# Decision Aid Methodologies In Transportation
# Lecture 5: Air transportation problem

Chen Jiang Hang

Transport and Mobility Laboratory

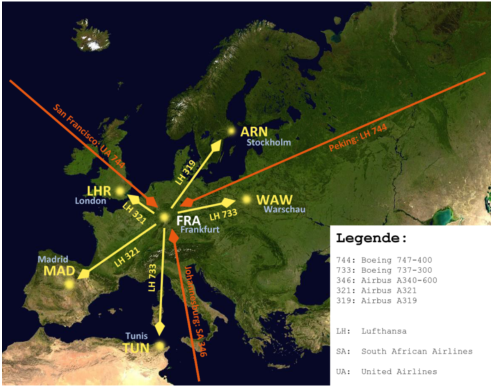
May 5, 2014

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

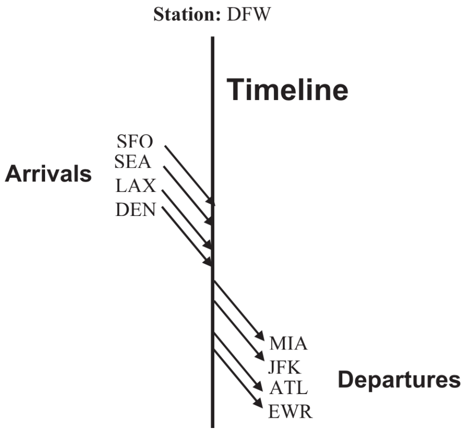
*Air transportation*

# Hub and spoke system: Network structure

Hub and spoke system is widely adopted in transportation especially for air transportation. In such a system (taking air transportation as an example), local airports offer air transportation to the central airport where long-distance flights are available.



# Time bank for hub and spoke airline



# Decision processes in air transportation

- **Schedule Design**: Estimate itinerary level demands and identify suitable flight legs and time
- **Fleet Assignment**: Match demand with supply
- **Aircraft Routing**: Assign individual aircraft to flight legs ensuring consistency and sequence
- **Crew Pairing**: Form sequence of flight legs satisfying human and labor work rules
- **Crew Rostering**: Assign crew (pilots and/or flight attendants) to flight duty sets

# Fleet assignment problem

- At this stage, the demand is known, the main task is to assign demand to supply
- Supply: airline companies operate different types of aircraft fleets
- Main question: which aircraft (fleet) type should fly each flight? Boeing 737, Boeing 767, or A380
  - Aircraft too small $\rightarrow$ lost revenue
  - Aircraft too big $\rightarrow$ costly and inefficient

# Fleet assignment problem

- Set $\mathcal{F}$: a set of available fleets; $S(f)$, $f \in \mathcal{F}$: the number of aircraft available in fleet $f$
- Set $\mathcal{C}$: the set of cities served by the schedule
- Set $\mathcal{L}$: the set of flights in the schedule; $(o, d, t)$, $o, d \in \mathcal{C}$ are OD of the flight and $t$ is the scheduled departure time
- $c_{f,odt}$: the cost for assigning an aircraft from fleet $f$ to the flight $(o, d, t)$
- Times $t_0, t_1, \ldots, t_n$: Assume that arrivals and departures only happen at these discrete instances
- $t^-$: the time preceding $t$; $t^+$: the time following $t$
- $t(f, o, d)$: the traveling time from $o$ to $d$ for an aircraft of type $f$
- $O(t_0)$: the set of flights that are flying during the time interval $[t_0, t_0^+]$
- Set $\mathcal{H}$: a set of pairs of flights that must be performed by an aircraft of the same fleet

# Fleet assignment problem

Decision variables:

- $x_{f,odt} = 1$, if fleet $f$ is used for the flight from $o$ to $d$ departing at time $t$; 0, otherwise.
- $y_{f,ot}$ = number of aircraft on the ground from fleet $f$ that stay at city $o$ during the interval $[t, t^+]$.
- $z_{f,ot}$ = number of aircraft from fleet $f$ that arrive at city $o$ at time $t$.

Obviously,

$$z_{f,ot} = \sum_{\{(d,o,\tau)\in\mathcal{L} \,|\, \tau+t(f,d,o)=t\}} x_{f,do\tau}$$

# Fleet assignment problem

$$\begin{aligned}
\min: \quad & \sum_{f\in\mathcal{F}} \sum_{(o,d,t)\in\mathcal{L}} c_{f,odt} x_{f,odt} \\
s.t. \quad & \sum_{f\in\mathcal{F}} x_{f,odt} = 1, \forall (o,d,t) \in \mathcal{F} \\
& z_{f,ot^-} + y_{f,ot^-} = \sum_{d\in\mathcal{C}} x_{f,odt} + y_{f,ot}, \forall f, o, t \\
& x_{f,odt} = x_{f,dd't'}, \forall f \in \mathcal{F}, \big((o,d,t),(d,d',t')\big) \in \mathcal{H} \\
& \sum_{(o,d,t)\in O(t_0)} x_{f,odt} + \sum_{o\in\mathcal{C}} y_{f,ot_0} \le S(f), \forall f \in \mathcal{F} \\
& x_{f,odt} \in \{0,1\}, y_{f,ot} \in \mathbb{Z}^+
\end{aligned}$$

