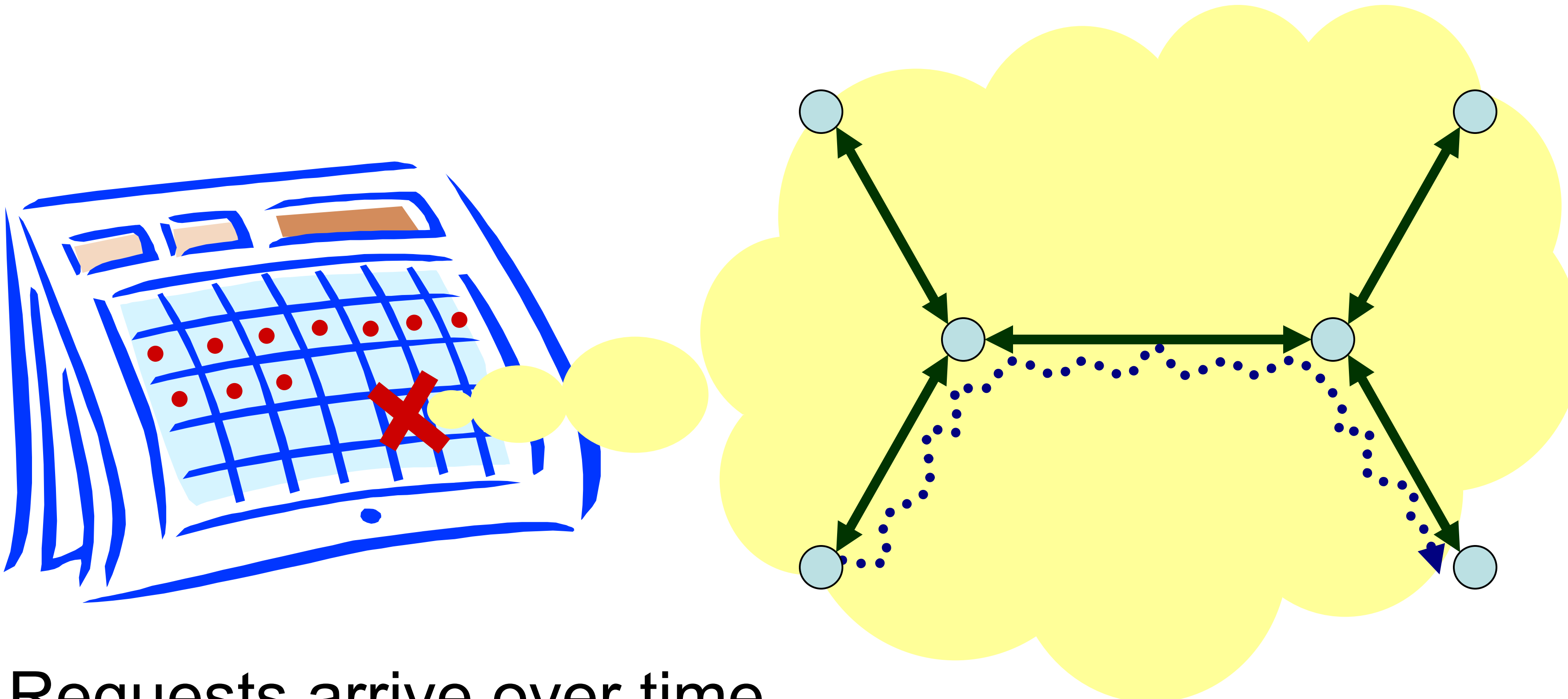
# Capacity-Dependent Bid-Prices for Network Revenue Management

Huseyin Topaloglu
School of Operations Research and Information Engineering
Cornell University

