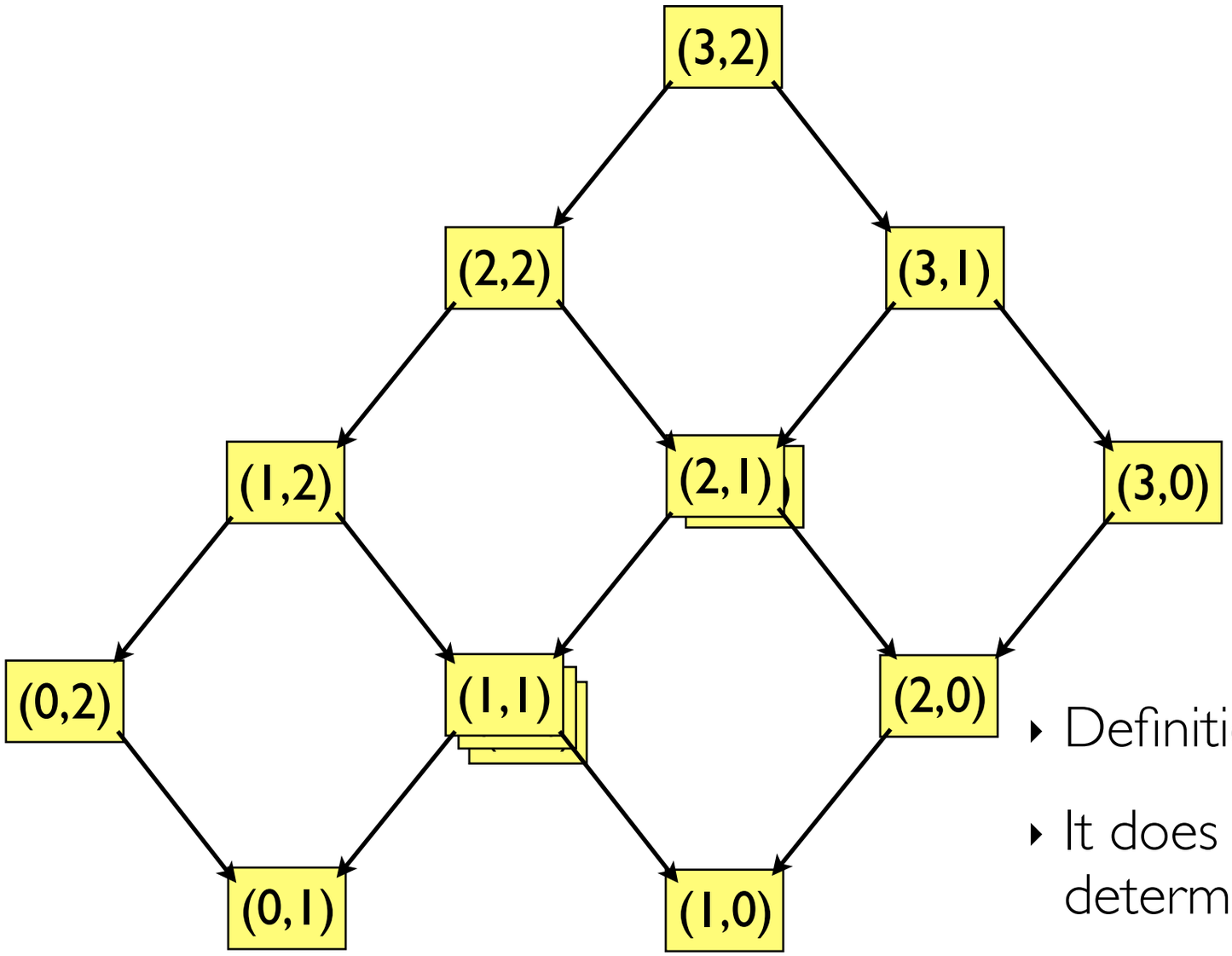
Partial solution: $\{b_1, b_2\}$

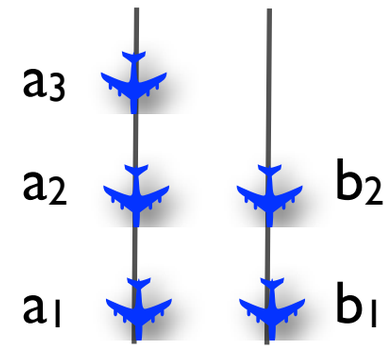
Exact Dynamic Program





Exact Dynamic Program





- Definition of state may not be rich enough
- It does not carry enough process history to determine optimality of remaining decisions
- Enhance state definition or consider multiple objectives

Exact Dynamic Program

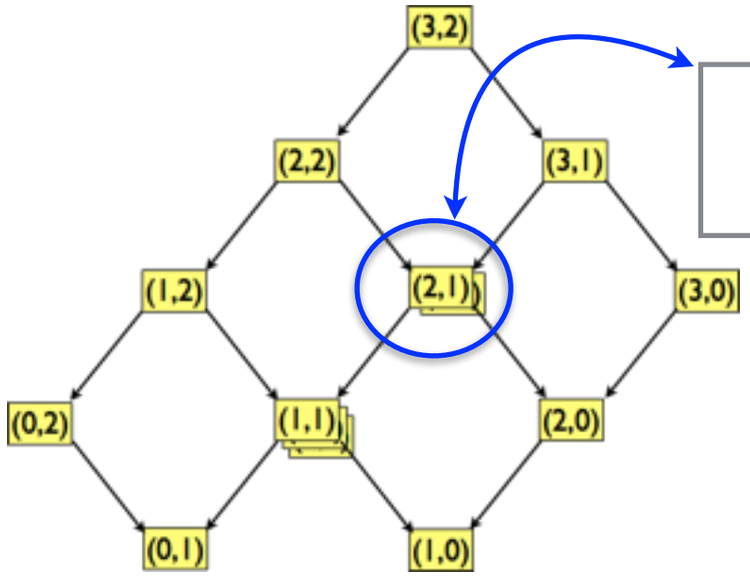
- State Definition:
  - heading of last departure
  - weight-class of last departure
  - last operation type
  - #aircraft in queue 1
  - #aircraft in queue 2
  - .
  - .
  - .
  - #aircraft in queue Q

Exact Dynamic Program

- State Definition:
  - heading of last departure
  - weight-class of last departure
  - last operation type
  - #aircraft in queue 1
  - #aircraft in queue 2
  - .
  - .
  - .
  - #aircraft in queue Q
- Value Function:
  - Last time a departure took off
  - Makespan
  - Cumulative delay

Exact Dynamic Program

- State Definition:
  - heading of last departure
  - weight-class of last departure
  - last operation type
  - #aircraft in queue 1
  - #aircraft in queue 2
  - .
  - .
  - .
  - #aircraft in queue Q
- Value Function:
  - Last time a departure took off
  - Makespan
  - Cumulative delay



Pareto dominance can be applied

Outline

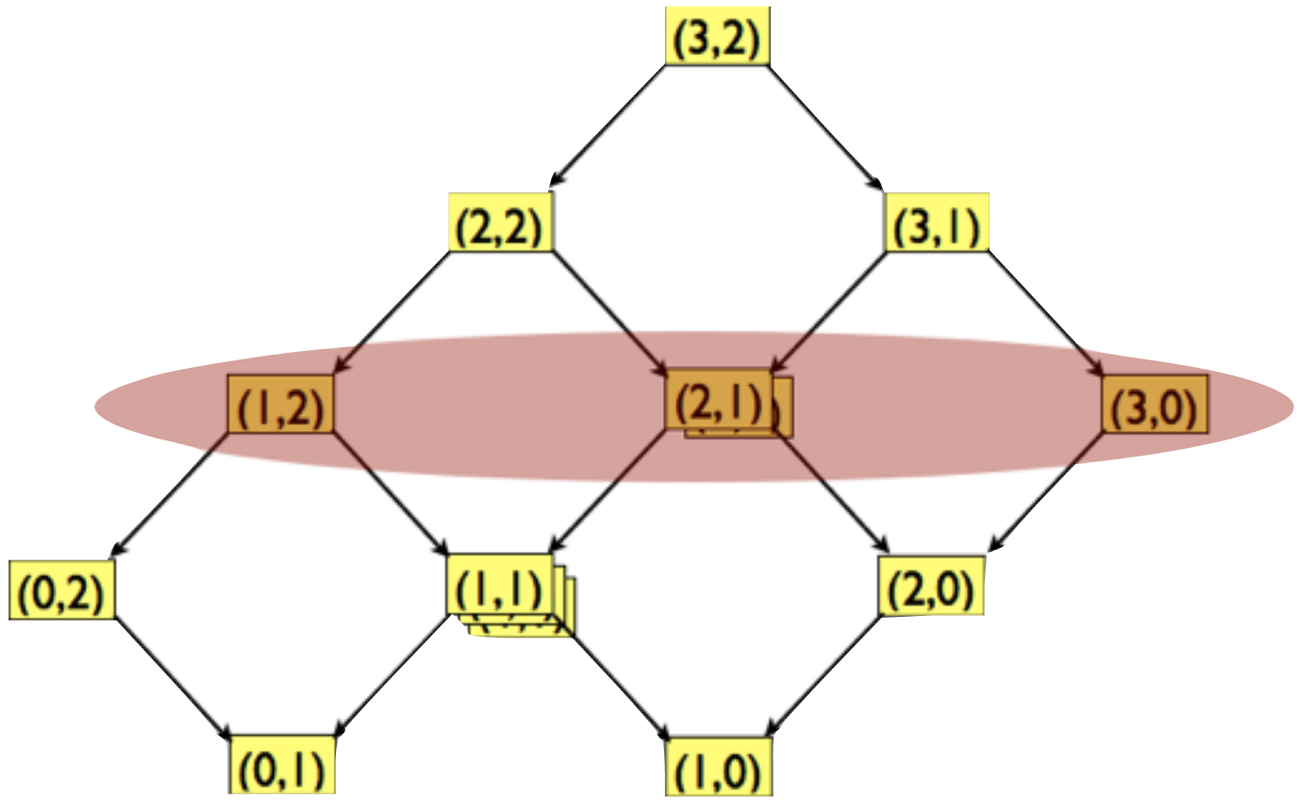
- Overview of airport surface management
- Single runway scheduling
  - One exact algorithm
  - **Two heuristics-based algorithms**
  - Simulation setup and results
- Multiple runway scheduling
  - CLT airport layout
  - Mixed Integer Linear Program Formulation
  - Simulation setup and results

