Decision aid methodologies in transportation

Lecture 7: More Applications

Prem Kumar

prem.viswanathan@epfl.ch

Transport and Mobility Laboratory

Summary

- We learnt about the different scheduling models
- We also learnt about demand-supply interactions in the form of revenue management concepts
- We learnt to mimic expectations and use solver with Spreadsheets
- We have learnt about OR applications in maritime and learnt to use mathematical solvers
- Today, we will see some more applications
- We will work more with MATHPROG in the lab

Applications in Railways: Locomotive Assignment

- Basic Inputs
  - Train Schedule over a period of planning horizon
  - A set of locomotives, their current locations and properties
- Output
  - Assignment schedule of locomotives to trains
- Constraints
  - Locomotive maintenance
  - Tonnage and HP requirement of train
  - Several other constraints
- Objective
  - Cost minimization

Locomotive Assignment: Some Features

- A train is typically assigned a group of multiple locomotives called a **consist** that usually travels together
- Each train has a different **HP** and **Tonnage requirement** that depends on the number of cars attached
- Locomotives can either pull trains actively or **deadhead** on them.
- Locomotives can also **light travel**.
- Trains need not have the same daily schedule.

Locomotive Assignment: Mathematical Model

- Decision Variables
  - Locomotive-Train assignment schedule
  - Active locomotives
  - Deadhead locomotives
  - Light travel locomotives
- Parameters
  - Locomotive availability, maintenance schedule and features
  - Train schedule / time-table and train features
  - Infrastructure features for sections and yards

Locomotive Assignment: Hard Constraints

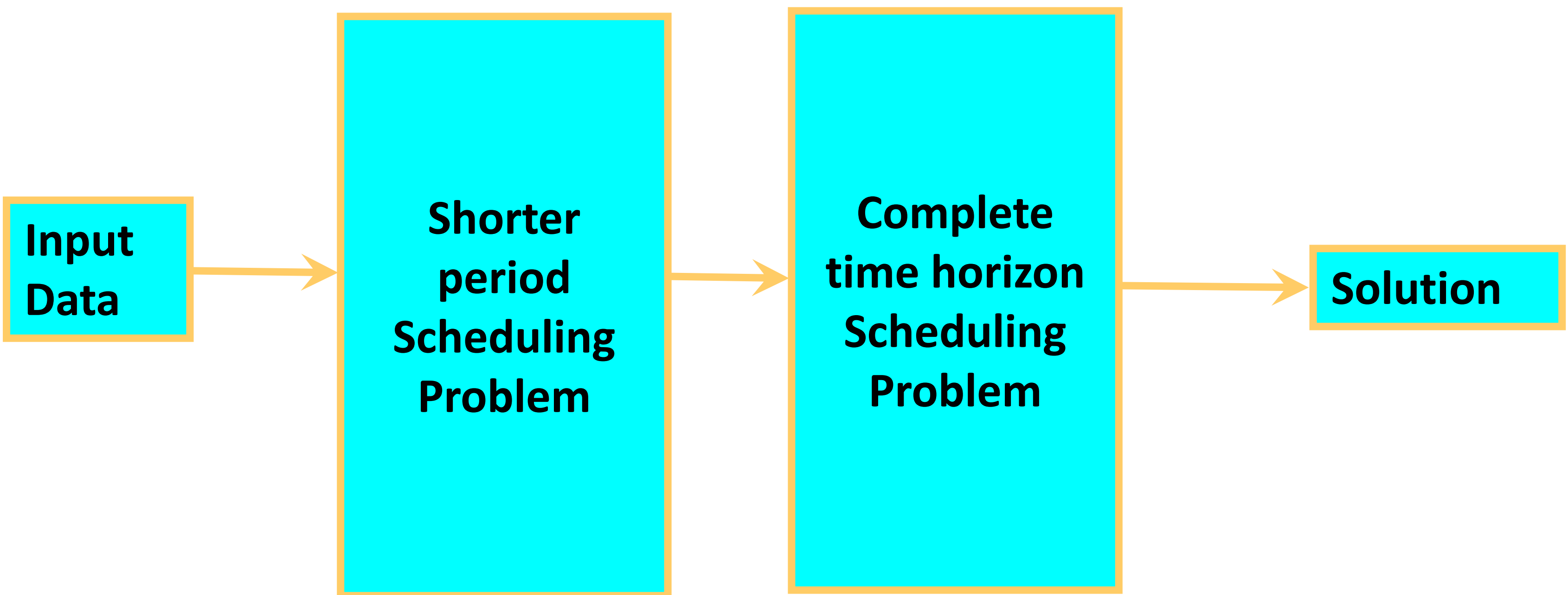
- Horsepower requirements
- Tonnage requirements
- Fleet size limitations
- Consistency of the assignments
- Locomotive availability at yards and sections
- Repeatability of the solution
- Solution robustness and recoverability

Locomotive Assignment: Literature

Reference: Ahuja et al (2003, 2005)