Sumit Kunnumkal
Indian School of Business

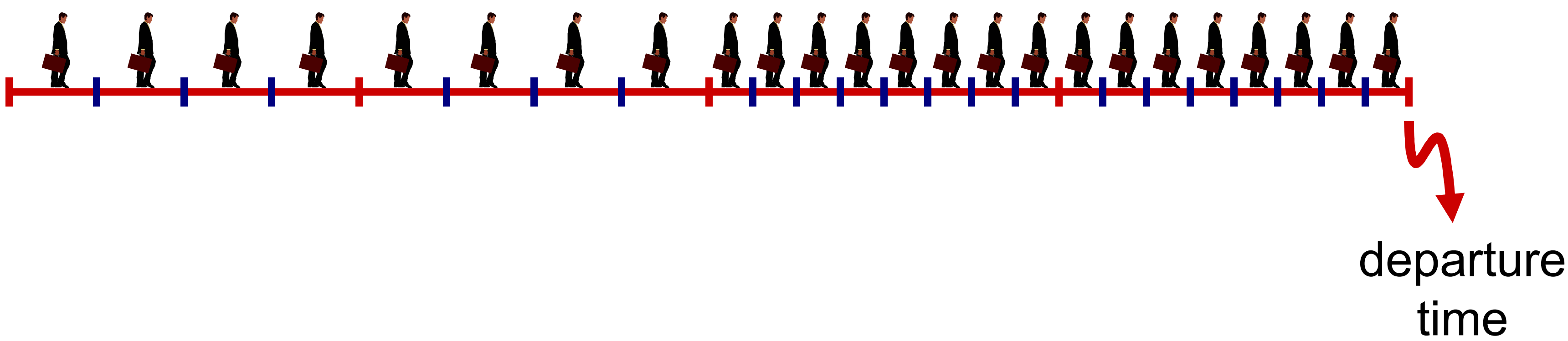
# Network Revenue Management



- Requests arrive over time
- Make an acceptance or rejection decision for each request
- Accepted requests use capacities on one or more flight legs

# Network Revenue Management

- $L$ : Set of flight legs
- $J$ : Set of itineraries
- $p_{jt}$ : Probability of observing a request for itinerary $j$ at time $t$
- $a_{ij}$ : 1 if itinerary $j$ uses flight leg $i$
  0 otherwise
- $f_j$ : Revenue from itinerary $j$
- $x_{it}$ : Remaining capacity on flight leg $i$ at time $t$



# Deterministic LP

- Assume that the numbers of itinerary requests take their expected values

$$LP = \max \sum_{j \in \mathcal{J}} f_j \, w_j$$

$$\sum_{j \in \mathcal{J}} a_{ij} \, w_j \leq x_{i1} \quad i \in \mathcal{L} \quad (\mu_i)$$

$$0 \leq w_j \leq \sum_{t=1}^{\tau} p_{jt} \quad j \in \mathcal{J}$$

- Simpson (1989), Williamson (1992), Talluri & Van Ryzin (2004)

# Control Policy from Deterministic LP

- $\{\mu_i : i \in \mathcal{L}\}$ are bid prices
- They capture the opportunity cost of a seat
- Accept a request for itinerary $j$ if

$$f_j \geq \sum_{i \in \mathcal{L}} a_{ij}\, \mu_i$$

…subject to capacity availability

- As time progresses, it is possible to refresh the bid prices by reoptimizing the deterministic LP
- The optimal objective value of the deterministic LP is an upper bound on the performance of any nonanticipatory policy

# Dynamic Programming Formulation

- $L$ : Set of flight legs
- $J$ : Set of itineraries
- $p_{jt}$ : Probability of observing a request for itinerary $j$ at time $t$
- $a_{ij}$ : 1 if itinerary $j$ uses flight leg $i$
  0 otherwise
- $f_j$ : Revenue from itinerary $j$
- $x_{it}$ : Remaining capacity on flight leg $i$ at time $t$
- $u_{jt}$ : 1 if a request for itinerary $j$ is accepted at time $t$
  0 otherwise