Restricted Dynamic Program

- Number of nodes in DP is proportional to product of number of aircraft in each queue
- Some stages of the EDP formulation of the SRS could have a large number of states

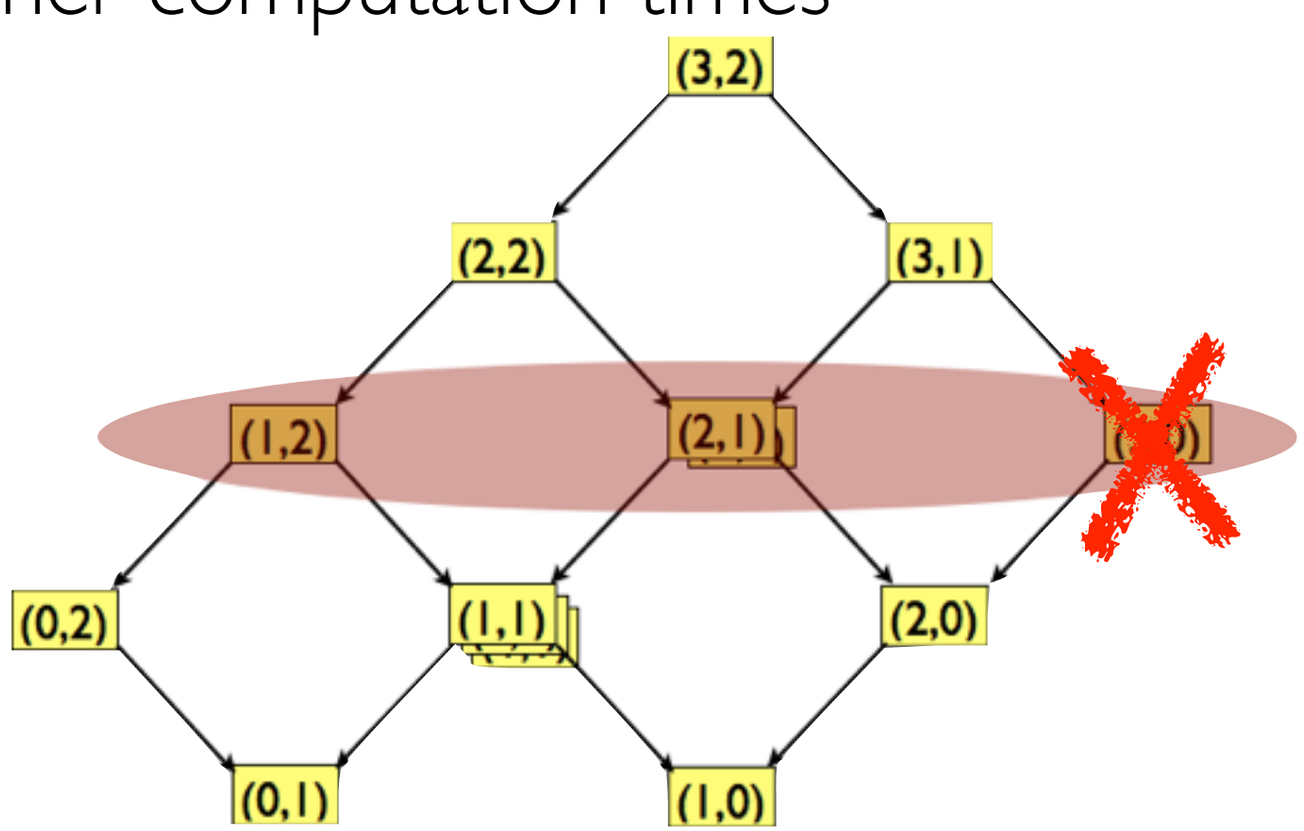
Restricted Dynamic Program

- Number of nodes in DP is proportional to product of number of aircraft in each queue
- Some stages of the EDP formulation of the SRS could have a large number of states



Restricted Dynamic Program

- In each stage, only a restricted subset of H states with the smallest delay is kept
- Increasing the value of H should yield better solutions, but will also result in higher computation times

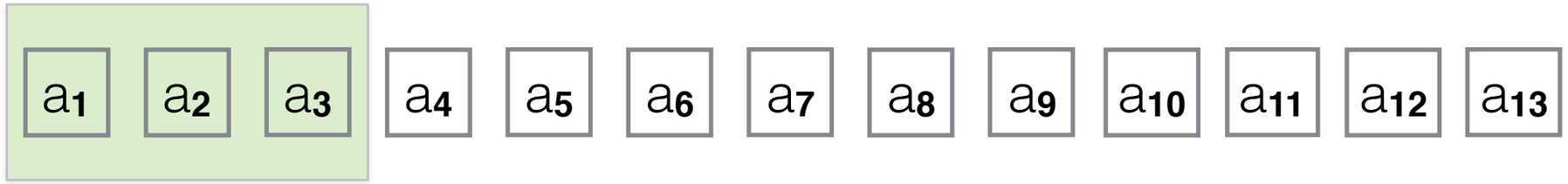


Insertion and Local Search

- Start with a First-Come-First-Served (FCFS) initial solution

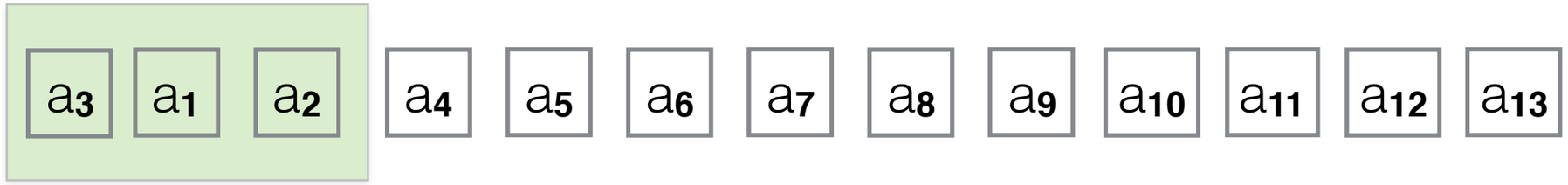
Insertion and Local Search

- Find all permutations of ‘k’ free aircraft



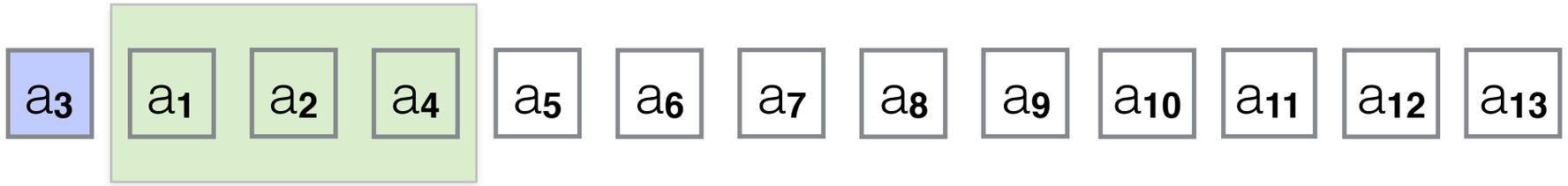
Insertion and Local Search

- Find all permutations of ‘k’ free aircraft
- Select sequence which gives best objective value



