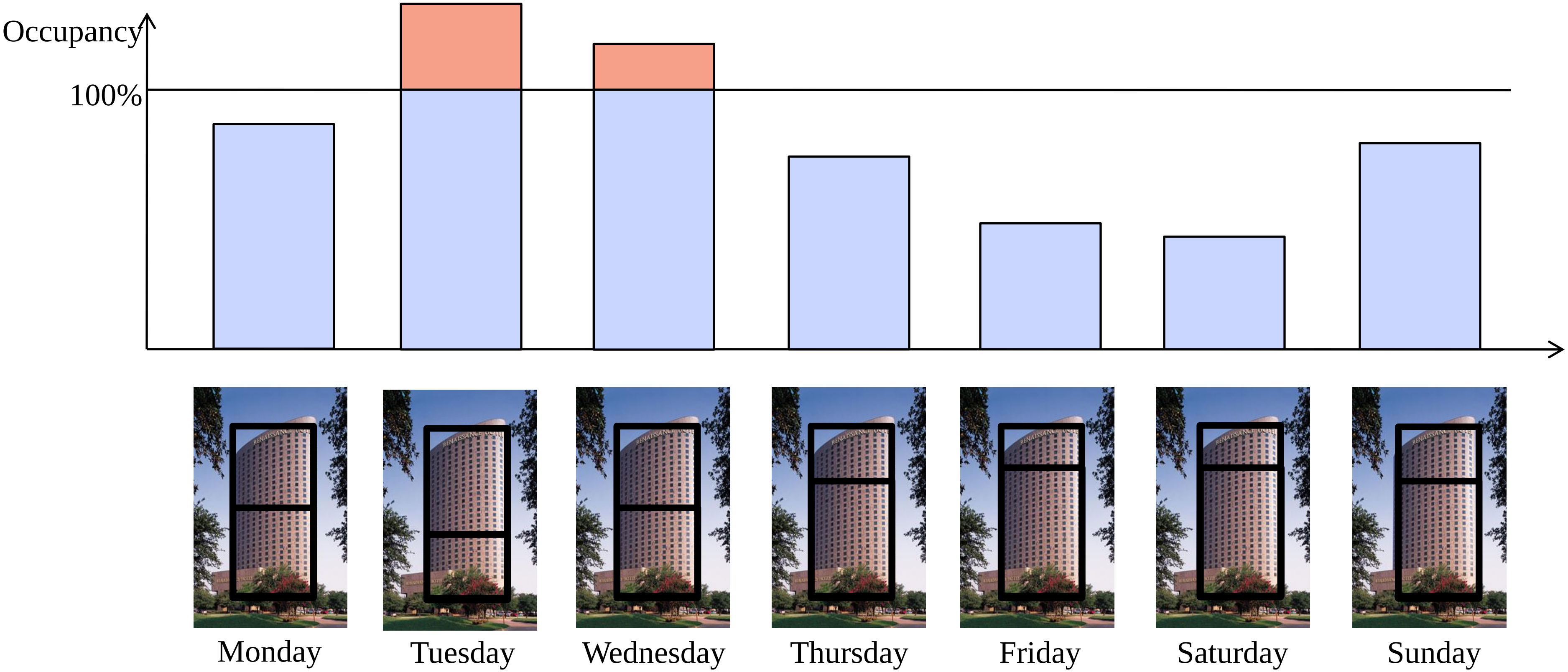
Network Revenue Management

Outline

- **Network Management Problem**
- **Greedy Heuristic**
- **LP Approach**
- **Virtual Nesting**
- **Bid Prices**

Based on Phillips (2005) Chapter 8

Demand for Hotel Rooms Vary over a Week



Since the demand varies, so should the booking limits.