# Dynamic Programming Formulation

$$V_t(x_t) = \max \sum_{j \in \mathcal{J}} p_{jt} \Big[ f_j\, u_{jt} + V_{t+1}(x_t - u_{jt} \textstyle\sum_{i \in \mathcal{L}} a_{ij}\, e_i) \Big]$$

$$\begin{aligned} & a_{ij}\, u_{jt} \le x_{it} && i \in \mathcal{L},\ j \in \mathcal{J} \\ & u_{jt} \in \{0, 1\} && j \in \mathcal{J} \end{aligned}$$

# Dynamic Programming Formulation

$$V_t(x_t) = \max \sum_{j \in \mathcal{J}} p_{jt} \Big[ f_j \, u_{jt} + V_{t+1}\big(x_t - u_{jt} \textstyle\sum_{i \in \mathcal{L}} a_{ij} \, e_i\big) \Big]$$

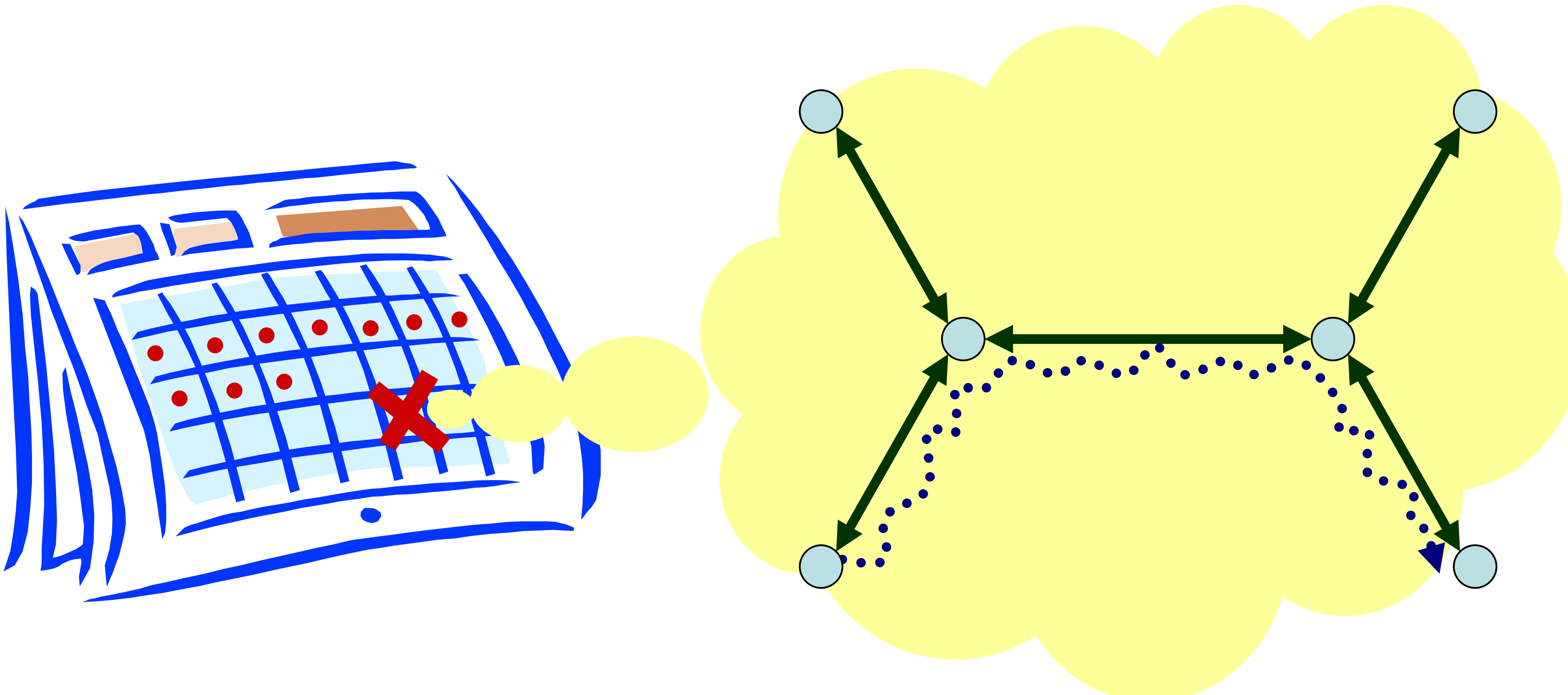
$$\begin{aligned} a_{ij} \, u_{jt} &\leq x_{it} && i \in \mathcal{L}, \; j \in \mathcal{J} \\ u_{jt} &\in \{0, 1\} && j \in \mathcal{J} \end{aligned}$$