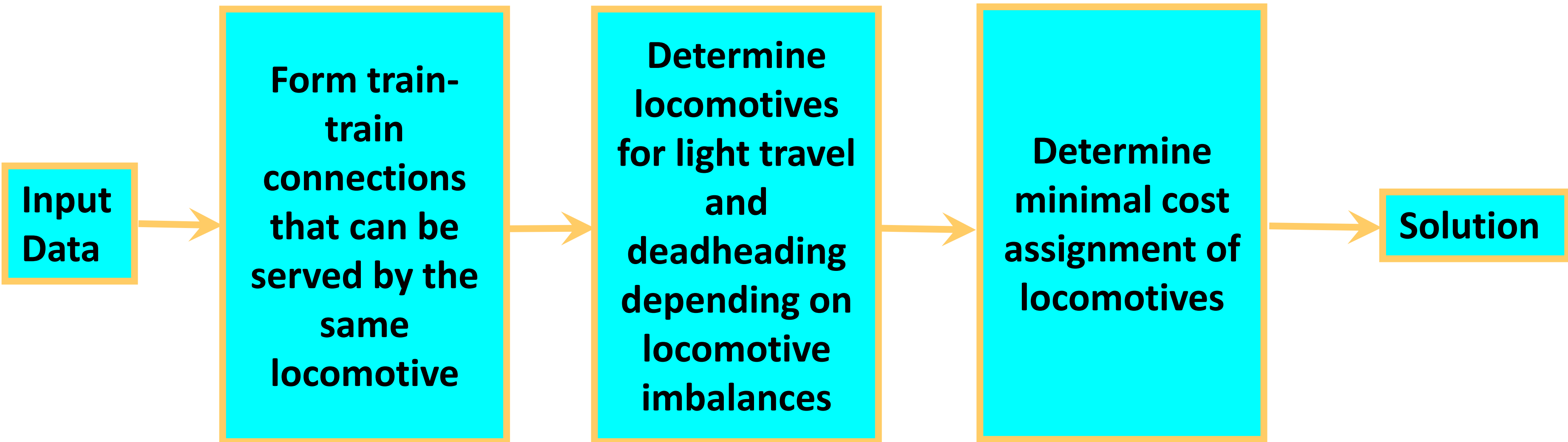
- Number of trains per week: over 3,500
- Number of locomotives: over 2,000
- Number of locomotive types: 5
- Size of the integer programming problem:
  - Number of integer variables: 200,000
  - Number of constraints: 67,000

Locomotive Assignment: Solution Methodology



Two-stage optimization allows us to reduce the problem size substantially while giving an opportunity to maintain consistency