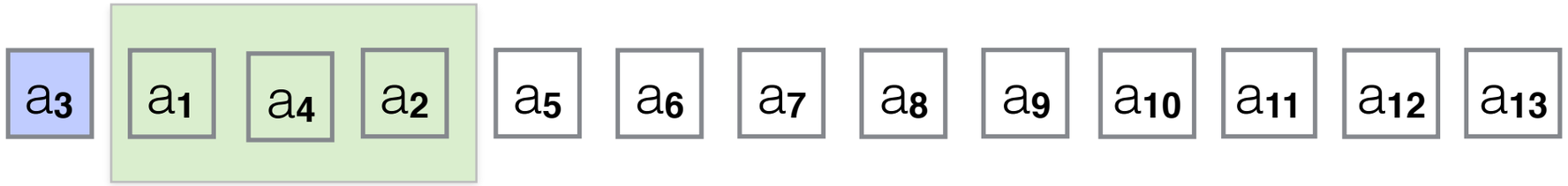
Insertion and Local Search

- Fix first free aircraft



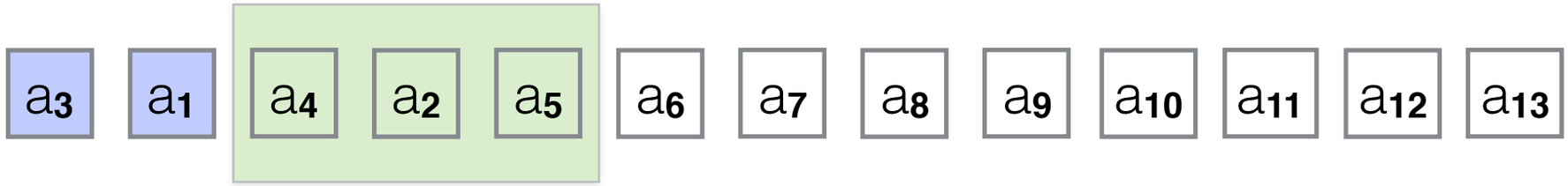
Insertion and Local Search

- Fix first free aircraft
- Find all permutations of next ‘k’ free aircraft



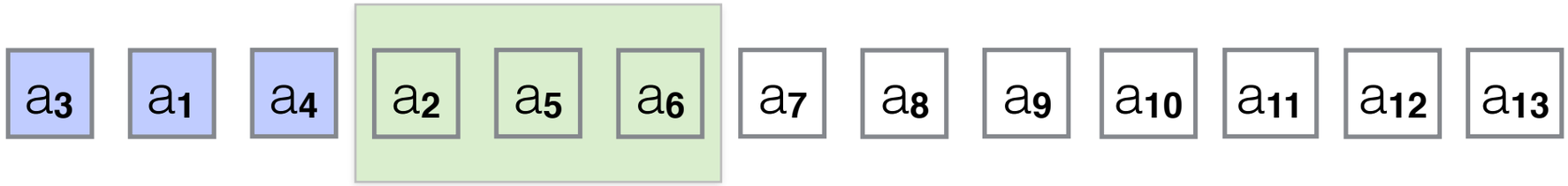
Insertion and Local Search

- Fix first free aircraft
- Find all permutations of next ‘k’ free aircraft



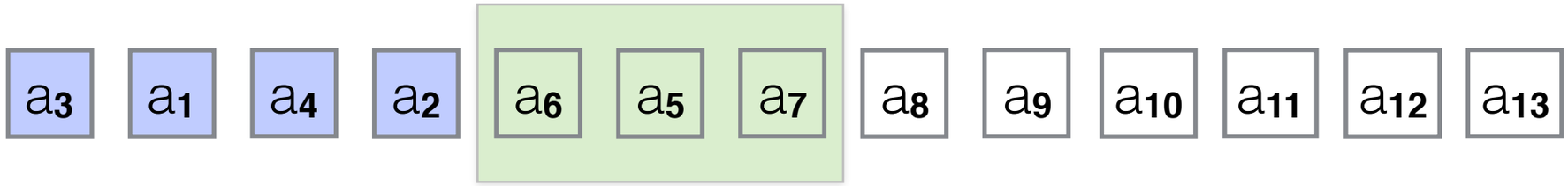
Insertion and Local Search

- Fix first free aircraft
- Find all permutations of next ‘k’ free aircraft



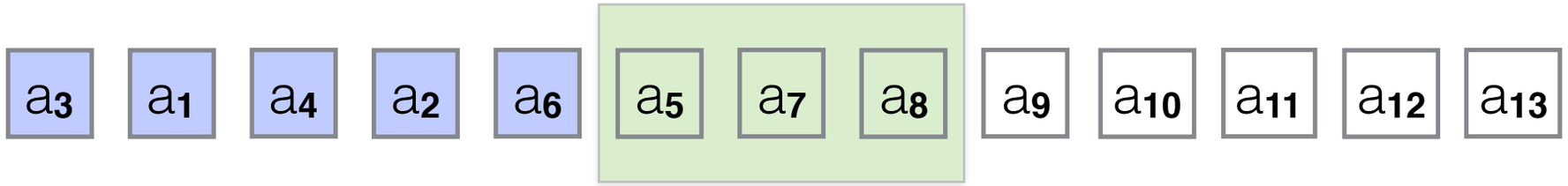
Insertion and Local Search

- Fix first free aircraft
- Find all permutations of next ‘k’ free aircraft



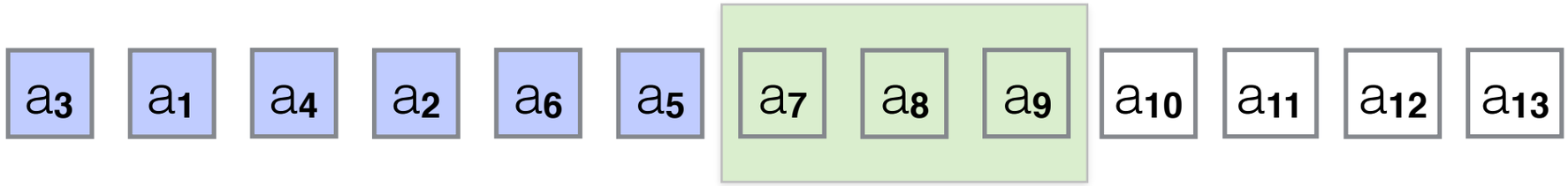
Insertion and Local Search

- Fix first free aircraft
- Find all permutations of next ‘k’ free aircraft



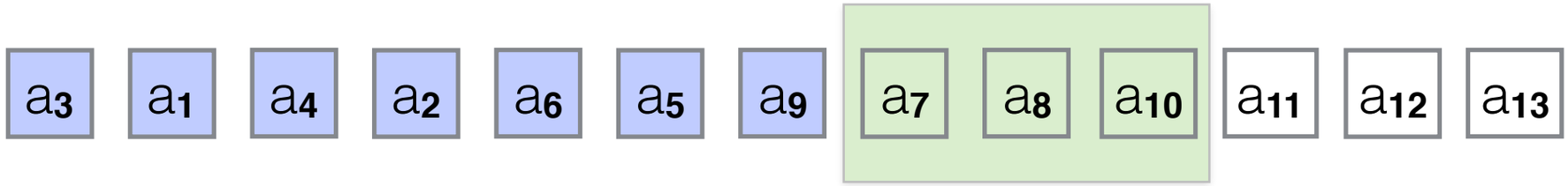
Insertion and Local Search

- Fix first free aircraft
- Find all permutations of next ‘k’ free aircraft



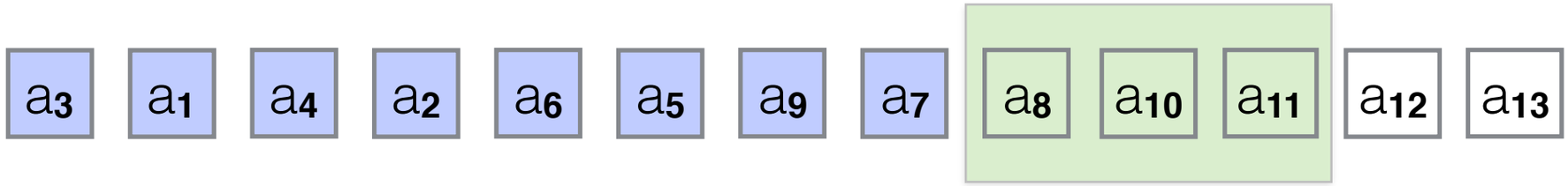
Insertion and Local Search

- Fix first free aircraft
- Find all permutations of next ‘k’ free aircraft



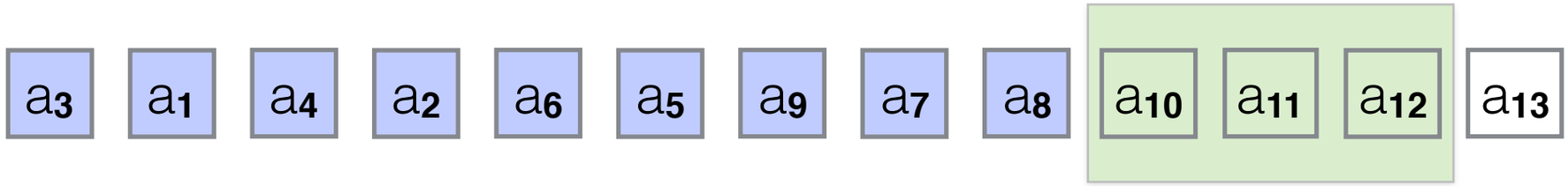
Insertion and Local Search

- Fix first free aircraft
- Find all permutations of next ‘k’ free aircraft



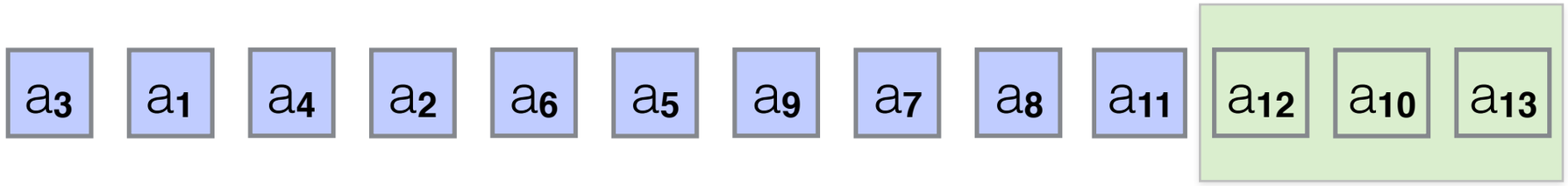
Insertion and Local Search

- Fix first free aircraft
- Find all permutations of next ‘k’ free aircraft



Insertion and Local Search

- Fix first free aircraft
- Find all permutations of next ‘k’ free aircraft



Insertion and Local Search

- Fix first free aircraft
- Find all permutations of next ‘k’ free aircraft

Insertion and Local Search

- If final sequence is different from starting sequence, repeat whole procedure (descent search)

Outline

- Overview of airport surface management
- Single runway scheduling
  - One exact algorithm
  - Two heuristics-based algorithms
  - **Simulation setup and results**
- Multiple runway scheduling
  - CLT airport layout
  - Mixed Integer Linear Program Formulation
  - Simulation setup and results