$$V_t(s) = \max_{a \in \mathcal{A}(s)} r(s,a) + \sum_{j \in \mathcal{S}} p(j \,|\, s, a) V_{t+1}(j)$$

set of feasible actions

state space

# Relaxations in Network Revenue Management



- Limited capacities on the flight legs
- We need to make an “all or nothing” decisions when accepting itinerary requests

# Relaxations in Network Revenue Management

- Relax complicating constraints by associating Lagrange multipliers with them
- Minimize a dual function to find a good set of Lagrange multipliers
- Formalized for dynamic programs in Hawkins (2003) and Adelman & Mersereau (2006)

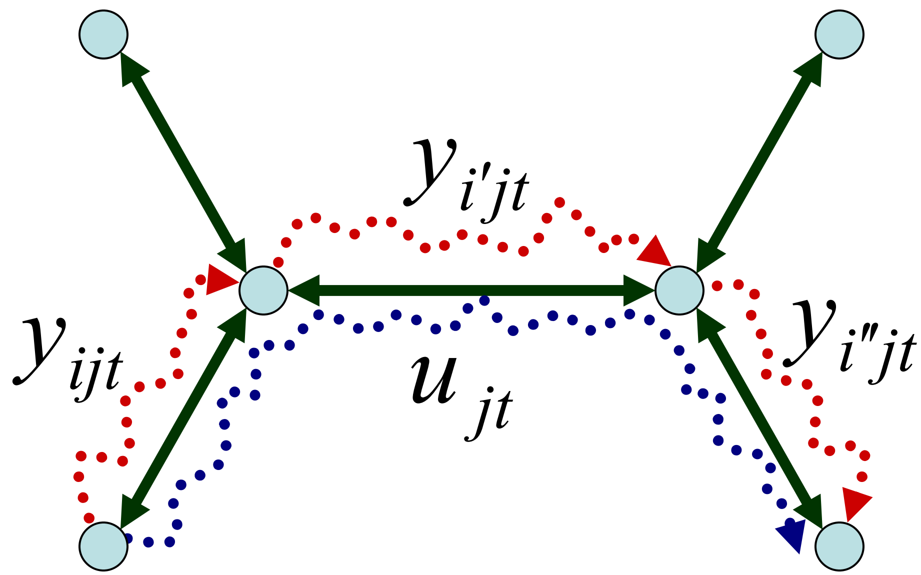
# Dynamic Programming Formulation

$$V_t(x_t) = \max \sum_{j \in \mathcal{J}} p_{jt} \Big[ f_j \, u_{jt} + V_{t+1}\big(x_t - u_{jt} \textstyle\sum_{i \in \mathcal{L}} a_{ij} \, e_i\big) \Big]$$

$$\begin{cases} a_{ij} \, u_{jt} \leq x_{it} & i \in \mathcal{L}, \;\; j \in \mathcal{J} \\ u_{jt} \in \{0, 1\} & j \in \mathcal{J} \end{cases}$$

$$\{y_{ijt} : i \in \mathcal{L}, \;\; j \in \mathcal{J}\}$$

$$\begin{cases} a_{ij} \, y_{ijt} \leq x_{it} & i \in \mathcal{L}, \;\; j \in \mathcal{J} \\ y_{ijt} = y_{i'jt} & i, i' \in \mathcal{L}, \;\; j \in \mathcal{J} \\ y_{ijt} \in \{0, 1\} & j \in \mathcal{J} \end{cases}$$