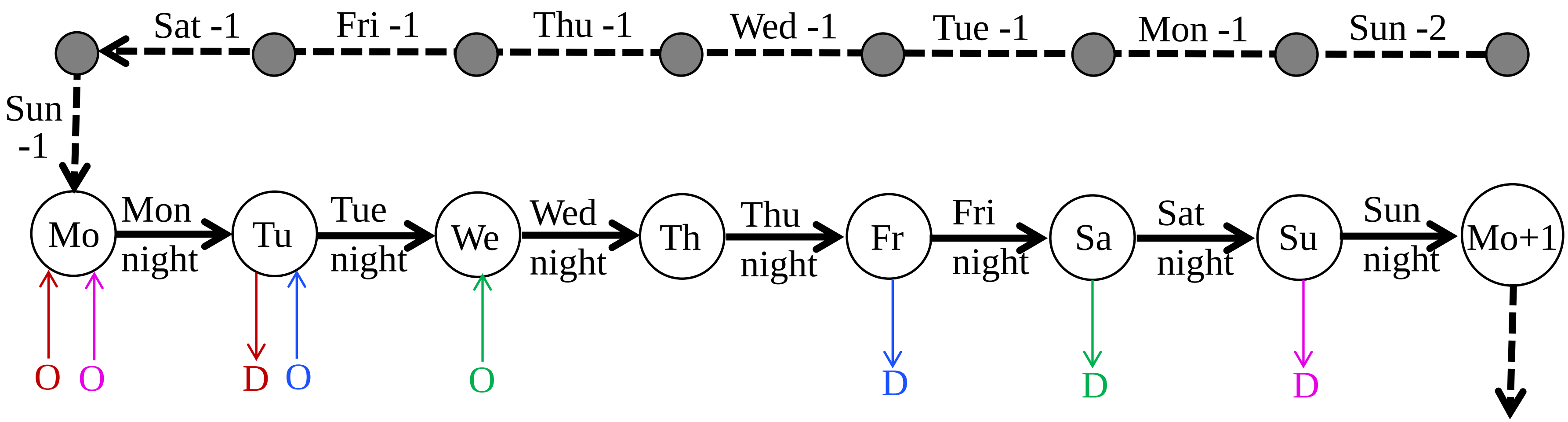
Many customers book for several days in a row

Which booking limits apply to a booking request for Tuesday night check-in and 3 night stay?

Since requests are coupled over days, so should the booking limits.

Compute the booking limits not separately day by day but by jointly for seven days!

Hotel Network: Chain



Origin-Destination (check-in,check-out) day pairs with 7 nights:

Mo-Tu, Mo-We, Mo-Th, Mo-Fr, Mo-Sa, Mo-Su, Mo-Mo+1 all check in on Monday but stay 1, 2, …, 7 nights

Tu-We, Tu-Th, Tu-Fr, Tu-Sa, Tu-Su, Tu-Mo+1 all check in on Tuesday but stay 1, 2, …, 6 nights

We-Th, We-Fr, We-Sa, We-Su, We-Mo+1 all check in on Wednesday but stay 1, 2, …, 5 nights

Th-Fr, Th-Sa, Th-Su, Th-Mo+1 all check in on Thursday but stay 1, 2, 3, 4 nights

Fr-Sa, Fr-Su, Fr-Mo+1 all check in on Friday but stay 1, 2, 3 nights

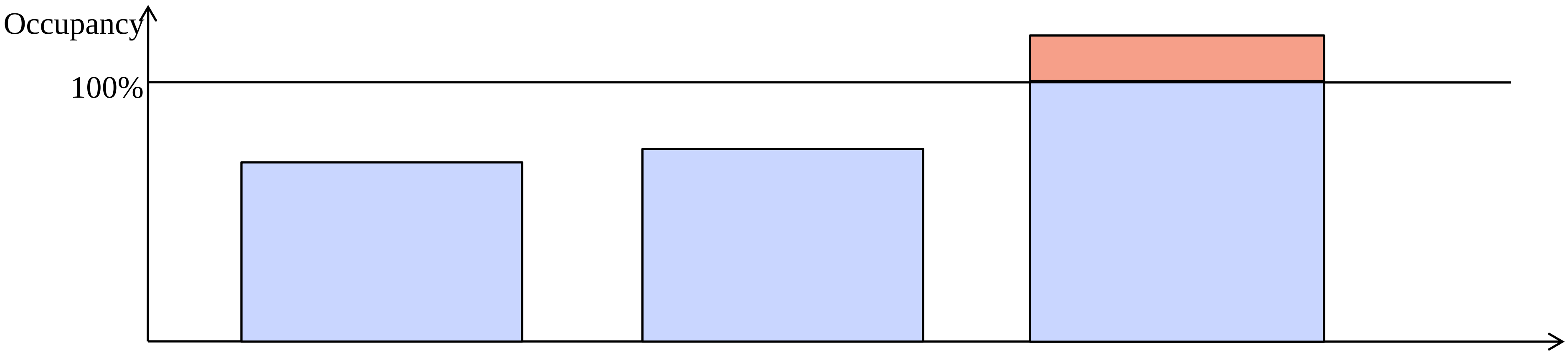
Sa-Su, Sa-Mo+1 all check in on Saturday but stay 1, 2 nights

Su-Mo+1 checks in on Sunday and stays 1 night

A total of 1+2+3+4+5+6+7=7*8/2=28 O-D pairs.