Locomotive Assignment: Solution Methodology



- **Determine the three sets of decision variables using a sequential process.**

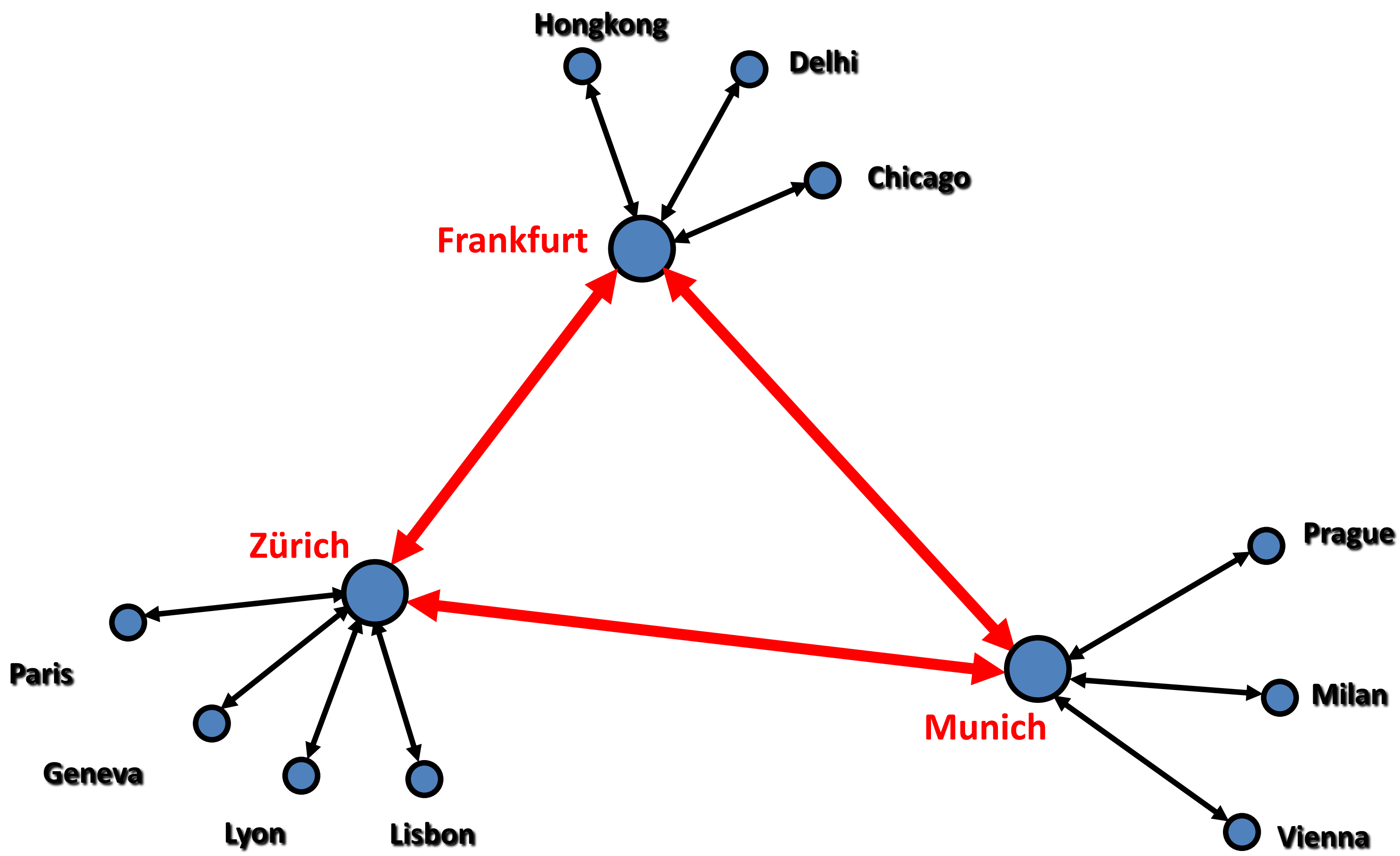
Locomotive Assignment: Model Results

- Increase in efficiency by about 15%.
- Railroad company felt that they could save about 50-100 locomotives by the use of this model.

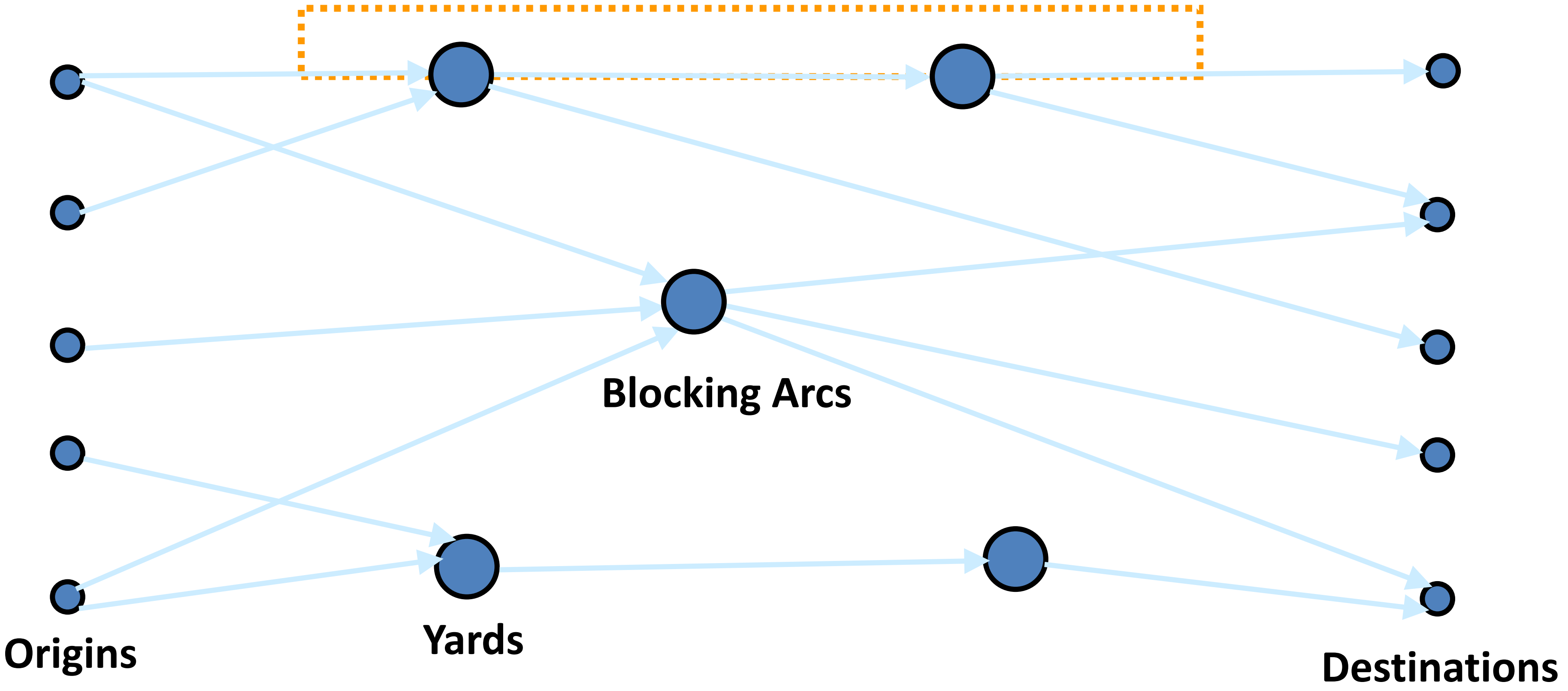
Railroad Blocking Problem

- Problem:
  - Origin-Destination of shipments given
  - Each shipment contains different number of cars
  - Train routes and time table known
  - Capacity of the network and trains known
- Magnitude:
  - Thousands of trains per month
  - 50,000 – 100,000 shipments with an average of 10 cars (Ahuja et al)
- Design the network on which commodities flow