$y_{i'jt}$
$y_{ijt}$
$u_{jt}$
$y_{i''jt}$

# Dynamic Programming Formulation

$$a_{ij}\, y_{ijt} \le x_{it} \quad i \in \mathcal{L},\ j \in \mathcal{J}$$
$$y_{ijt} = y_{i'jt} \quad i, i' \in \mathcal{L},\ j \in \mathcal{J}$$
$$y_{ijt} \in \{0,1\} \quad j \in \mathcal{J}$$

$$x_1 = x_2$$
$$x_1 = x_3$$
$$x_2 = x_1$$
$$x_2 = x_3 \ldots$$

$$\Downarrow$$

$$a_{ij}\, y_{ijt} \le x_{it} \quad i \in \mathcal{L},\ j \in \mathcal{J}$$
$$y_{ijt} = y_{\phi jt} \quad i \in \mathcal{L} \cup \{\phi\},\ j \in \mathcal{J}$$
$$y_{ijt} \in \{0,1\} \quad j \in \mathcal{J}$$

$$x_1 = x_\phi$$
$$x_2 = x_\phi$$
$$x_3 = x_\phi$$

# Lagrangian Relaxation

$$V_t(x_t) = \max \sum_{j \in \mathcal{J}} p_{jt} \Big[ f_j \, y_{\phi jt} + V_{t+1}\big(x_t - \textstyle\sum_{i \in \mathcal{L}} a_{ij} \, y_{ijt} \, e_i\big) \Big]$$

$$a_{ij} \, y_{ijt} \leq x_{it} \qquad i \in \mathcal{L}, \; j \in \mathcal{J}$$

$$y_{ijt} = y_{\phi jt} \qquad i \in \mathcal{L}, \; j \in \mathcal{J}$$

$$y_{ijt} \in \{0, 1\} \qquad i \in \mathcal{L} \cup \{\phi\}, \; j \in \mathcal{J}$$

$$\{\lambda_{ijt} : i \in \mathcal{L}, \; j \in \mathcal{J}\}$$

# Lagrangian Relaxation

$$V_t^{\lambda}(x_t) = \max \sum_{j \in \mathcal{J}} p_{jt} \Big\{ \Big[ f_j - \sum_{i \in \mathcal{L}} \lambda_{ijt} \Big] \, y_{\phi jt} + \sum_{i \in \mathcal{L}} \lambda_{ijt} \, y_{ijt}$$

$$+ V_{t+1}^{\lambda}(x_t - \textstyle\sum_{i \in \mathcal{L}} a_{ij} \, y_{ijt} \, e_i) \Big\}$$

$$a_{ij} \, y_{ijt} \leq x_{it} \qquad i \in \mathcal{L}, \; j \in \mathcal{J}$$

$$y_{ijt} \in \{0, 1\} \qquad i \in \mathcal{L} \cup \{\phi\}, \; j \in \mathcal{J}$$

- This dynamic program decomposes by the flight legs…

# Lagrangian Relaxation

- The value function for flight leg $i$ has the form

$$v_{it}^{\lambda}(x_{it}) = \max \sum_{j \in \mathcal{J}} p_{jt} \Big\{ \lambda_{ijt}\, y_{ijt} + v_{i,t+1}^{\lambda}(x_{it} - a_{ij}\, y_{ijt}) \Big\}$$

$$\begin{aligned} a_{ij}\, y_{ijt} &\leq x_{it} \qquad && j \in \mathcal{J} \\ y_{ijt} &\in \{0, 1\} \qquad && j \in \mathcal{J} \end{aligned}$$

- Putting it all together

$$V_t^{\lambda}(x_t) = \sum_{t'=t}^{\tau} \sum_{j \in \mathcal{J}} p_{j't} \Big[ f_j - \sum_{i \in \mathcal{L}} \lambda_{ijt'} \Big]^{+} + \sum_{i \in \mathcal{L}} v_{it}^{\lambda}(x_{it})$$