Over 365 nights 365*366/2=66795 O-D pairs. Standard, deluxe, deluxe with a view, suit rooms, and suppose each has a fare class of its own. 66795*4=267180 ODFs.

Demand for Flights Vary over Legs



College Station → Dallas

Dallas → Los Angeles

Los Angeles → Honolulu

Since the demand varies, so should the booking limits.

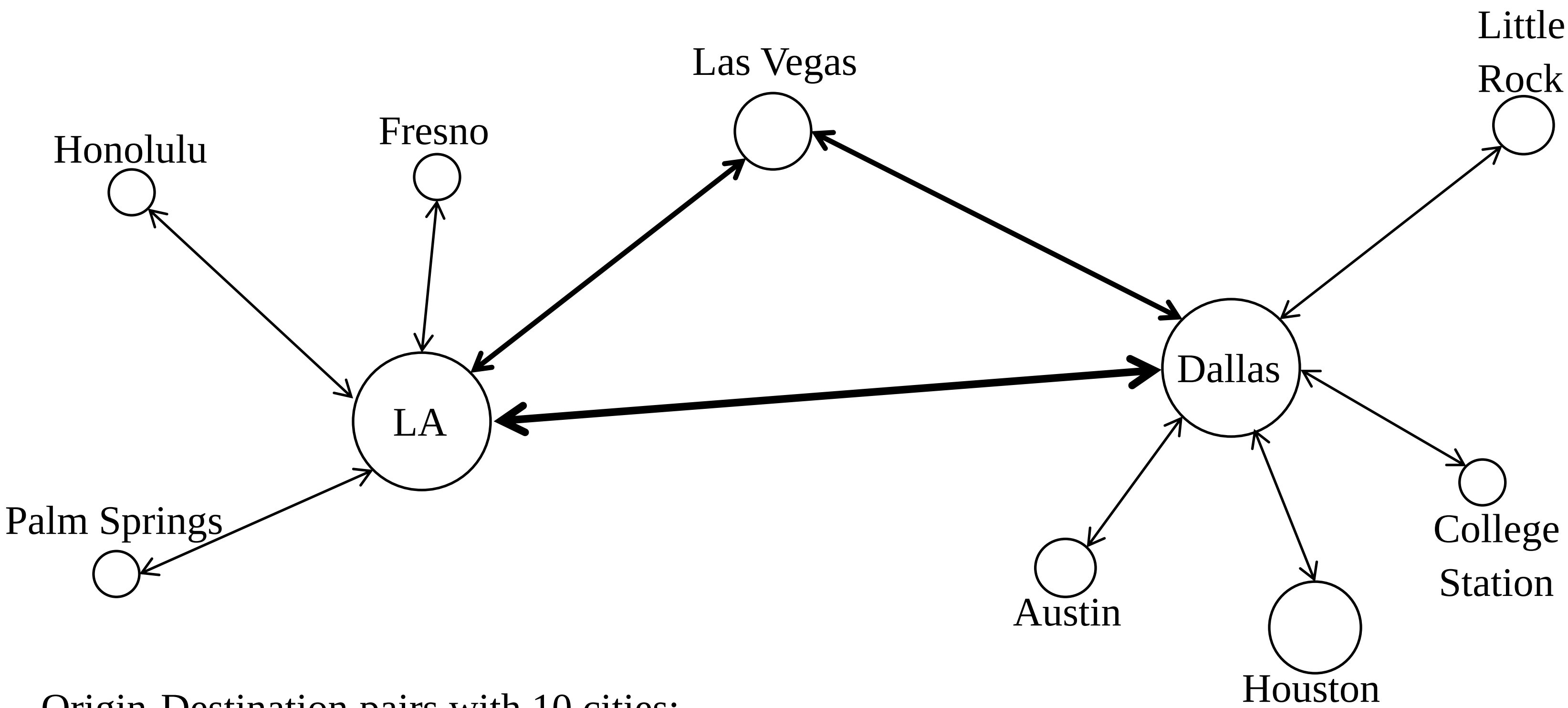
Many customers book for College Station → Dallas → Los Angeles → Honolulu trip

Which booking limits apply to this booking request?

Since requests are coupled over legs, so should the booking limits.

Compute the booking limits not separately leg by leg but by jointly for an itinerary!

Airline Network: Hub-and-spoke



Origin-Destination pairs with 10 cities:

- College Station → {Dallas, Houston, Austin, Little Rock, Las Vegas, LA, Fresno, Palm Springs, Honolulu}
- Fresno → {LA, Honolulu, Palm Springs, Las Vegas, Dallas, Houston, Austin, Little Rock, College Station}
- Dallas → {College Station, Houston, Austin, Little Rock, Las Vegas, LA, Fresno, Palm Springs, Honolulu}

Number of pairs is 10*9=90, where Dallas → Austin and Austin → Dallas treated as different legs.