Comparison with Airline Schedule Design



Railroad Blocking Problem



Reference: Ahuja et al: Railroad Blocking Problems

Railroad Blocking Problem: Model

- Decision Variables:
  - Blocking arcs to a yard with origin (or destination) selected, or not
  - Route followed by the shipments along the blocking arcs
- Constraints:
  - Number of blocking arcs at each node
  - Volume of cars passing through each node
  - Capacity of the network and train schedule
- Objective Function:
  - Minimize the number of intermediate handling and the sum of distance travelled (different objectives can be weighted)

Railroad Blocking Problem: Problem Scale

- Network size:
  - 1,000 origins
  - 2,000 destinations
  - 300 yards
- Number of network design variables:
  - 1,000x300 + 300x300 + 300x2,000 ≈ 1 million
- Number of flow variables:
  - 50,000 commodities flowing over 1 million potential arcs

Reference: Ahuja et al: Railroad Blocking Problems

Railroad Blocking Problem: Complexity

- Network design problems are complex for many reasons. Apart from the large number of variables, there can be several competing solutions with the same value of the objective function
- Problems with only a few hundred network design variables can be solved to optimality
- Railroads want a near-optimal and implementable solution within a few hours of computational time.

Railroad Blocking Problem: Solution Approach

- Integer Programming Based Methods
  - Slow and impractical for large scale instances

- Network Optimization Methods
  - Start with a feasible solutions
  - Gradually improve the solution – one node at a time

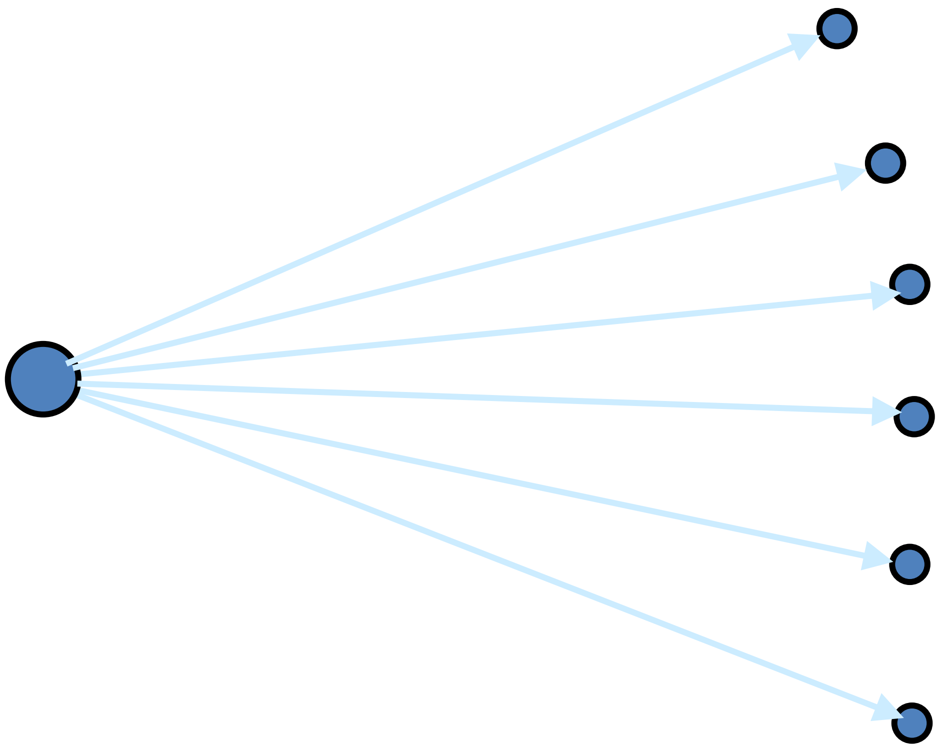
Reference: Ahuja et al: Railroad Blocking Problems

Railroad Blocking Problem: Solution Approach

- Start with a feasible solution of the blocking problem
- Optimize the blocking solution at only one node (leaving the solution at other nodes unchanged) and reroute shipments
- Repeat as long as there are improvements.

Reference: Ahuja et al: Railroad Blocking Problems

Railroad Blocking Problem: Solution Approach



Out of about 3,000 arcs emanating from a node, select 50 arcs and redirect up to 50,000 shipments to minimize the cost of flow.

Problem instance could be solved for one node using CPLEX in one hour.

Railroad Blocking Problem: Future

- This is one of the ongoing research open problems that is currently being tackled by the railroad industry
- Of course there are many such interesting problems in railways and we could give example of only two in this lecture