# Crew pairing problem

- Crew pairing: after the schedule is constructed and fleet are assigned to the flights
- Typically a crew is composed of a pilot, co-pilot and a number of flight attendants
- A crew pairing is one or several days long
- Crew pairing should be checked based on rules and regulations

# Crew pairing problem

Some terms:

- Duty period: mostly a working day of a crew, consists of a sequence of flight legs with short rest periods separating them. Also the duty starts with a brief period and ends with a debrief period.
- Pairing: a sequence of duties and each pairing begins and ends at the same crew base.
- Crew base: a city where crews are stationed.
- Deadhead: to reposition a crew from one base to another base. Generally deadheads are used to transport a crew where they are needed to cover a flight or to return to their home base.

# Crew pairing problem

| | | |
|---|---|---|
| Flight 1: | City A–City B | 08:00–09:00 |
| Flight 2: | City B–City C | 10:00–11:00 |
| Flight 3: | City C–City D | 13:00–14:00 |
| Flight 4: | City C–City A | 07:00–08:00 |
| Flight 5: | City D–City A | 07:00–08:00 |
| Flight 6: | City A–City B | 17:00–18:00 |
| Flight 7: | City B–City C | 11:00–12:00 |

The the possible pairings can be:

$$\begin{array}{ll} P_1 = \{F_1, F_2, F_4\} & c_1 = 4 \\ P_2 = \{F_1, F_3, F_5, F_7\} & c_2 = 3 \\ P_3 = \{F_2, F_3, F_5, F_6\} & c_3 = 5 \end{array}$$

# Crew pairing problem

- $\mathcal{F}$: the set of all flights
- $\mathcal{P}$: the set of all possible pairings
- $\mathcal{P}^i$: the set of pairing which cover the flight $i$, $i \in \mathcal{F}$
- $c_j$: the cost of pairing $j \in \mathcal{P}$

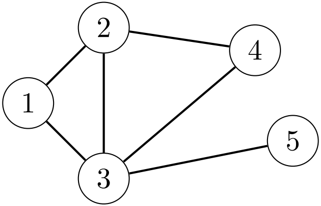
$$\begin{aligned}
\min: \quad & \sum_{j \in \mathcal{P}} c_j x_j \\
s.t. \quad & \sum_{j \in \mathcal{P}^i} x_j = 1, \forall i \in \mathcal{F} \\
& x_j \in \{0, 1\}, \forall j \in \mathcal{P}
\end{aligned}$$

Basically, it is a **set covering problem**! Question: if the $|\mathcal{P}|$ is huge, what kind of technique can be used to speed up the the solving?

*Network flow problem*

# Undirected graphs

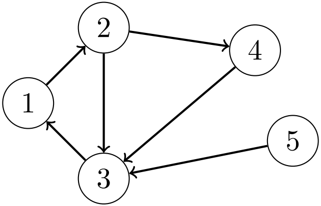
An *undirected graph* $G = (\mathcal{N}, \mathcal{E})$ consists of a set $\mathcal{N}$ of nodes and a set $\mathcal{E}$ of undirected edges, where an edge $e$ is an unordered pair of distinct nodes, that is, a two-element subset $\{i, j\}$ of $\mathcal{N}$.




- Walk: a finite sequence of nodes $i_1, i_2, \cdots, i_t$ such that $\{i_k, i_{k+1}\} \in \mathcal{E}$, $k = 1, 2, \cdots, t-1$
- Path: a walk without repeated nodes
- Cycle: a walk $i_1, i_2, \cdots, i_t$ such that nodes $i_1, i_2, \cdots, i_{t-1}$ are distinct and $i_t = t_1$
- Connected undirected graph

# Directed graphs

An *directed graph* $G = (\mathcal{N}, \mathcal{A})$ consists of a set $\mathcal{N}$ of nodes and a set $\mathcal{A}$ of directed arcs, where an arc $a$ is an ordered pair of distinct nodes, that is, a two-element subset $(i, j)$ of $\mathcal{N}$.