Origin-Destination-Fare (ODF) class Combinations with 10 cities and 6 fares on every flight

Number of ODFs is 90*6=540

Too many ODFs, so joint optimization is a big challenge!

1. Greedy Heuristic

- Since joint optimization is difficult, let us focus on each leg to develop a greedy heuristic. This is decomposition of the itineraries by force.
- Suppose that we consider the origin destination pair of College Station → Dallas → LA.
- For College Station → Dallas leg,
  - Full fare is $200 and discount fare is $120.
- For Dallas → LA leg,
  - Full fare is $400 and discount fare is $350.
- Because of the high passenger traffic out of College Station, we may close the discount fare class on College Station → Dallas leg.
  - Once this class is closed, we cannot accept a discount booking from College Station to LA which brings in a revenue of $470.
  - This revenue of $470 is more than the full fare of $200 for a customer flying from College Station → Dallas.
  - We would rather displace a full fare customer on College Station → Dallas leg than reject a discount fare customer on College Station → LA OD pair.
- By considering legs one by one, we never entertain the possibility of protecting seats for a discount fare passenger that travels multiple legs from a full fare customer that travels a single leg although multiple leg customer pays (much) more than single leg customer.
- Greedy heuristic does not work; it failed miserably when applied to hotel booking.

2. Linear Programming Approach

- Suppose that the demands are known in advance. Think of assembling resources (legs, night-stays) to make up products (itineraries, hotel-stays).
- Index products by $j=1,2,\ldots,n$ and resources by $i=1,2,\ldots,m$.
- p_j: price of product j; d_j: demand of product j; c_i: capacity of resource i;
- All resources

  1,11: College Station →, ← Dallas; 2,12: Little Rock →,← Dallas; 3,13: Austin →,← Dallas;
  4,14: Houston →,← Dallas; 5,15: Dallas →,← LA; 6,16: Dallas→,← Las Vegas; 7,17: LA→,← Las Vegas;
  8,18: LA→,← Fresno; 9,19: LA→,← Honolulu; 10,20: LA →,← Palm Springs;
- Some products
  - Product 1: College Station → LA discount fare, uses resources 1 and 5 or resource 1 and 5 are used to make up product 1: $a_{1,1}=1$, $a_{5,1}=1$.
  - Product 2: Little Rock → Honolulu full fare, uses resources 2, 5 and 9 or resources 2, 5, and 9 make up product 2: $a_{2,2}=1$, $a_{5,2}=1$, $a_{9,2}=1$.
  - Product 3: Las Vegas → Dallas discount fare, uses only resource 16 or resource 16 is product 3: $a_{16,3}=1$.
  - Product 4: Las Vegas → Little Rock discount fare, uses resources 12 and 16 or resources 12 and 16 make up product 4: $a_{12,4}=1$, $a_{16,4}=1$.
  - Product 5: Las Vegas → Little Rock full fare, uses resources 12 and 16 or resources 12 and 16 make up product 5: $a_{12,5}=1$, $a_{16,45}=1$.
- $a_{ij}=1$ if resource i is used in product j; 0 otherwise.
- x_j=Total seats allocated to product (ODF) j.

$$\max_{x_j} \sum_{j=1}^{n} p_j x_j$$

st (subject to)