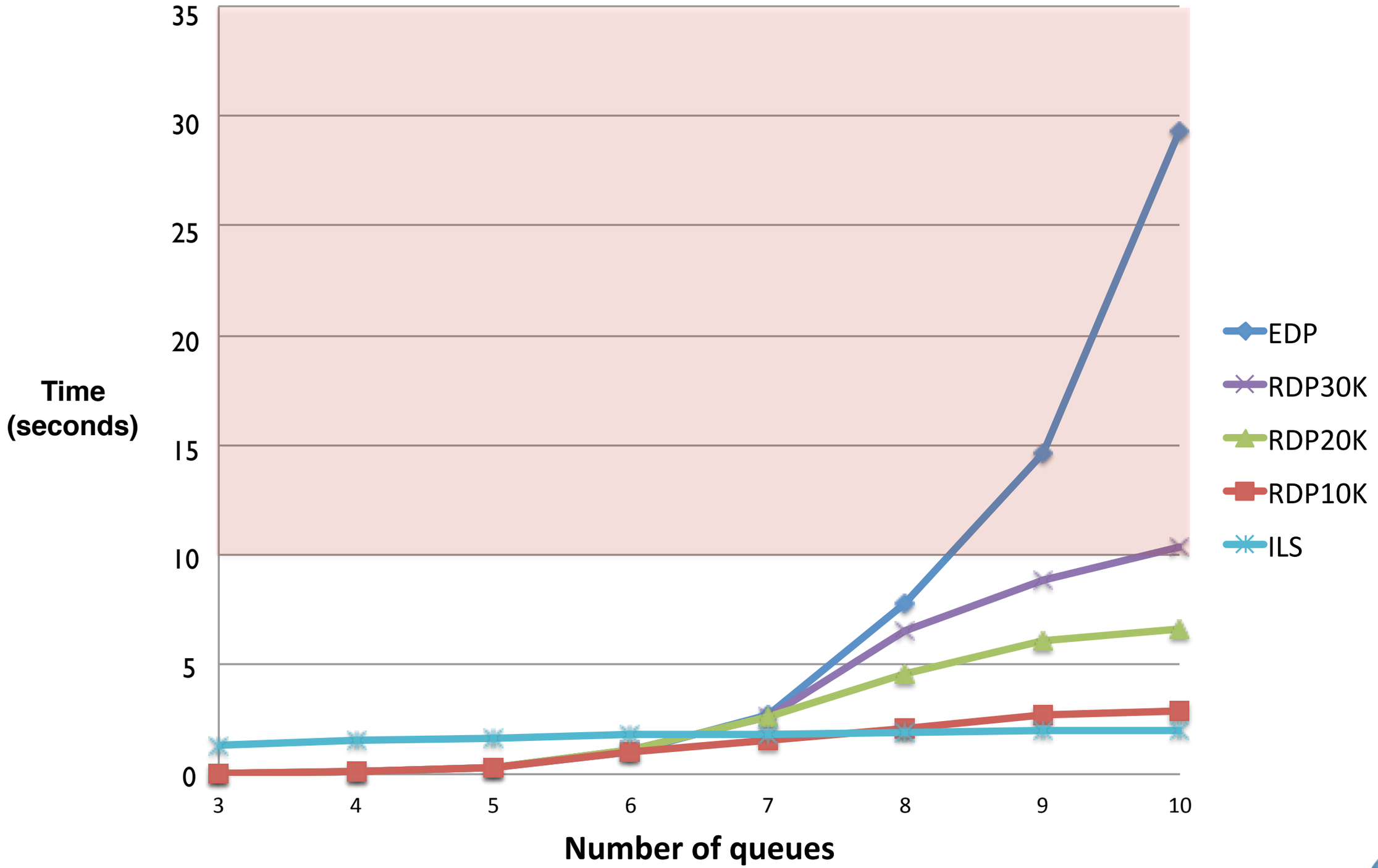
Simulation Setup

- DFW East-side in South-flow configuration
- 20 Departures and 15 runway crossings considered.
- Planning window of 15 minutes
- Earliest available times were uniformly distributed within 0-900 seconds
- 80% of type Large, 10% of type Heavy, 10% B75x
- Heading were randomly assigned to 0 or 1

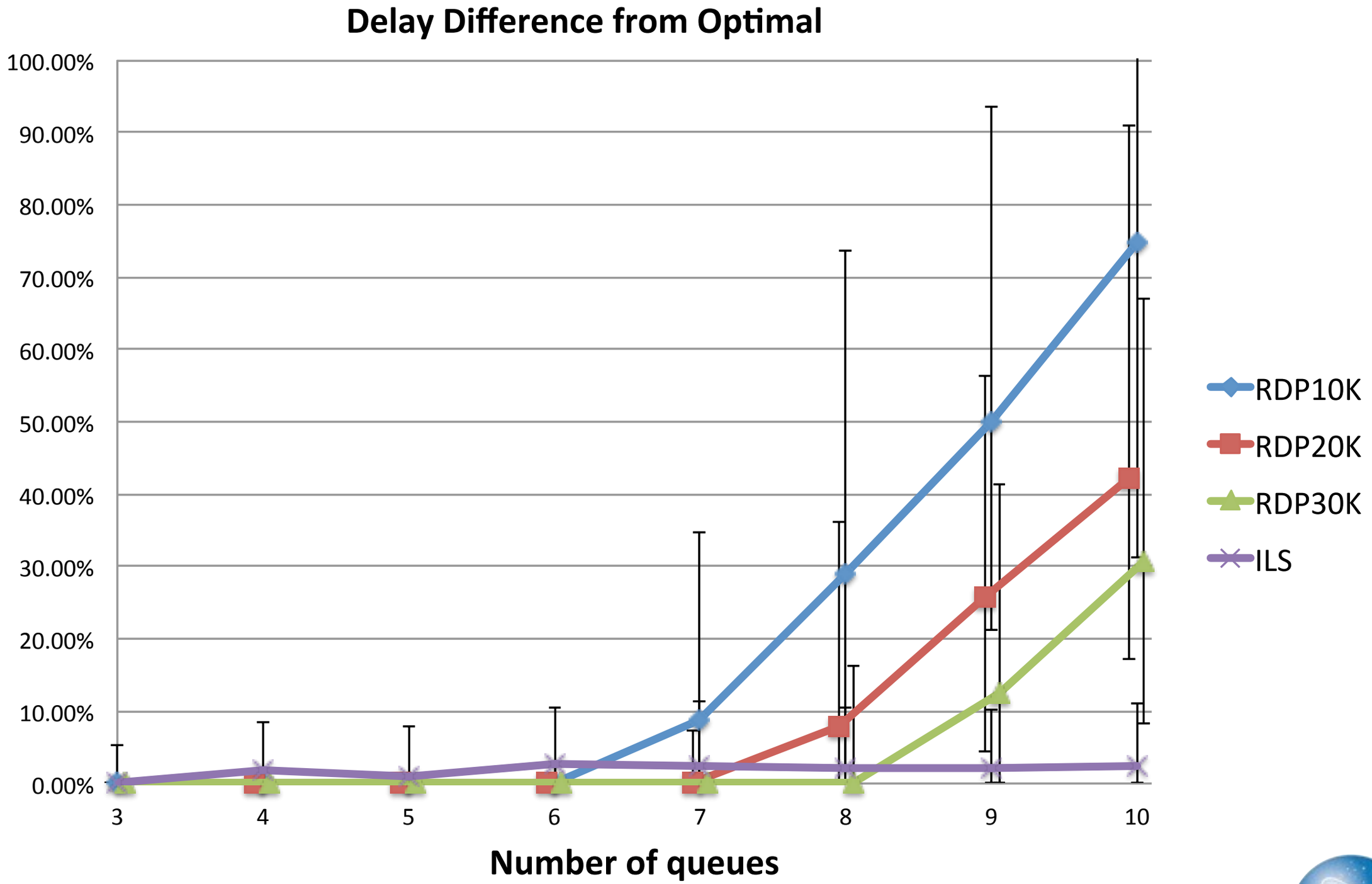
Simulation Setup

- Scenarios were generated with varying numbers of queues (from 3 to 10)
- Hundred different scenarios generated for each queue number
- Three variants of RDP algorithm: RDP10K (H=10,000), RDP20K (H=20,000), RDP30K (H=30,000)
- ILS algorithm used a value of 7 for the neighborhood parameter k

Computation times



Solution Quality



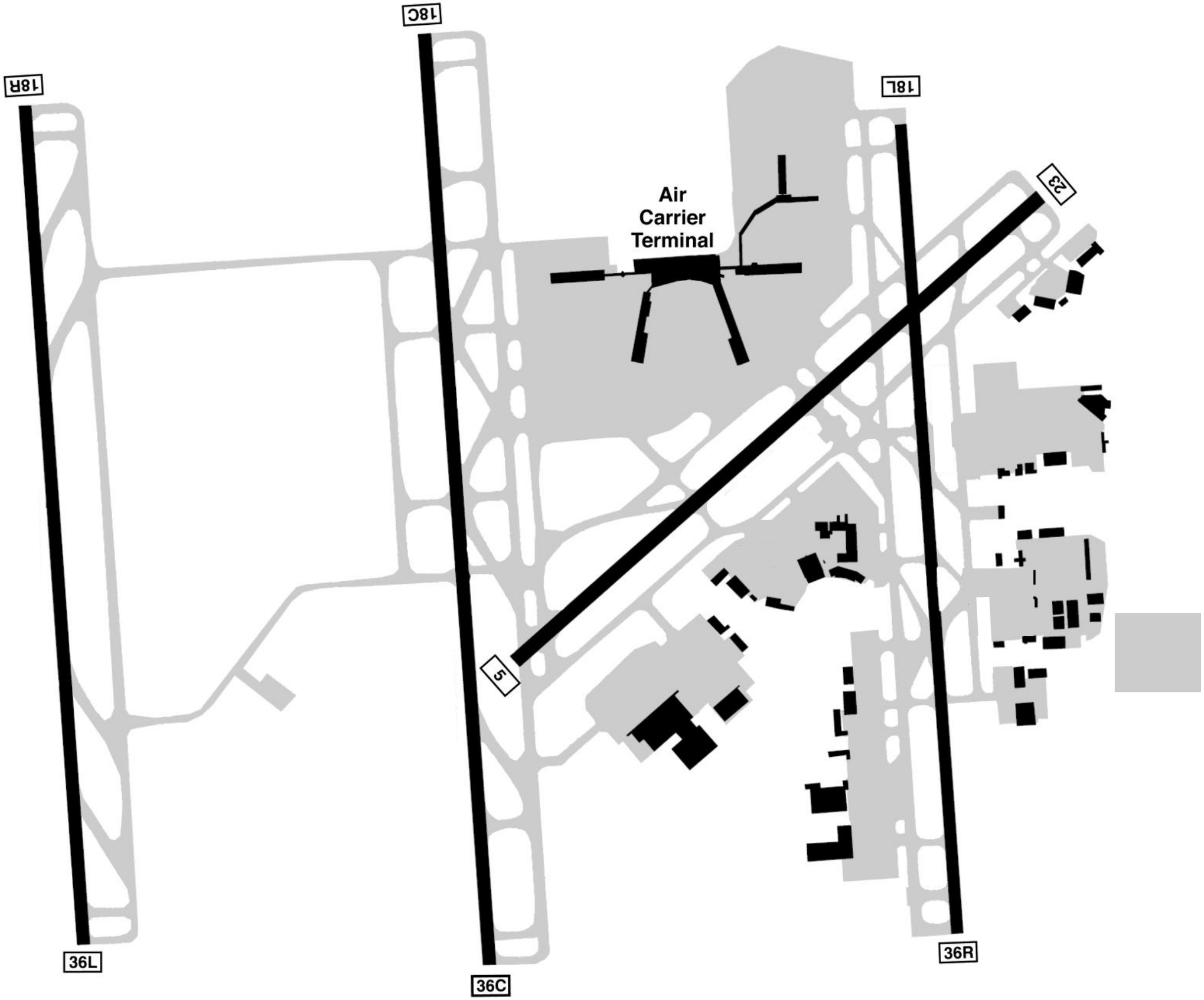
Summary: Single Runway Scheduling

- Comparative study of three algorithms:
  - Exact Dynamic Programming (EDP)
  - Restricted Dynamic Programming (RDP)
  - Insertion and Local Search (ILS)
- Simulations conducted for the east side of the Dallas/Fort Worth International Airport (DFW)
- ILS heuristics is the most suitable candidate for application in tactical surface decision support tools