# Properties of Lagrangian Relaxation

- We obtain upper bounds on the value function

$$V_t(x_t) \leq V_t^{\lambda}(x_t)$$
$$V_1(x_1) \leq V_1^{\lambda}(x_1)$$

- To find a good set of Lagrange multipliers, we solve

$$V_1(x_1) \leq \min_{\lambda} V_1^{\lambda}(x_1)$$

convex in $\lambda$

- Lagrangian relaxation gives a tighter bound than deterministic LP

$$V_1(x_1) \leq \min_{\lambda} V_1^{\lambda}(x_1) \leq LP$$

# Control Policy from Lagrangian Relaxation

- The value function approximation at time $t$ is

$$V_t^\lambda(x_t) = \sum_{t'=t}^{\tau} \sum_{j \in \mathcal{J}} p_{j't} \Big[ f_j - \sum_{i \in \mathcal{L}} \lambda_{ijt'} \Big]^+ + \sum_{i \in \mathcal{L}} v_{it}^\lambda(x_{it})$$

- Greedy policy solves the problem

$$\max \sum_{j \in \mathcal{J}} p_{jt} \Big[ f_j \, u_{jt} + V_{t+1}^\lambda(x_t - u_{jt} \textstyle\sum_{i \in \mathcal{L}} a_{ij} \, e_i) \Big]$$

$$a_{ij} \, u_{jt} \leq x_{it} \qquad i \in \mathcal{L}, \; j \in \mathcal{J}$$

$$u_{jt} \in \{0, 1\} \qquad j \in \mathcal{J}$$

# Greedy Policy from Lagrangian Relaxation

$$\max \ f_j\, u_{jt} + \sum_{i \in \mathcal{L}} v^{\lambda}_{i,t+1}(x_{it} - a_{ij}\, u_{jt})$$

$$a_{ij}\, u_{jt} \leq x_{it} \qquad i \in \mathcal{L}$$

$$u_{jt} \in \{0, 1\}$$

- Accept a request for itinerary $j$ if

$$f_j + \sum_{i \in \mathcal{L}} v^{\lambda}_{i,t+1}(x_{it} - a_{ij}) \geq \sum_{i \in \mathcal{L}} v^{\lambda}_{i,t+1}(x_{it})$$

…subject to capacity availability

$$f_j \geq \sum_{i \in \mathcal{L}} a_{ij} \left[ v^{\lambda}_{i,t+1}(x_{it}) - v^{\lambda}_{i,t+1}(x_{it} - 1) \right]$$

bid-price

# Control Policy from Deterministic LP

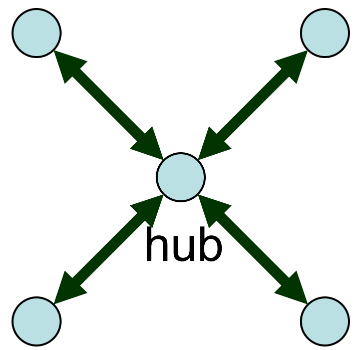
- $\{\mu_i : i \in \mathcal{L}\}$ are bid prices
- They capture the opportunity cost of a seat
- Accept a request for itinerary $j$ if

$$f_j \geq \sum_{i \in \mathcal{L}} a_{ij}\, \mu_i$$

  …subject to capacity availability
- As time progresses, it is possible to refresh the bid prices by reoptimizing the deterministic LP
- The optimal objective value of the deterministic LP is an upper bound on the performance of any nonanticipatory policy