Airport Gate Assignment: Objectives

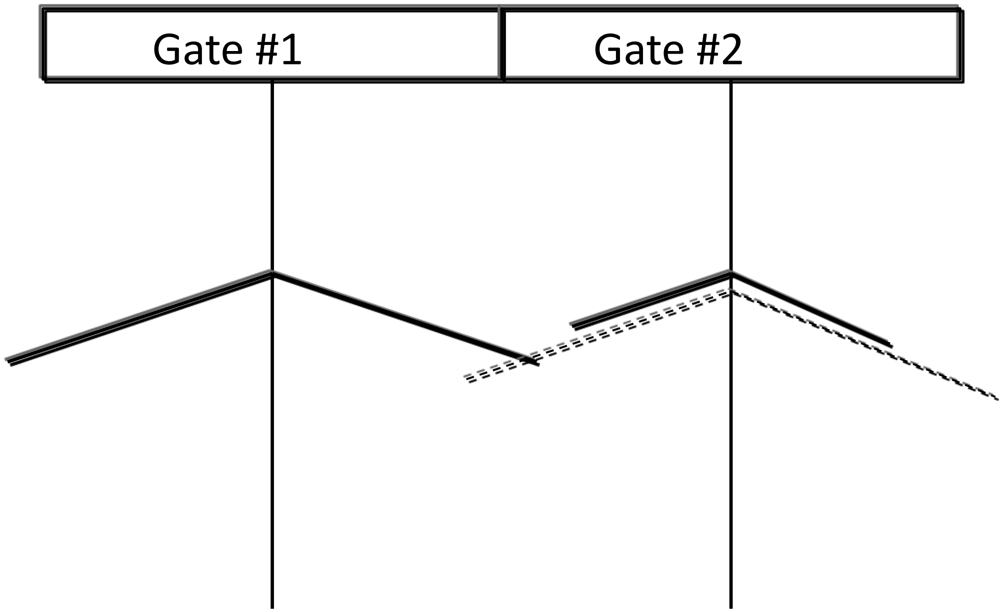
- Given a set of flight arrivals and departures at a major hub airport, what is the **best* assignment of these incoming flights to airport gates so that all flights are gated?
- Gating constraints such as adjacent gate, LIFO gates, gate rest time, towing, push back time and PS gates are applicable

Airport Gate Assignment: Problem Instance

- One of the largest in the world
- Over 1200 flights daily
- Over 25 different fleet types handled
- 60 gates and several landing bays
- Around 50,000 connecting passengers

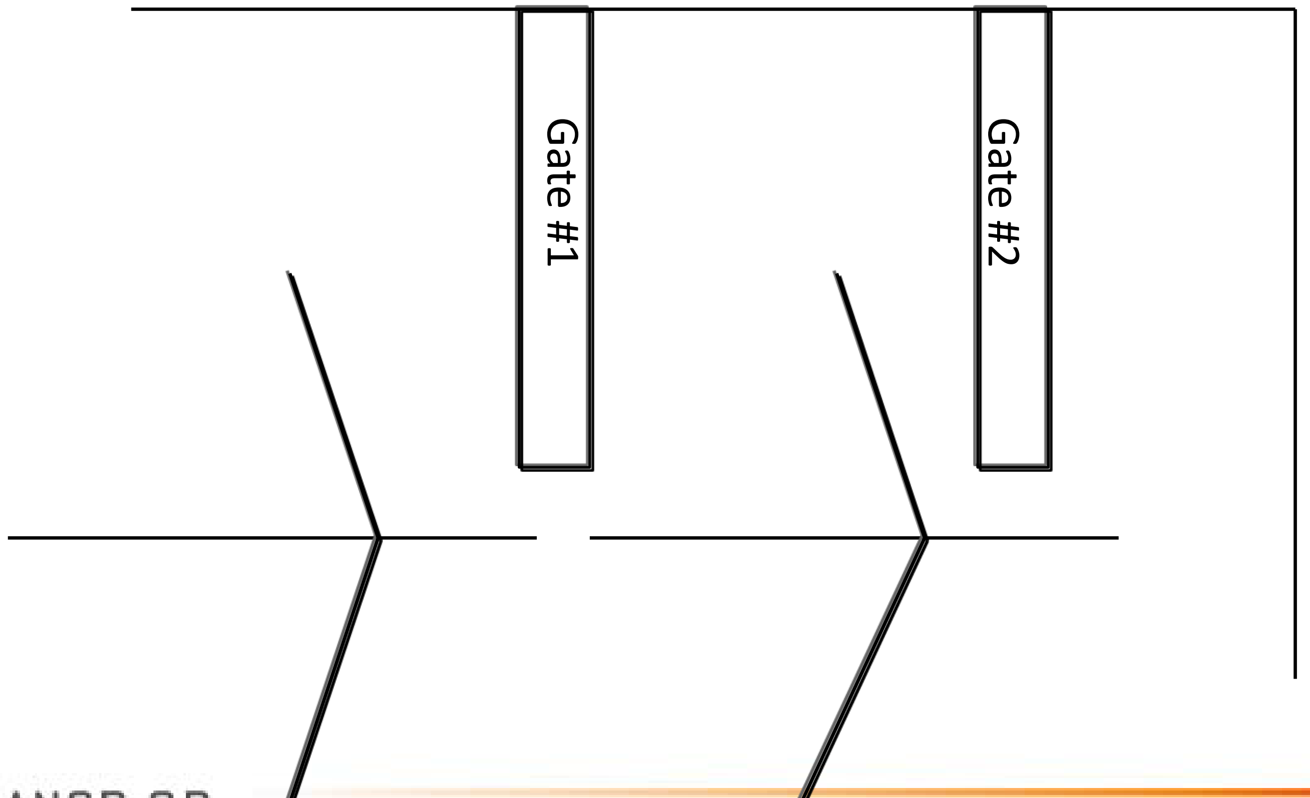
Terminology

- Adjacent Gates: Two physically adjacent gates such that when one gate has a wide bodied aircraft parked on it, the other gate cannot accommodate another wide body