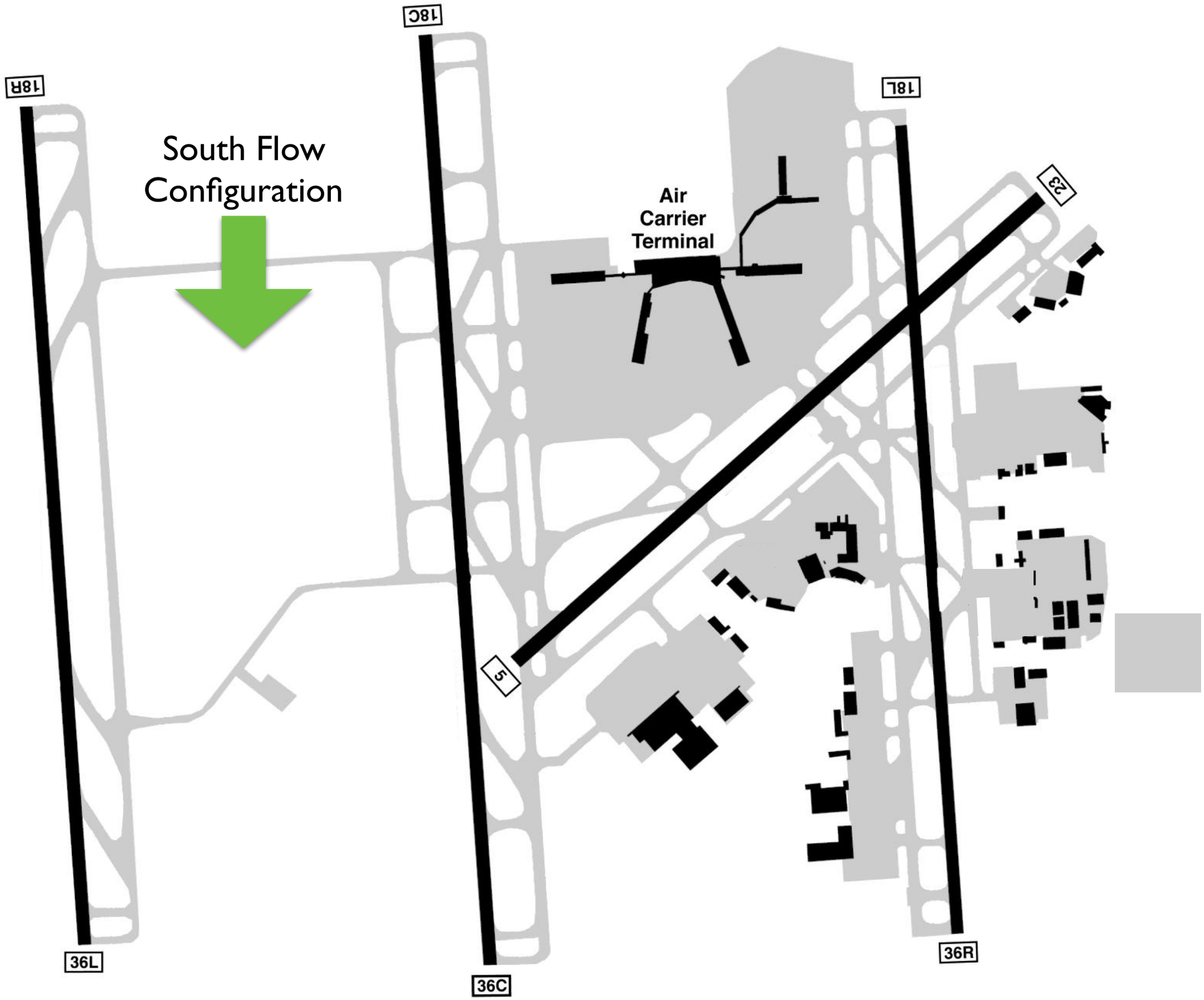
Outline

- Overview of airport surface management
- Single runway scheduling
  - One exact algorithm
  - Two heuristics-based algorithms
  - Simulation setup and results
- **Multiple runway scheduling**
  - CLT airport layout
  - Mixed Integer Linear Program Formulation
  - Simulation setup and results