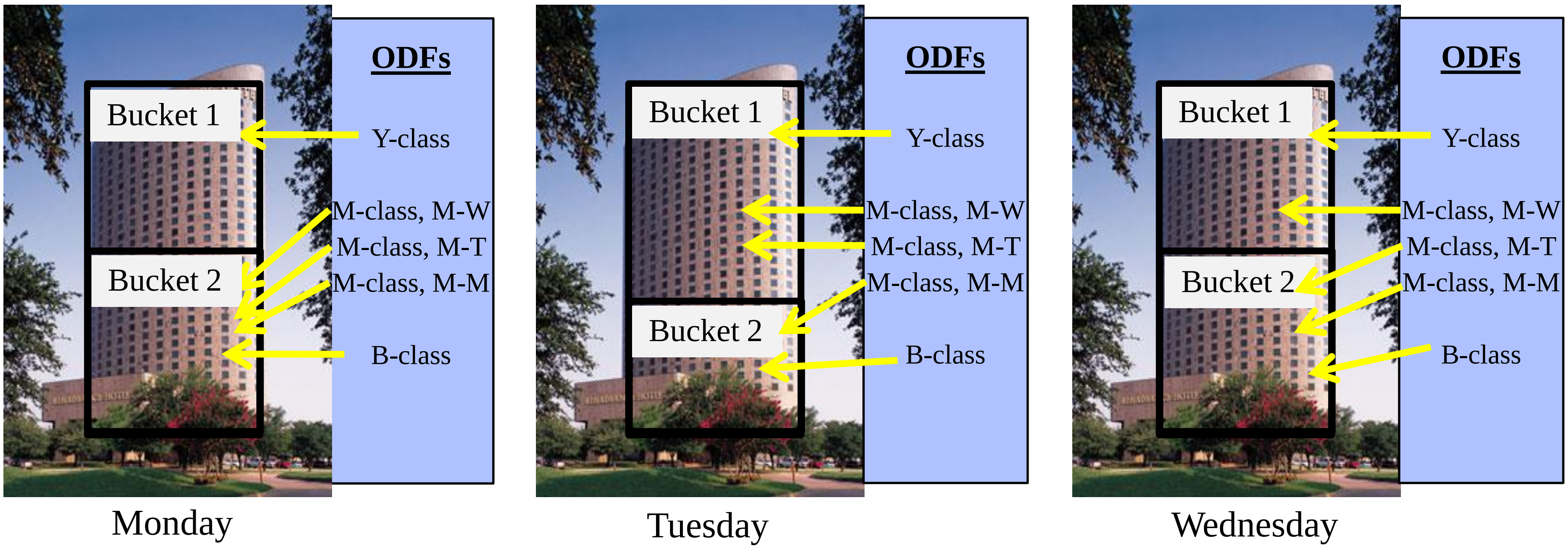
$$\sum_{j=1}^{n} a_{ij} x_j \le c_i \quad \text{for all resources } i$$

$$0 \le x_j \le d_j \quad \text{for all products } j$$

Linear Programming Discussion

- Example Solution:
  - x_4=5 for ODF Las Vegas → Little Rock discount fare. 5 discount fare seats are allocated to this ODF on resource 12 and 16: Dallas → Little Rock and Las Vegas → Dallas legs.
  - x_5=10 for ODF Las Vegas → Little Rock full fare. 10 full fare seats are allocated to this ODF on resource 12 and 16: Dallas → Little Rock and Las Vegas → Dallas legs.
- Observations:
  - Different fare classes on the same OD pair consume identical resources.
  - Consider two fare classes on the same pair: high-fare class and low-fare class.
  - If the high-fare class has no allocation, then the low-fare class will have no allocation.
  - If the low-fare class is allocated a capacity equal to all of its demand, then the high-fare class is allocated a capacity equal to all of its demand.
- In summary, a high-fare class OD pair has higher priority than the low-fare class OD pair when receiving resource capacities provided that both classes consume the same capacity.
- Linear programming is an efficient methodology; it is fast to find a solution.
- Linear programming may not be effective
  - It assumes that the demands are known.
  - It is an allotment method as it does not allow for nested classes.
- Question: Can we achieve some nesting, if not all?

3. Virtual Nesting



- Virtual Nesting starts by assigning every ODF to a bucket in a resource that is used in the ODF.
  - A bucket works like a fare class.
  - A bucket is a collection of ODF's.
  - Think of higher buckets as higher customer fare classes where we price the resources higher.
- ODF-to-bucket assignment is called indexing.
- Given this assignment, we find the booking limits by using EMSR heuristics or another method.
- The key issue here is indexing.

Indexing

- Which ODF should go into the highest bucket?
- First attempt: Order ODF by prices and put highest priced ODFs into the highest bucket.
  - Problem: High priced ODFs consume more resources
    - You pay more if you travel further
    - You pay more if you stay longer
  - Classification by ODF price alone ignores the consumption of other sources.
  - Discourage ourselves from giving priority to high priced ODFs that consume a lot of resources.
- Inspired from profit margin=price-cost, consider net leg fare of an ODF based on opportunity cost of capacity:

Net leg fare of ODF *j* on resource *i* = p_j − **sum of opportunity costs on all resources other than resource *i*.**

- Example: An airline estimates the opportunity costs for

  a seat on College Station → Dallas to be $80

  a seat on Dallas→ LA to be $210

What are the net leg fares for M-fare class whose College Station→ LA itinerary is priced at $400?

On College Station → Dallas leg, net leg fare of this itinerary is 400-210=190, use this number when bucketing College Station → LA itinerary M-fare class on College Station → Dallas leg.

On Dallas→ LA, net leg fare of this itinerary is 400-80=320, use this number when bucketing College Station → LA itinerary M-fare class on Dallas → LA leg.