# Computational Results

- Hub serving $N$ spokes with $2N$ flights



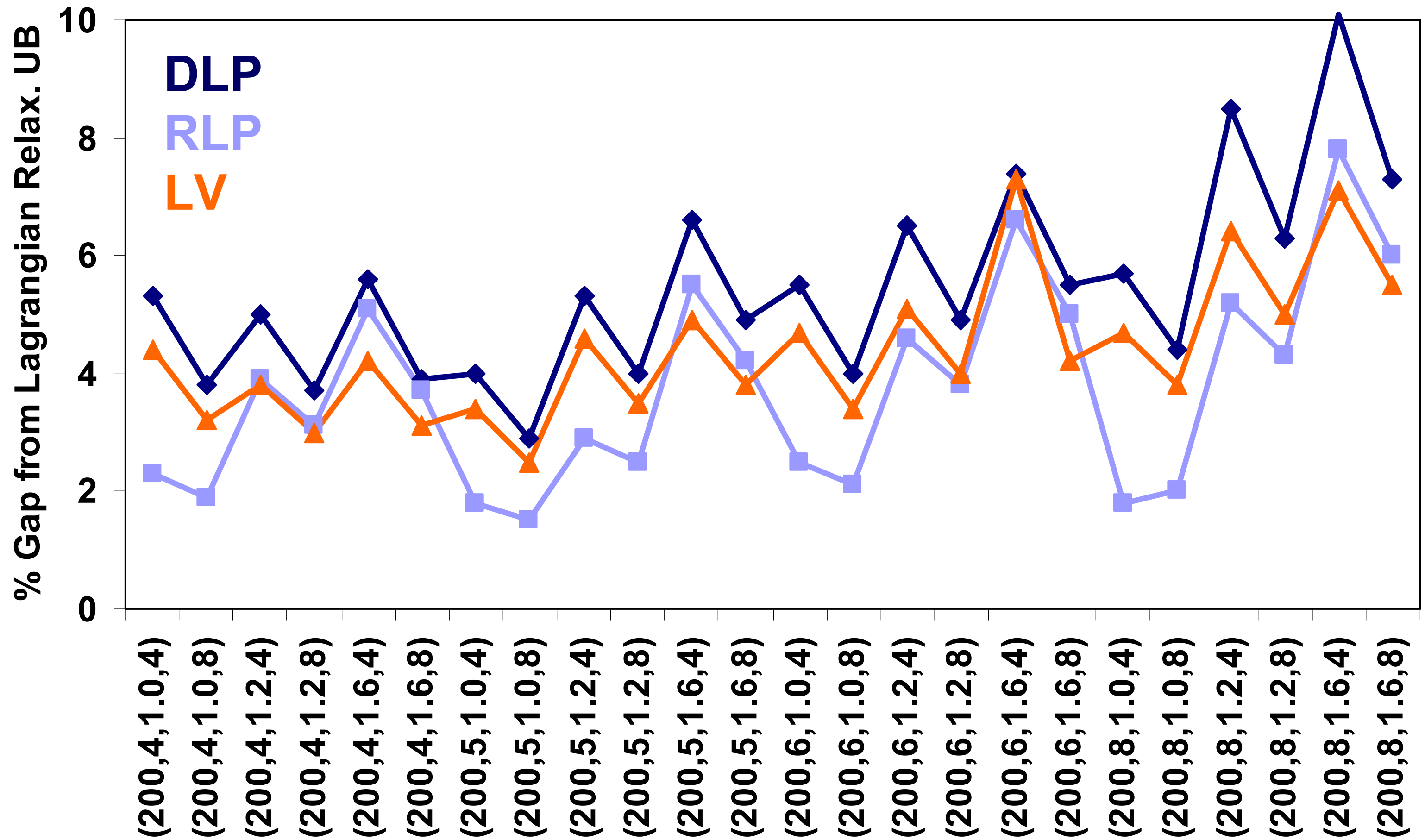

- A high-fare and a low-fare itinerary connecting each pair of locations
- Probability of having a request for a high-fare itinerary increases over time

# Computational Results

DLP : Deterministic LP
RLP : Randomized LP of Talluri & Van Ryzin (1999)
FD : Finite differences on deterministic LP of Bertsimas & Popescu (2003)
LV : Linear value function approximation of Adelman (2006)