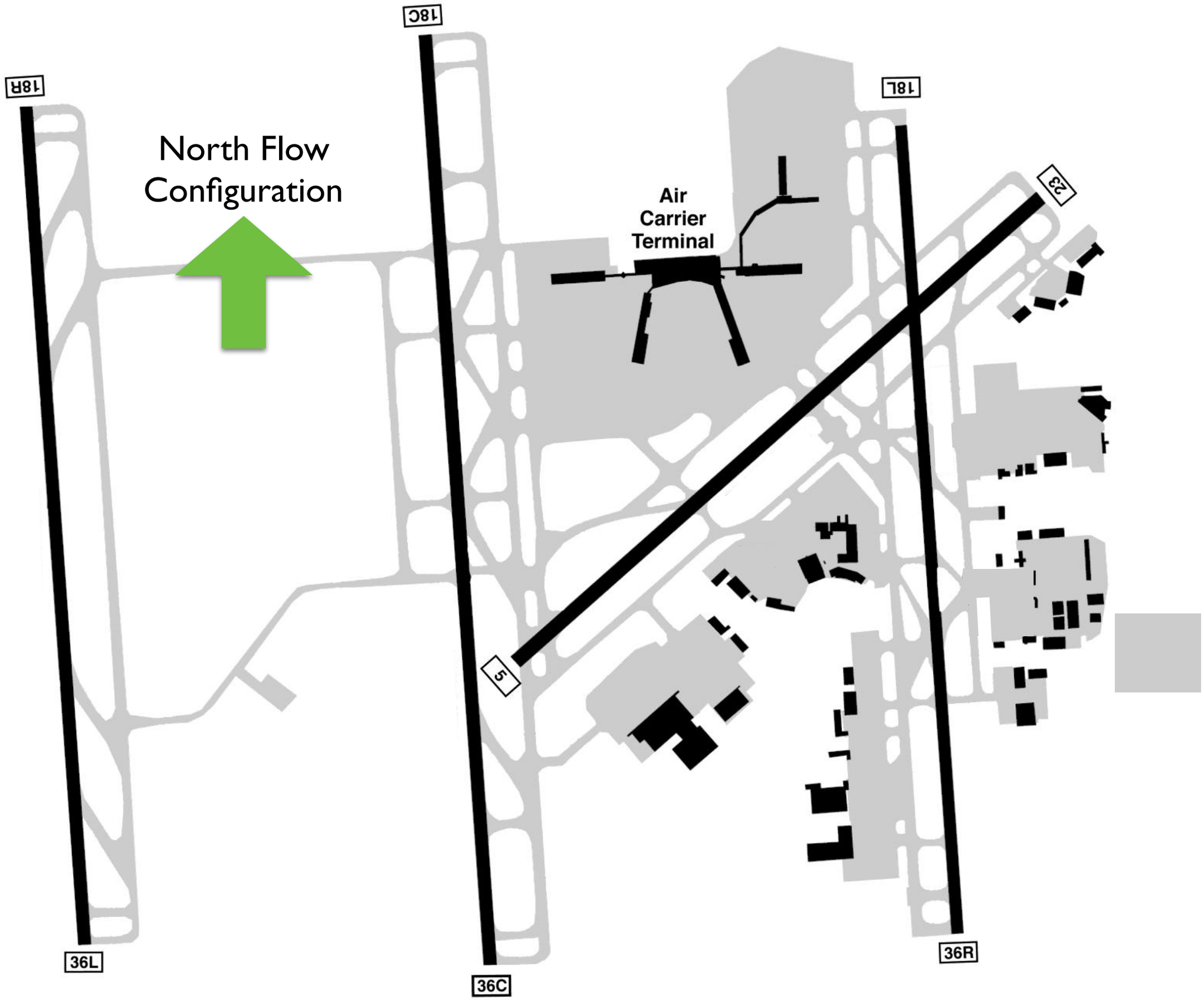
Airport Layout



Airport Layout



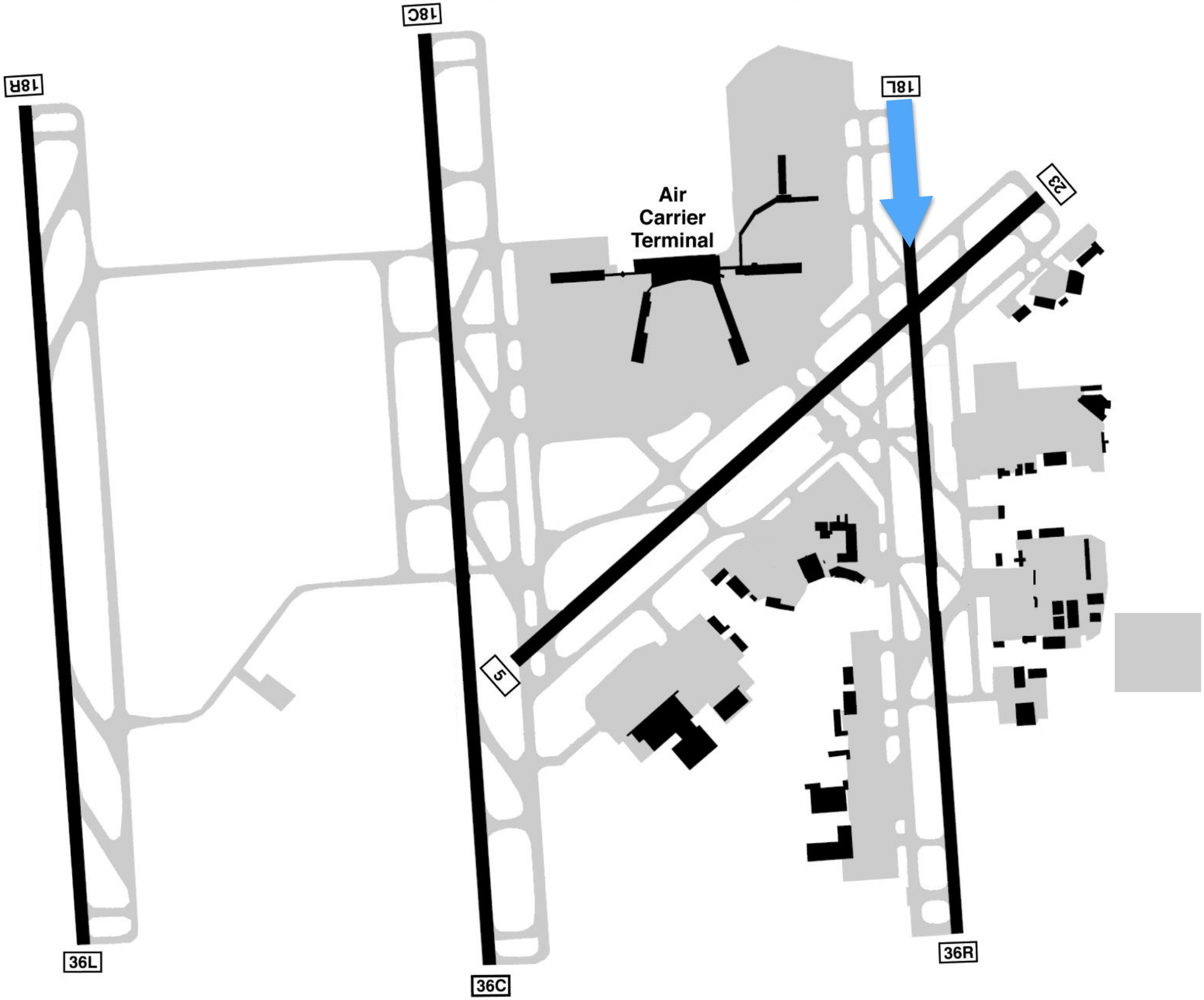
Airport Layout



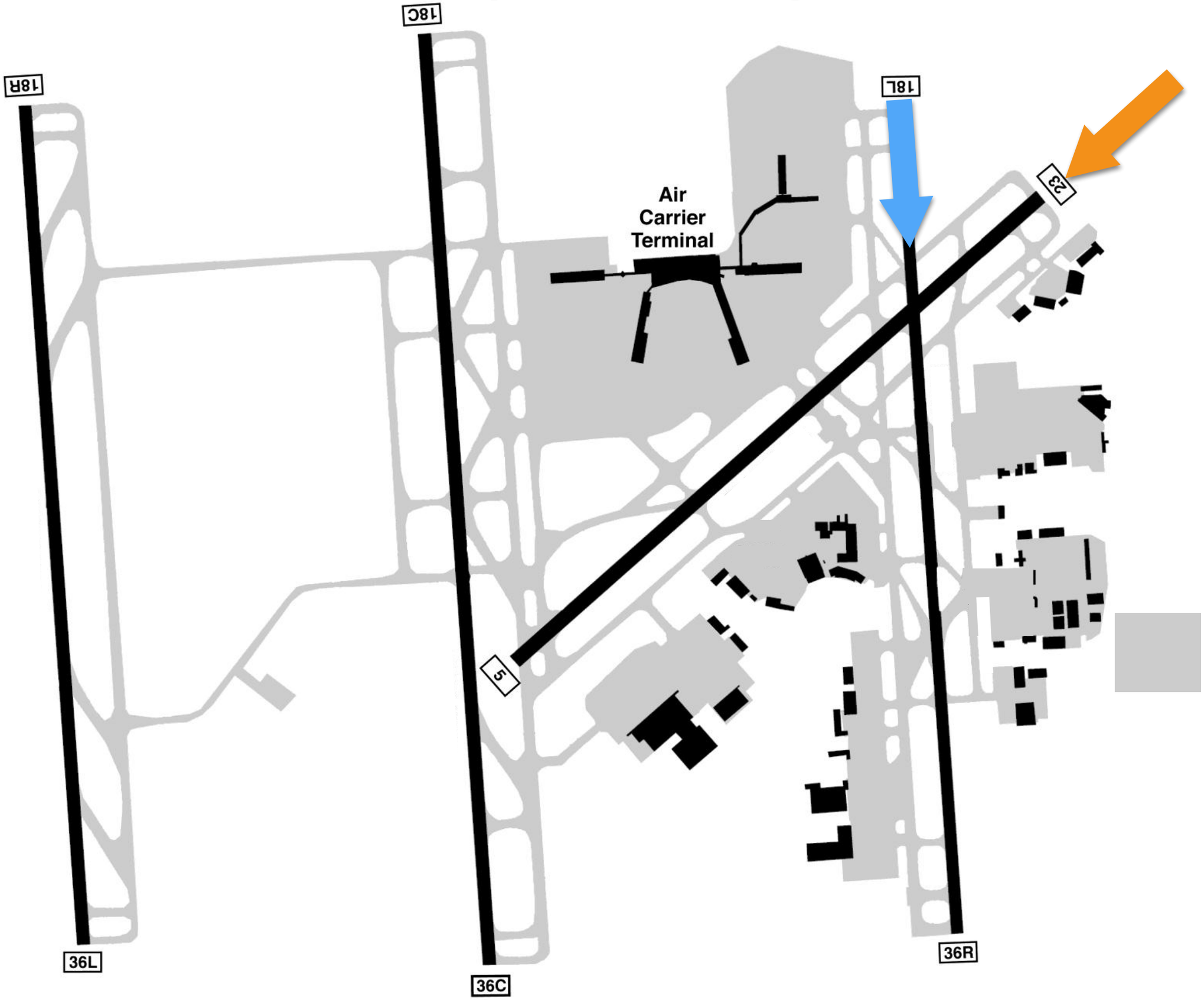
Airport Layout

Airport Layout

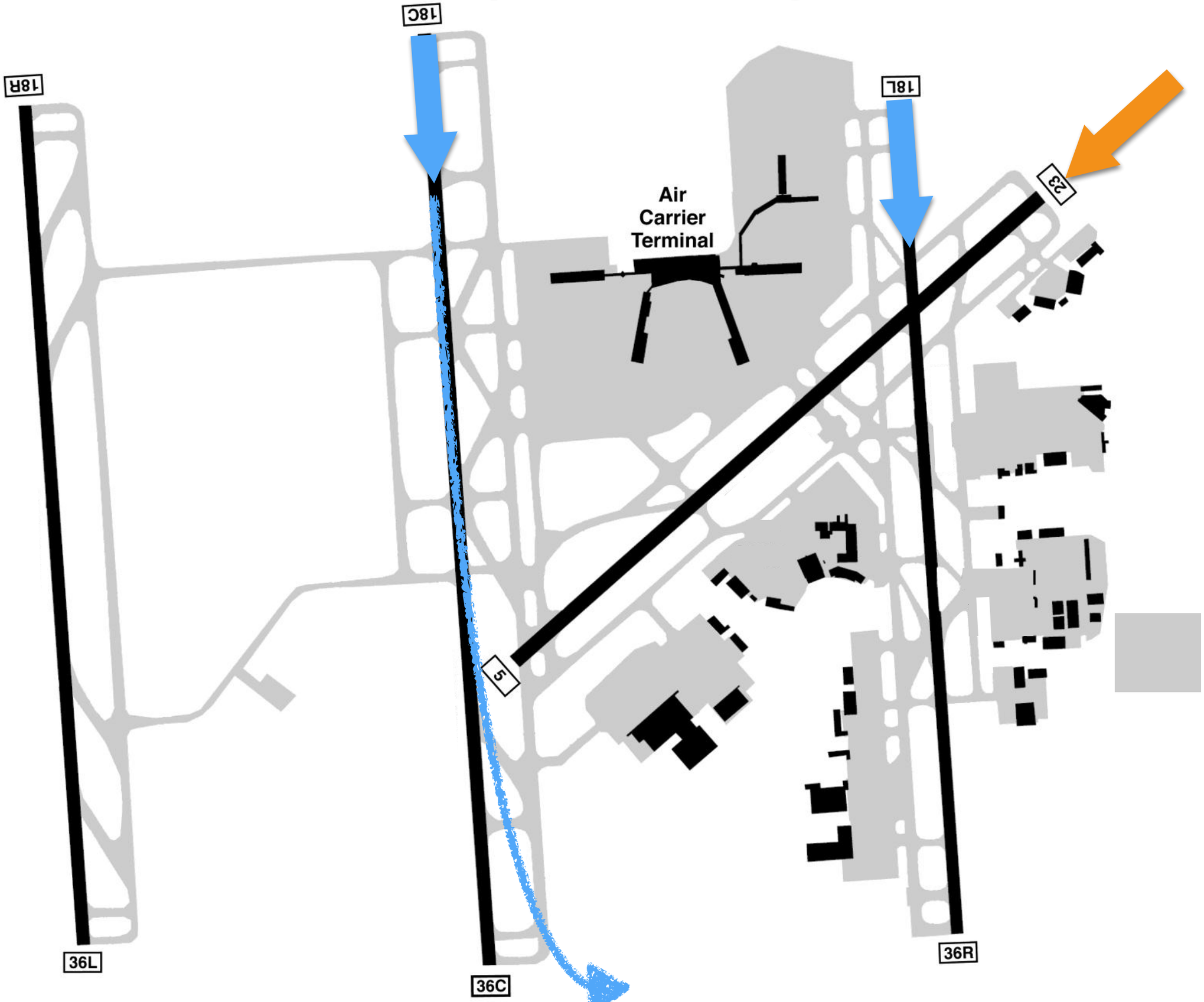
Airport Layout



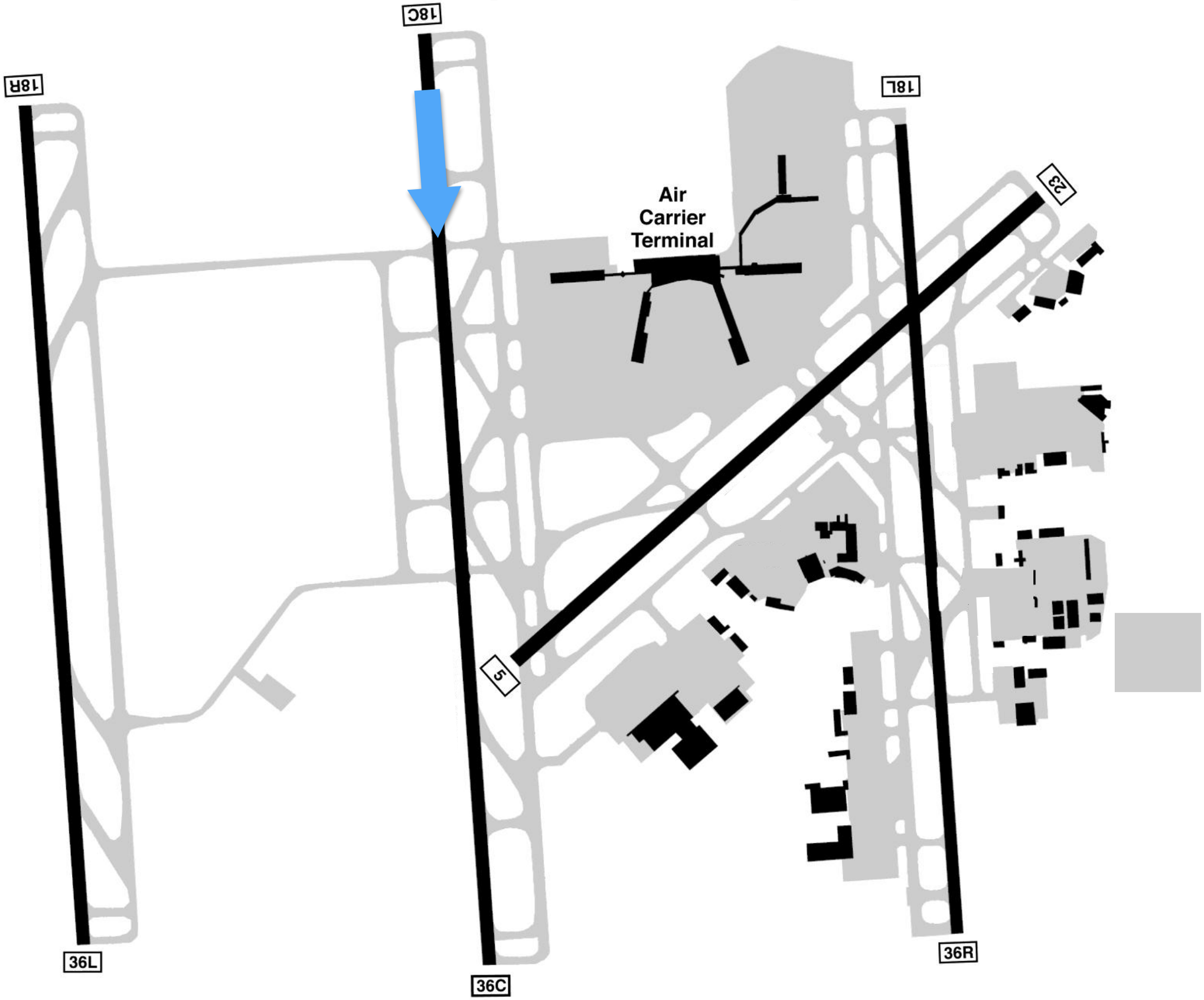
Airport Layout



Airport Layout



Airport Layout



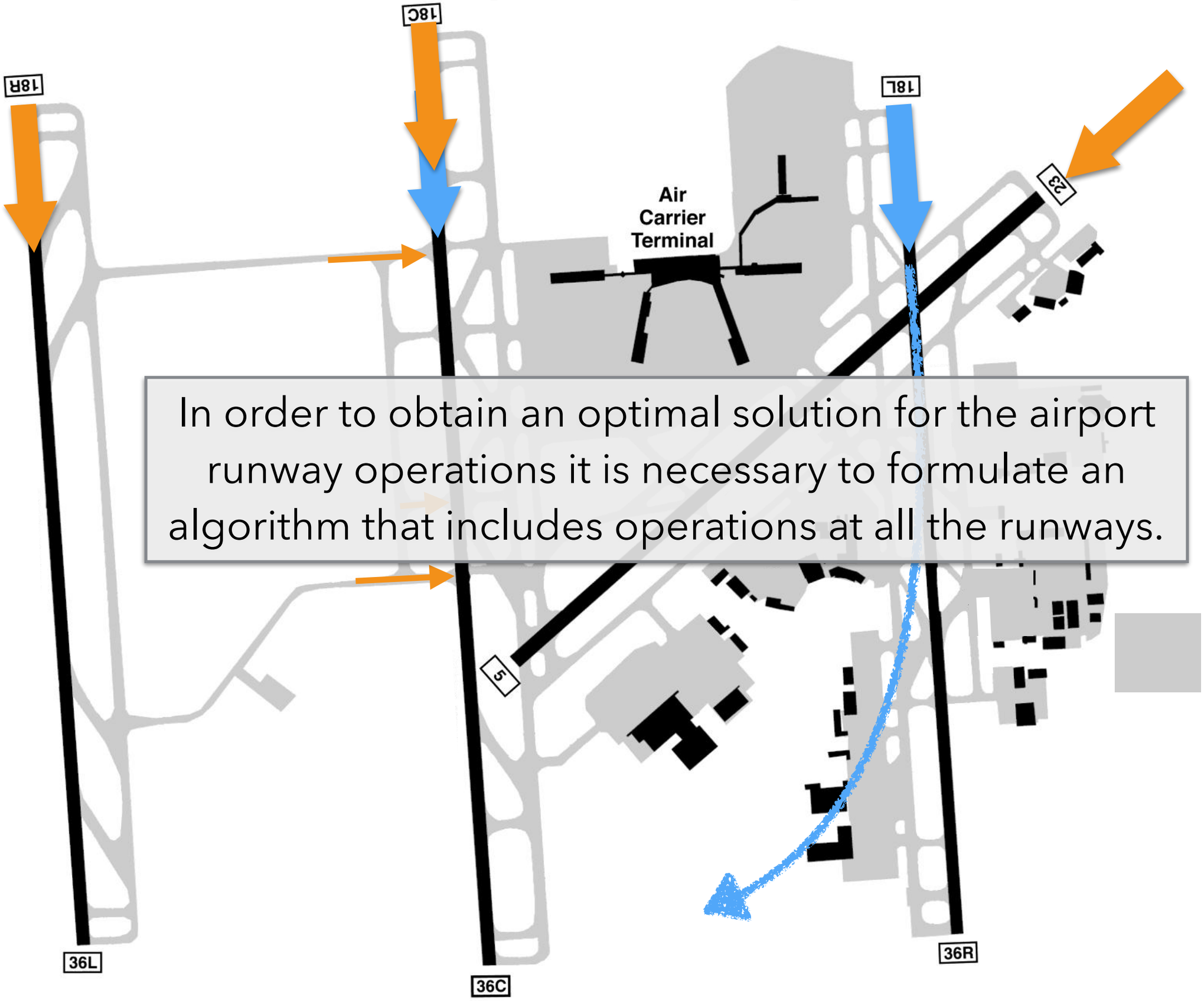
Airport Layout

Airport Layout

Airport Layout

Airport Layout

Airport Layout



In order to obtain an optimal solution for the airport runway operations it is necessary to formulate an algorithm that includes operations at all the runways.

Runway Scheduler: Inputs

- Estimated runway queue entry times (for departures)
- Estimated runway time (for arrivals)
- Spot, runway, position and fix/exit for each aircraft
- Type (weight class) of each aircraft
- Separation requirements between pair of aircraft
- Individual time-windows of intended take-off times for departing aircraft (EDCT, CFR)

Separation Requirements

- Between departures on same runway (wake vortex and RNAV separation)
- Between arrivals and departures for mixed use runway, runway crossings and converging runway operations
- Separation between departure from parallel runways going to same fix
- Separation between departures going to same constraint fix (MIT)

Separation Requirements

- Between departures on same runway (wake vortex and RNAV separation)
- Between arrivals and departures for mixed use runway, runway crossings and converging runway operations
- Separation between departure from parallel runways going to same fix
- Separation between departures going to same constraint fix (MIT)

These separations are converted to time-based separations for use in the MILP

MILP Runway Scheduler

Decision Variables