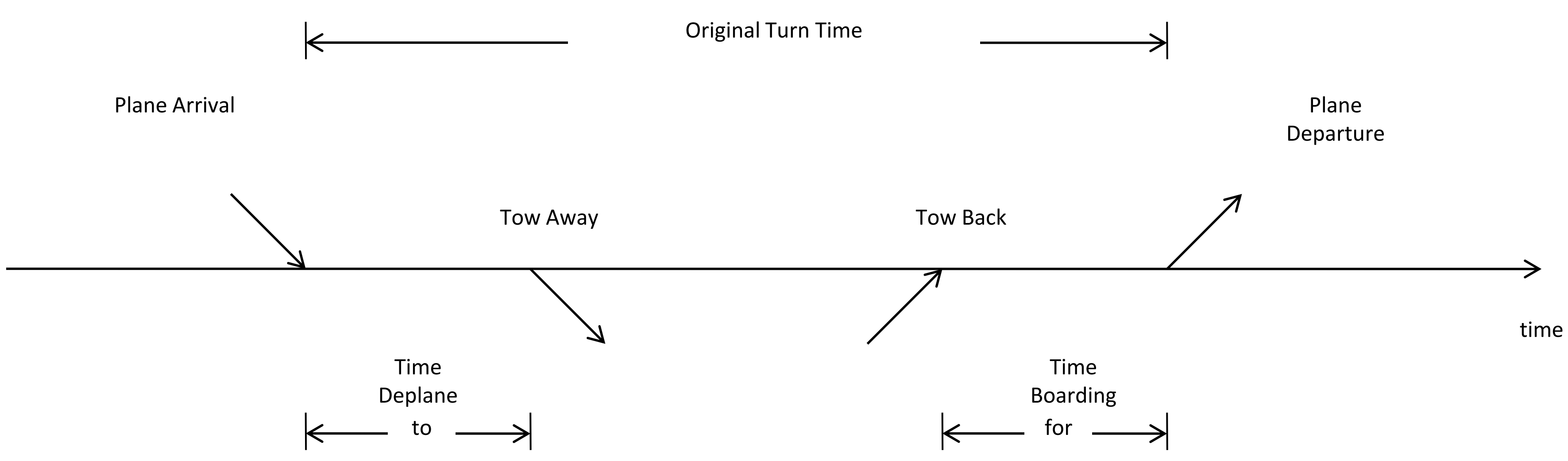
Terminology

- LIFO: Last-In First-Out Gates – These gates are one behind the other making it physically impossible for the aircraft in the inner gate to leave before the aircraft at outer gate departs



Terminology

- Towing: At times, a turn occupies gate for a long time because of the long gap between an incoming flight arrival time and outgoing flight departure time. Aircraft in such cases is towed away to a remote bay so that subsequent arrivals can be gated. Aircraft is brought back to the gate closer to its departure time.



Terminology

- Market: An origin-destination pair
- Turns: A pair of incoming and outgoing flights with the same aircraft or equipment
- Gate Rest: Idle time between a flight departure and next flight arrival to the gate. Longer gate rest helps pad any minor schedule delays, though at the cost of schedule feasibility
- PS Gates: Premium Service gates are a set of gates that get assigned to premium markets – typically where VIPs travel

Mathematical Model

- Parameters
  - a_i: scheduled arrival time of turn
  - b_i: scheduled departure time of turn
  - (*k,l*): two gates restricted in the adjacent pair
  - E_k^1, E_l^1 : sets of equipment types such that when an aircraft of a type in E_k^1 is occupying *k*, no aircraft of any type in E_l^1 may use *l*; and vice versa.
- Decision variables
  - {0,1}: 1 if turn i is assigned to gate k; 0 otherwise
  - {0,1}: 1 if turn i is not assigned to any gate; 0 otherwise
  - {0,1}: 1 if long turn t is towed; 0 otherwise

Mathematical Model

Maximize $\sum_{i \in T} \sum_{k \in K} C_{ik} x_{ik} - C_1 \sum_{t \in L} w_t - C_2 \sum_{i \in T} y_i$

subject to:

$$\sum_{k \in K} x_{ik} + y_i = 1 \qquad i \in T$$

$$y_{ik} + y_{jk} \leq 1 \qquad i, j \in T; k \in K : a_i < b_j + \alpha, a_j < b_i + \alpha, \; i \neq j$$

$$y_{ik} + y_{jl} \leq 1 \qquad i, j \in T; k, l \in K; (k^F_{LiFo}, l^R_{LiFo}) \in LF : a_j \leq a_i \leq b_j, \; i \neq j, \; e_i \in E^F_{k^{LiFo}}, e_j \in E^R_{l^{LiFo}}$$

$$y_{ik} + y_{jl} \leq 1 \qquad i, j \in T; k, l \in K; (k^F_{LiFo}, l^R_{LiFo}) \in LF : a_j \leq b_i \leq b_j, \; i \neq j, e_i \in E^F_{k^{LiFo}}, e_j \in E^R_{l^{LiFo}}$$