Dynamic Opportunity Cost of Capacity

- If we have a network with 8 legs in a network and b_i denotes the capacity on leg (O-D) *i* while $\Pi(b)$ is the profit that we can make from the network with capacity vector $b=[b_1,b_2,b_3, b_4, b_5, b_6, b_7, b_8]$.
- The marginal cost of accepting an ODF *j* on leg *i* is

$$a_{ij} \frac{\mathrm{d}}{\mathrm{d}b_i} \Pi(b,t)$$

Recall $a_{ij}=1$ if leg (resource) *i* is used in ODF (product) *j*; 0 otherwise.

- If we are considering to accept an ODF 2 that uses legs 3,4 and 7, we consider the marginal costs

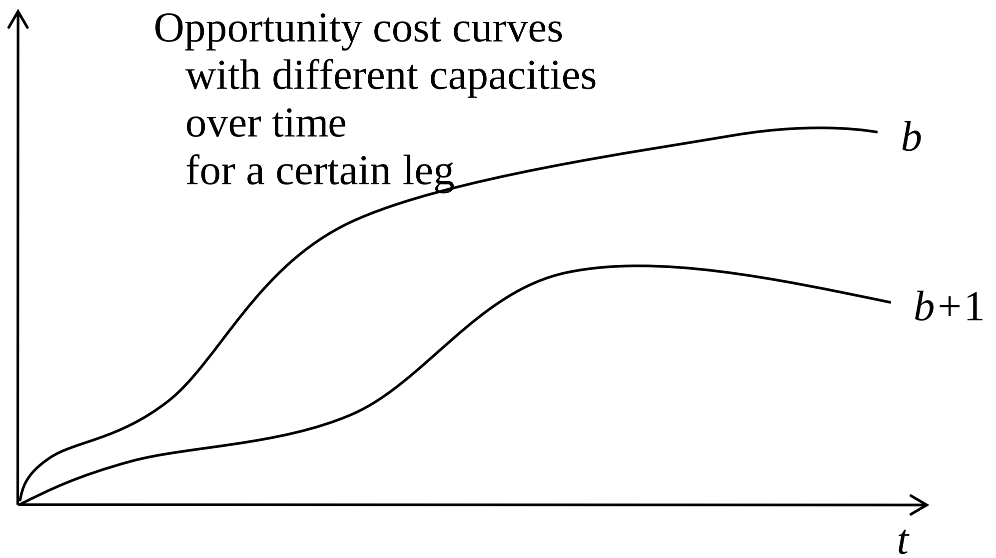
$$\frac{\mathrm{d}}{\mathrm{d}b_3} \Pi(b,t), \quad \frac{\mathrm{d}}{\mathrm{d}b_4} \Pi(b,t), \quad \frac{\mathrm{d}}{\mathrm{d}b_7} \Pi(b,t).$$

Because only $a_{32}=a_{42}=a_{72}=1$ for this ODF and others are all 0.

- This marginal cost computations are conceptually same as those in the case of a single-leg product.
- However, there are complexity and numerical issues when this type of computation is done in
  - a network that includes hundreds and sometimes thousands of legs (resources),
  - a network whose remaining capacity *b* changes dynamically over time *t*.

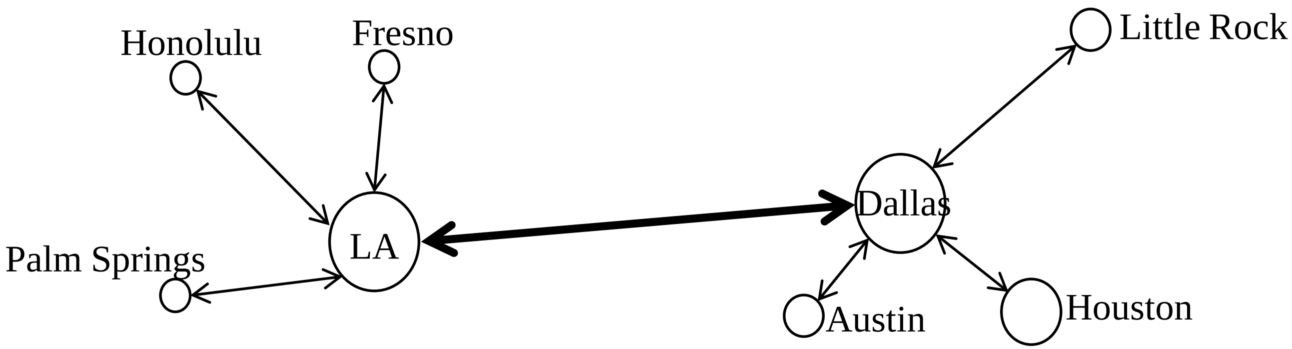
Static Opportunity Cost of Capacity Ever Possible?