- Let t_i denote the calculated time at which the aircraft uses the runway (take-off, land or cross)

- Let $Z_{i,j}$ be a binary sequencing variable

$$Z_{i,j} = \begin{cases} 1 & \text{if aircraft } i \text{ uses runway before } j \\ 0 & \text{otherwise} \end{cases}$$

MILP Runway Scheduler

Objective Function

- System Delay – cumulative waiting time of all aircraft
- Let α_i be a earliest available time

- System Delay:

$$\min \sum_{i \in F} (t_i - \alpha_i)$$

MILP Formulation

MILP Formulation

- Linear ordering constraints : $Z_{i,j} + Z_{j,i} = 1$

MILP Formulation

- Linear ordering constraints : $Z_{i,j} + Z_{j,i} = 1$
- Runway use after earliest time : $t_i \geq \alpha_i$

MILP Formulation

- Linear ordering constraints : $Z_{i,j} + Z_{j,i} = 1$
- Runway use after earliest time : $t_i \geq \alpha_i$
- Arrival landing time cannot be changed : $t_i \leq \alpha_i + \delta$

MILP Formulation

- Linear ordering constraints : $Z_{i,j} + Z_{j,i} = 1$
- Runway use after earliest time : $t_i \geq \alpha_i$
- Arrival landing time cannot be changed : $t_i \leq \alpha_i + \delta$
- Time-Window constraints : $TMI_L \leq t_i \leq TMI_H$

MILP Formulation

- Linear ordering constraints : $Z_{i,j} + Z_{j,i} = 1$
- Runway use after earliest time : $t_i \geq \alpha_i$
- Arrival landing time cannot be changed : $t_i \leq \alpha_i + \delta$
- Time-Window constraints : $TMI_L \leq t_i \leq TMI_H$
- Separation requirements : $Z_{i,j}(t_j - t_i - \Delta_{i,j}) \geq 0$

MILP Formulation

- Linear ordering constraints : $Z_{i,j} + Z_{j,i} = 1$
- Runway use after earliest time : $t_i \geq \alpha_i$
- Arrival landing time cannot be changed : $t_i \leq \alpha_i + \delta$
- Time-Window constraints : $TMI_L \leq t_i \leq TMI_H$
- Separation requirements : $Z_{i,j}(t_j - t_i - \Delta_{i,j}) \geq 0$
- FCFS constraints on crossing aircraft : $Z_{i,j} = 1, \text{ if } \alpha_i < \alpha_j$