Mathematical Model

$$y_{i_1 k} - y_{i_2 k} \leq w_t \quad i_1, i_2 \in T, t \in T_L, k \in K : i_1 \neq i_2, p_{i_1} = p_{i_2} = t$$

$$y_{i_1 k} + y_{i_3 k} - 1 \leq w_t \quad i_1, i_2, i_3 \in T; t \in T_L; k \in K : a_{i_1} < a_{i_3}, b_{i_3} < a_{i_2}, \; p_{i_1} = t, p_{i_2} = t$$

$$y_{i_1 k_1} + y_{i_3 k_2} - 1 \leq w_t \quad i_1, i_2, i_3 \in T; t \in T_L; k_1, k_2 \in K; j \in J : a_{i_1} < a_{i_3}, b_{i_3} < b_{i_2}, \; p_{i_1} = t, p_{i_2} = t, \; k_1 = q_j^1, k_2 = q_j^2, e_{i_1} \in E_j^1, e_{i_2} \in E_j^2$$

Additional Objectives

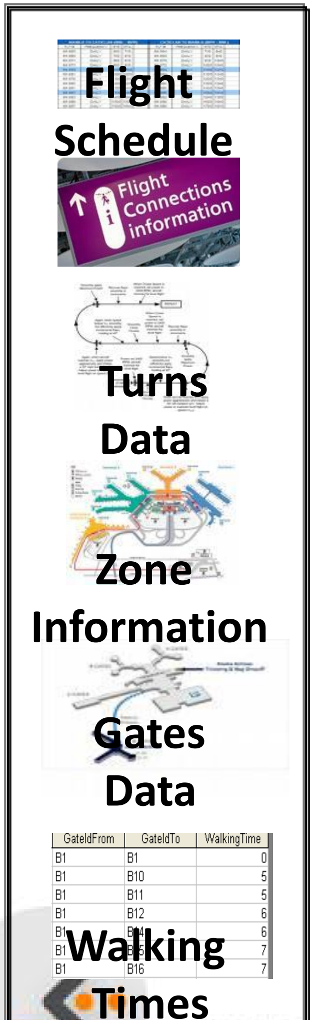
- **Minimize the Gating Costs**
  - Even the airport gates are infrastructure investments, it costs the airport to manage them operationally by providing pushbacks, tow vehicles, etc. There are also penalization
- **Maximize Connection Revenues**
  - This gating objective identifies connections at risk for a hub station and gates the turns involved such that connection revenue is maximized
- **Maximize Robustness**
  - Flights must be gated based on the past pattern of flight delays to provide adequate gate rest between a departing flight and the next arriving flight

Assumptions

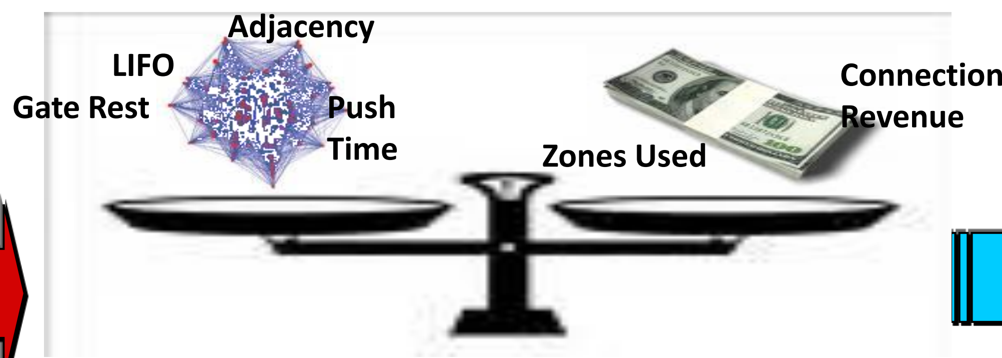

- Connection revenue is realized only if the passenger is able to deplane, walk between the gates and board before the connecting flight departs
- Deplaning time, walking time and boarding times are provided as point estimate inputs
- Connection revenue is provided as point estimate inputs
- For schedule robustness, gate rest accounts for minimum gate rest and average delay of the turns

Input, Model and Output

Inputs



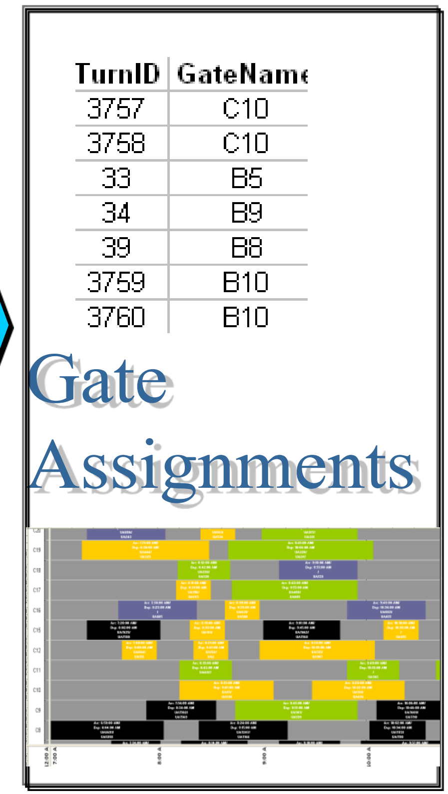
Methodology



- Gate Adjacency
- Gate Rest
- Last In First Out Gates
- Push Back Time
- Towing
- PS Gates