- Opportunity cost of capacity depends on the time *t* that remains to sell the capacity.
  - If *t* is very small, the opportunity cost is low. Because we do not have time to sell the capacity if we deny the current request.
- Opportunity cost of capacity also depends on the capacity b.
  - If b is very large, the opportunity cost is low. Because we cannot sell the large capacity if we deny the current request.



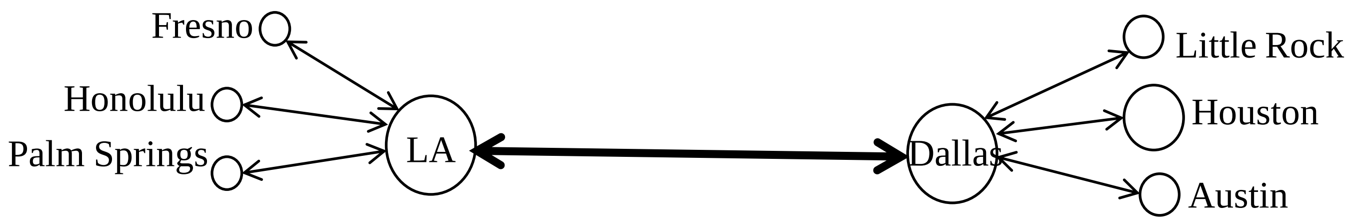
- How to summarize several curves for a single leg in a single number?

Indexing with Net leg fares on Houston→ Dallas leg



Origin	Destination	Net leg fare	Bucket
Houston	Fresno	180	1
Houston	Little Rock	120	1
Houston	Austin	110	1
Houston	Dallas	100	2
Houston	Palm Springs	85	2
Houston	**Honolulu**	**40**	**3**
Houston	LA	30	3

Indexing with Net leg fares on Dallas → LA leg



Origin	Destination	Net leg fare	Bucket
Houston	Fresno	160	1
Austin	Honolulu	160	1
Houston	**Honolulu**	**140**	**1**
Austin	LA	110	1
Austin	Fresno	90	2
Little Rock	Fresno	85	2
Houston	Palm Springs	80	2
Austin	Palm Springs	60	2
Little Rock	Honolulu	40	3
Dallas	LA	30	3
Little Rock	Palm Springs	25	3
Little Rock	LA	25	3
Houston	LA	20	3

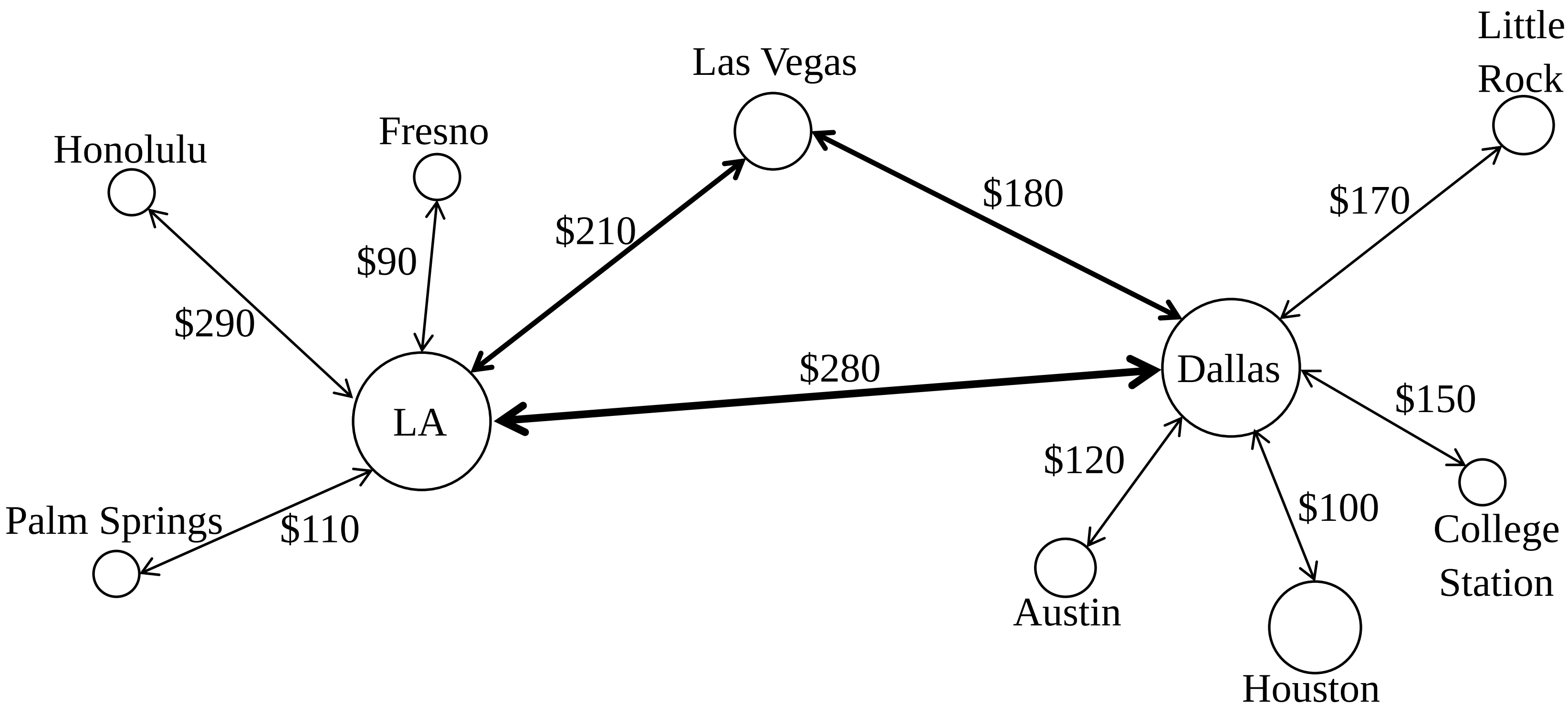
Houston-Honolulu ODF is
in bucket 3 on Houston → Dallas flight, but
in bucket 1 on Dallas→ LA flight.