MILP Formulation

- Linear ordering constraints : $Z_{i,j} + Z_{j,i} = 1$
- Runway use after earliest time : $t_i \geq \alpha_i$
- Arrival landing time cannot be changed : $t_i \leq \alpha_i + \delta$
- Time-Window constraints : $TMI_L \leq t_i \leq TMI_H$
- Separation requirements : $Z_{i,j}(t_j - t_i - \Delta_{i,j}) \geq 0$
- FCFS constraints on crossing aircraft : $Z_{i,j} = 1, \text{ if } \alpha_i < \alpha_j$
- FCFS constraints on MIT aircraft : $Z_{i,j} = 1, \text{ if } \alpha_i < \alpha_j$

MILP Formulation

- Linear ordering constraints : $Z_{i,j} + Z_{j,i} = 1$
- Runway use after earliest time : $t_i \geq \alpha_i$
- Arrival landing time cannot be changed : $t_i \leq \alpha_i + \delta$
- Time-Window constraints : $TMI_L \leq t_i \leq TMI_H$
- Separation requirements : $Z_{i,j}(t_j - t_i - \Delta_{i,j}) \geq 0$
- FCFS constraints on crossing aircraft : $Z_{i,j} = 1, \text{ if } \alpha_i < \alpha_j$
- FCFS constraints on MIT aircraft : $Z_{i,j} = 1, \text{ if } \alpha_i < \alpha_j$
- Constrained Position Shift (CPS) constraint on sequence of departures only

Simulation Setup

- Mixed operation runway with arrivals, departures and crossing traffic
- Another stream of arrivals was modeled to simulate converging runway operations
- Planning window of 15 minutes
- Number of aircraft in the scenarios was varied from 10 to 35 in increments of 5
- Hundred different scenarios generated for each aircraft count

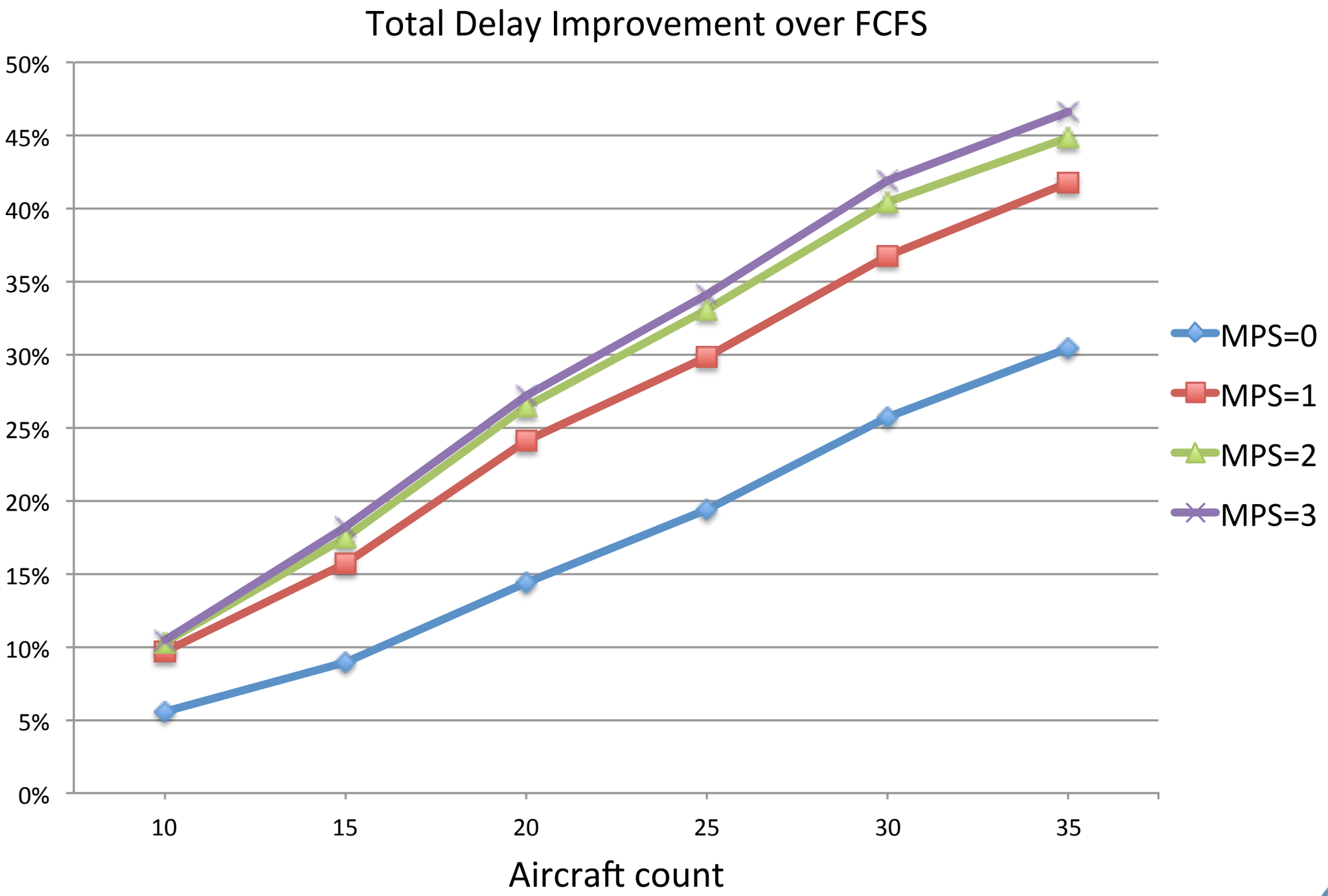
Simulation Setup

- Earliest available times were uniformly distributed within 0-900 seconds
- Sixty percent of the traffic was chosen to be departures, 20% arrivals and 20% crossing aircraft
- 80% of type Large, 10% of type Heavy, 10% B75x
- Departure fix assigned randomly from 6 discrete choices

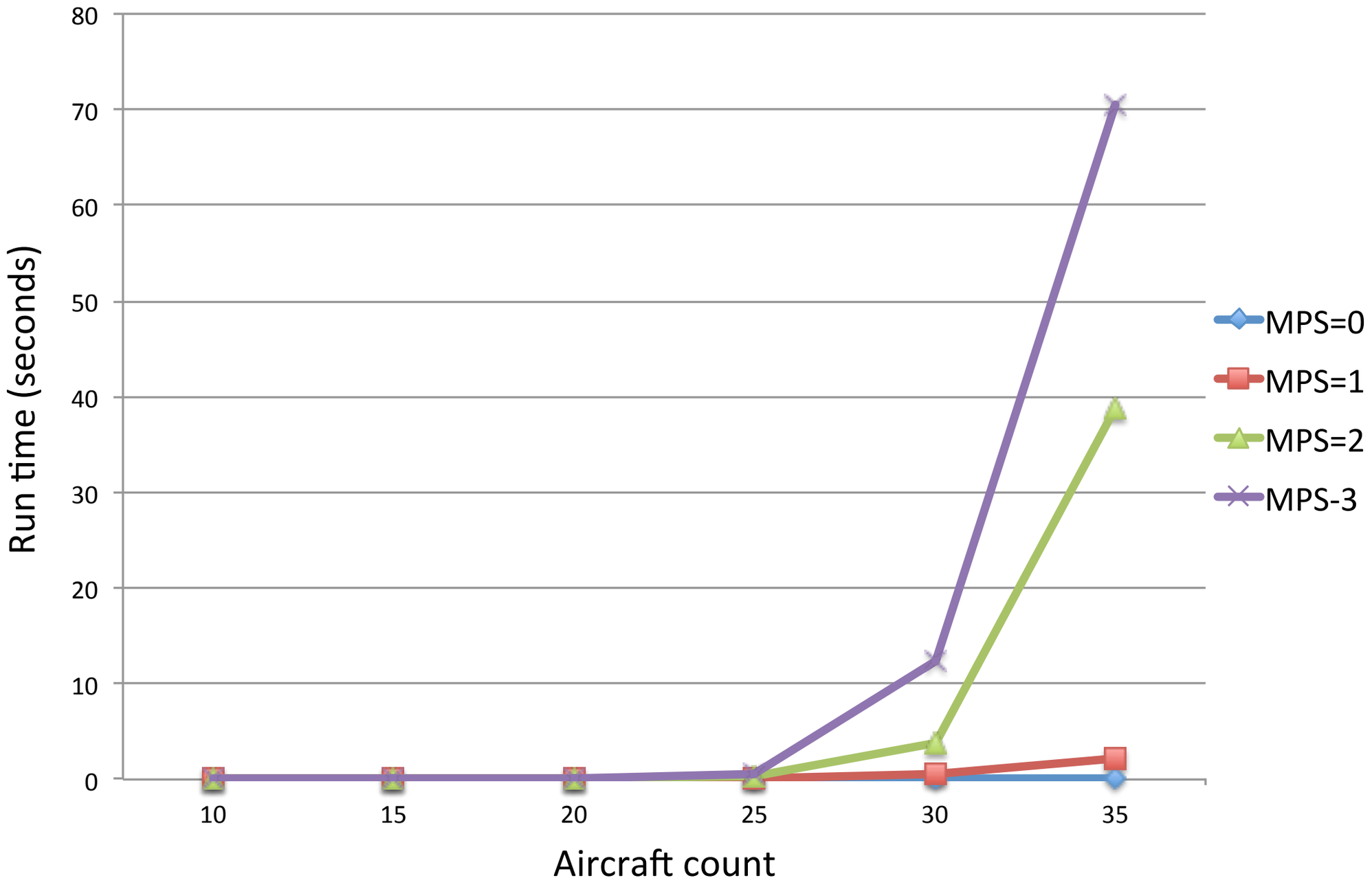
Simulation Setup

- Earliest available times were uniformly distributed within 0-900 seconds
- Sixty percent of the traffic was chosen to be departures, 20% arrivals and 20% crossing aircraft
- 80% of type Large, 10% of type Heavy, 10% B75x
- Departure fix assigned randomly from 6 discrete choices
- MILP formulation is compared with a FCFS to examine the benefits of the proposed algorithm
- The MILP is solved using Gurobi

Solution Quality



Computation times



Summary: Multiple Runway Scheduling

- A MILP formulation for multiple runway scheduling
- 30% average improvement in total delay over FCFS
- Maximum position shift (MPS) parameter value of 2 is a good trade-off between solution quality and computation times
- MPS value of 0 and 1 are also good for cases with limited computational resources


NASA