Optimization algorithm

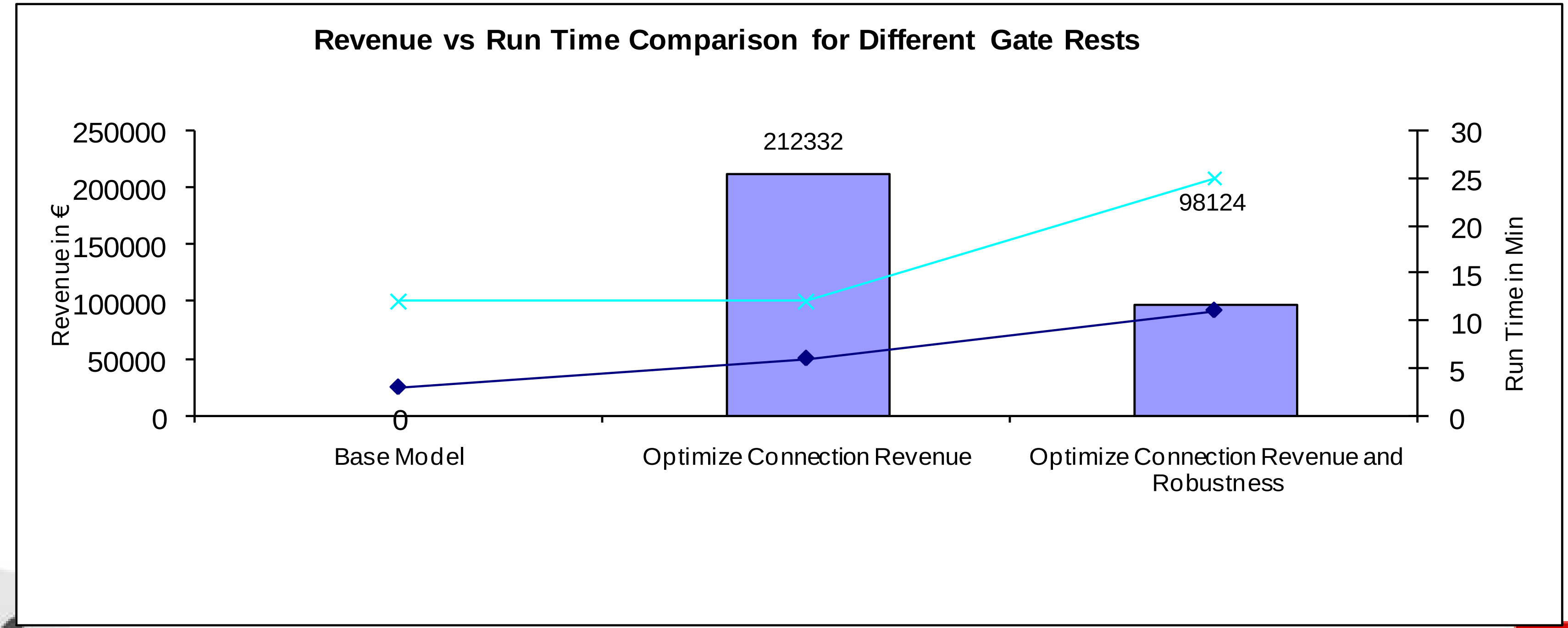
Outputs



TurnID	GateName
3757	C10
3758	C10
33	B5
34	B9
39	B8
3759	B10
3760	B10

Results

- Lift in revenue observed for this objective when evaluated separately as well as in conjunction with other objectives
- Run time increases as
  - number of objectives increase
  - minimum gate rest increases

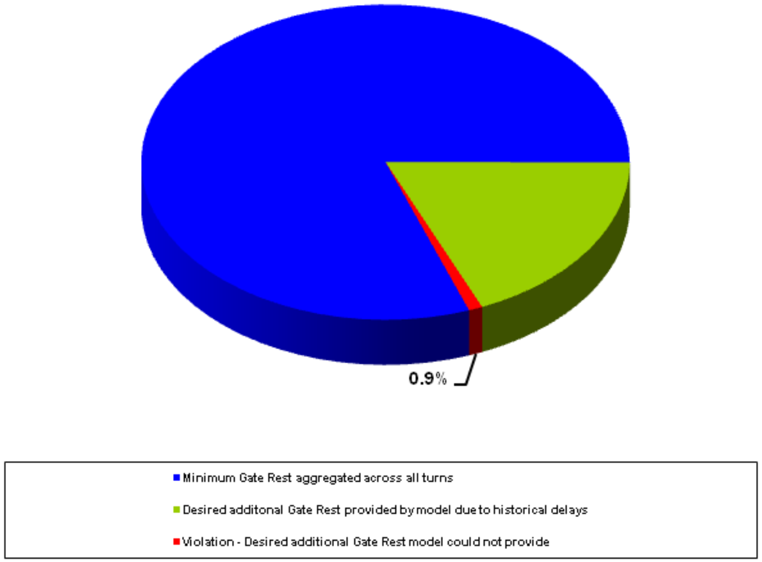


TRANSP-OR

EPFL ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Results

- Desired gate rest is obtained based on historical pattern of delays; violation is only 0.9%
- Run time increases with the complexity of objectives and increasing gate rest



Objective	% Violation
Gating without Robustness Objective	4.0%
Gating with Robustness Objective	0.9%
Gating with Optimizing Robustness and Connection Revenue	0.9%