This is possible when
Dallas→LA flight has a higher opportunity cost than Houston→Dallas flight.

Recall that Dallas→LA opportunity cost does not affect the leg fare of Dallas→LA but it does affect the leg fare of Houston→Dallas.

Inconsistency or not: As a result of these, if you are going from Houston to Honolulu, you can find the Dallas→LA ticket but not the Houston→Dallas ticket. The funny thing is that your friend buys a Houston→Dallas ticket after you!!

4. Additive Bid Prices



Would you close a fare class on

Las Vegas → Fresno itinerary if the price is $350?

Little Rock → Honolulu itinerary if the price is $800?

Palm Springs → Austin itinerary if the price is $400?

Skip the section on calculation of the bid prices. Recall that bid prices are about opportunity costs.

Just like opportunity costs, bid prices must be dynamic. Applications in Airlines (Scandinavian Airline System) and Rental Cars (Hertz).

Virtual Nesting or Bid Prices

- Virtual nesting and bid prices both depend on the opportunity cost of the capacity.
- The opportunity cost needs to be updated dynamically as time passes and remaining capacity drops.
  - Both virtual nesting and bid pricing methods perform similarly when opportunity costs are updated frequently.
- If the bid prices are not updated and

  demand is stronger than expected, bid price becomes lower than what it should be (underpricing),

  demand is weaker than expected, bid price become higher than what it should be (overpricing).
  - Ignorance of dynamism is more costly when bid pricing methods are used.

- Historically, airlines had RM before hotels for single sources. Airlines used fare class based booking controls.
- When network RM management came out, the legacy systems at Airlines are modified to take legs of an OD to take into account. This led to virtual nesting.
- On the other hand, Hotels did not have any (legacy) RM systems so they started with bid pricing systems.
- Over time, some airlines adopted bid pricing.

Practice as of 2011
Dynamic Virtual Nesting or Bid Prices

Dynamic Virtual Nesting
- United Airlines
- Delta Airlines

Bid Prices
- American Airlines
- Lufthansa/Swiss
- LAN
- Iberia
- British Airways
- Air France/KLM
- SAS
- Cathay Pacific
- Qatar
- Etihad
- Royal Jordanian
- Thai

"Some of these airlines are using what we refer to as 'hybrid' controls, whereby leg authorization limits are used to manage local services, and bid prices are used for connecting ones. That particular variation of bid price controls retains aspects of leg authorizations levels (for RM analysts who are more familiar with these types of controls), but it enables bid prices for connecting services to avoid the huge increase in the total volume of controls required in an O&D implementation."

Richard Ratliff , Senior Research Scientist, Sabre, Nov 2011

Lists above are based on his Nov 21 guest lecture.

Summary

- Network Management Problem
- Greedy Heuristic
- LP Approach
- Virtual Nesting
- Bid Prices