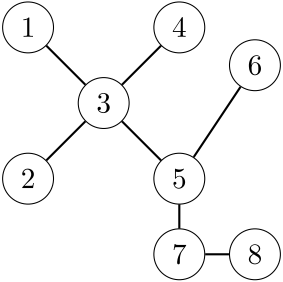

- Walk: a finite sequence of nodes $i_1, i_2, \cdots, i_t$ such that $(i_k, i_{k+1}) \in \mathcal{A}$ or $(i_{k+1}, i_k) \in \mathcal{A}$; directed walk
- Path: a walk without repeated nodes; directed Path
- Cycle: a walk $i_1, i_2, \cdots, i_t$ such that nodes $i_1, i_2, \cdots, i_{t-1}$ are distinct and $i_t = t_1$; directed cycle
- Connected directed graph

# Tree

An undirected graph $G = (\mathcal{N}, \mathcal{E})$ is called a tree if it is connected and has no cycles.

### Properties of a tree

1. An undirected graph is a tree **if and only if** it is connected and has $|\mathcal{N}| - 1$ edges
2. If we start with a tree and add a new arc, the resulting graph contains exactly one cycle



# Network flow problem

A network is a directed graph $G = (\mathcal{N}, \mathcal{A})$ together with some additional numerical information.

- $b_i$: external supply to node $i$
- $u_{ij}$: capacity of arc $(i, j)$
- $c_{ij}$: cost per unit of flow along arc $(i, j)$

Let $f_{ij}$ be the amount of flow through arc $(i, j)$ and we call a node $i$ *source (sink)* if $b_i > 0 (b_i < 0)$.

### Flow conservation constraints

$$b_i + \sum_{j \in I(i)} f_{ji} = \sum_{j \in O(i)} f_{ij}, \forall i \in \mathcal{N}$$

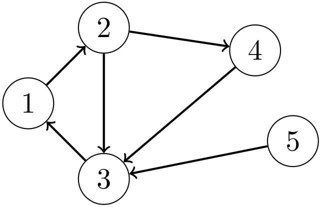
### Flow capacity constraints

$$0 \leq f_{ij} \leq u_{ij}, \forall (i, j) \in \mathcal{A}$$

# Incidence matrix of a network

A matrix associated with a network and its $(i,k)$th entry $a_{ik}$ is associated with the $i$th node and the $k$th arc.

$$a_{ik} = \begin{cases} 1, & \text{if } i \text{ is the start node of the } k\text{th arc;} \\ -1, & \text{if } i \text{ is the end node of the } k\text{th arc;} \\ 0, & \text{otherwise.} \end{cases}$$




$$\mathbf{A} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

# Flow conservation and circulations

Given the incidence matrix $\mathbf{A}$, the flow conservation constraints can be written in a concise way:

$$\mathbf{A}\mathbf{f} = \mathbf{b}$$

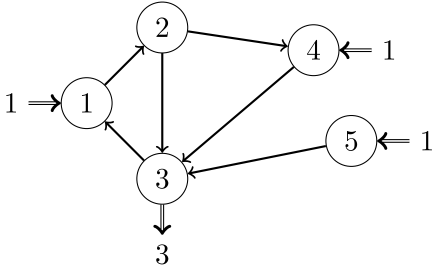
Let's examine the nullspace of matrix $\mathbf{A}$, $N(\mathbf{A}) = \{\mathbf{f} \mid \mathbf{A}\mathbf{f} = \mathbf{0}\}$. We call any flow $\mathbf{f} \in N(\mathbf{A})$ a circulation of the network. Now consider a cycle $C$ of the network, let $F$ and $B$ be the set of forward and backward arcs of the cycle. The flow vector $\mathbf{h}^C$ with components

$$h_{ij}^C = \begin{cases} 1, & \text{if } (i,j) \in F; \\ -1, & \text{if } (i,j) \in B; \\ 0, & \text{otherwise.} \end{cases}$$

is called the **simple circulation** associated with the cycle $C$. Obviously, $\mathbf{h}^C \in N(\mathbf{A})$, i.e., $\mathbf{A}\mathbf{h}^C = \mathbf{0}$.

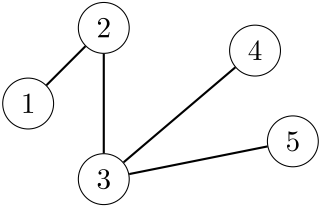
# Network simplex method: Tree solution

Given the network:



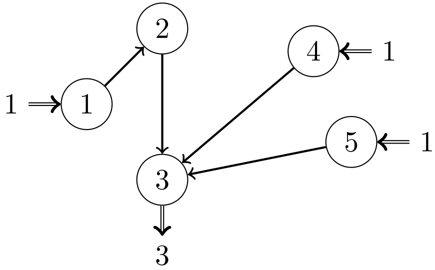
# Network simplex method: Tree solution

Ignore directions of arcs and find a spanning tree:



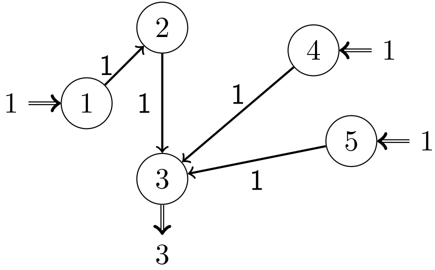
# Network simplex method: Tree solution

Recover the direction & flow information:



# Network simplex method: Tree solution

Use flow conservation constraint to obtain a tree solution:




### The importance of tree solutions

Tree solution is equivalent to basic solution! Feasible tree solution is the basic feasible solution of the network optimization problem $\min\{\mathbf{c}'\mathbf{f} \mid \mathbf{Af} = \mathbf{b}, \mathbf{f} \geq \mathbf{0}\}$!

# Pivoting process

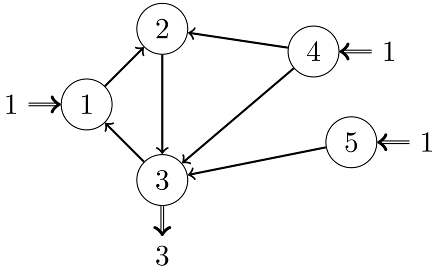
- Creating cycle: add an arc not in the tree with **zero** flow to the current tree and a cycle $C$ will be created (Why?!).
- Construct a simple circulation $\mathbf{h}^C$ associated with cycle $C$. Note that if the positive value $\theta$ is small enough, then $\mathbf{A}(\mathbf{f}_T + \theta\mathbf{h}^C) = \mathbf{b}$ and $\mathbf{f}_T + \theta\mathbf{h}^C \geq \mathbf{0}$. That is, $\mathbf{f}_T + \theta\mathbf{h}^C$ is a feasible flow (Why?!).
- Calculate the reduced cost for the selected arc.

$$\bar{c}_{ij} = \sum_{(k,l)\in F} c_{kl} - \sum_{(k,l)\in B} c_{kl}$$

- Select one arc with negative reduced cost and try to "push" flow around the cycle $C$ as much as possible (greedy!).
- Determine the arc in the current tree who carries zero flow now after the flow "pushing".
- A new and better tree solution has been found.

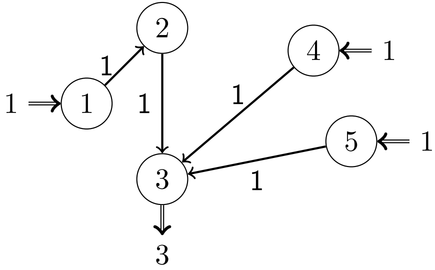
# An example

We have following network and suppose $c_{ij} = 1$ $(i, j) \in \mathcal{A}$ except that $c_{42} = 0.5$ and $c_{43} = 2$.



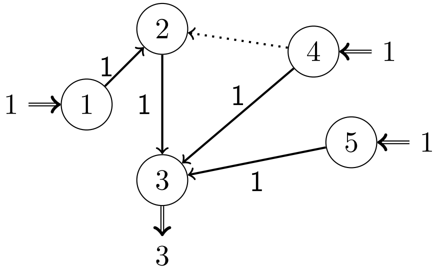
# An example

Let's start from the following feasible tree solution:



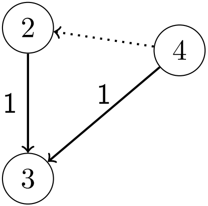
# An example

Let's try to bring arc $(4, 2)$ into the basis:



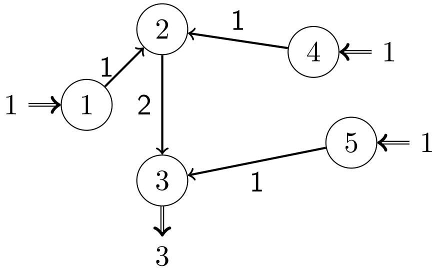
# An example

The reduced cost associated with arc $(4, 2)$ is equal to $c_{42} + c_{23} - c_{43} = 0.5 + 1 - 2 = -0.5$.



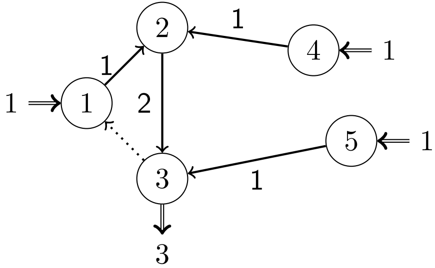
# An example

After pushing 1 flow around the cycle:



# An example

Let's try to bring arc $(3, 1)$ into the basis:



# An example

However, the reduced cost associated with arc $(3, 1)$ is equal to $c_{31} + c_{12} + c_{23} = 1 + 1 + 1 = 3 > 0$.