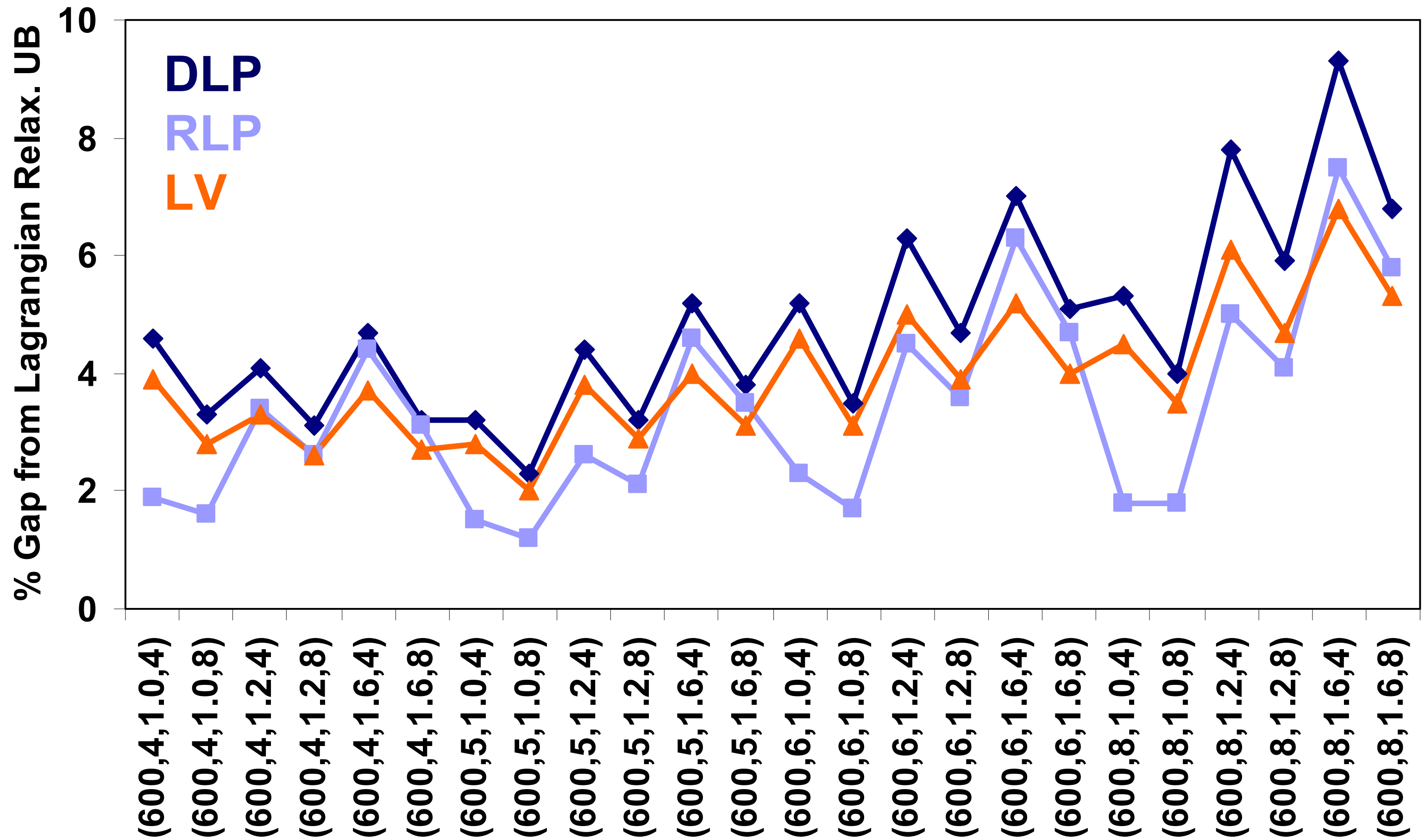
DLP, RLP and LV also provide upper bounds on the optimal total expected revenue

# Comparing the Upper Bounds



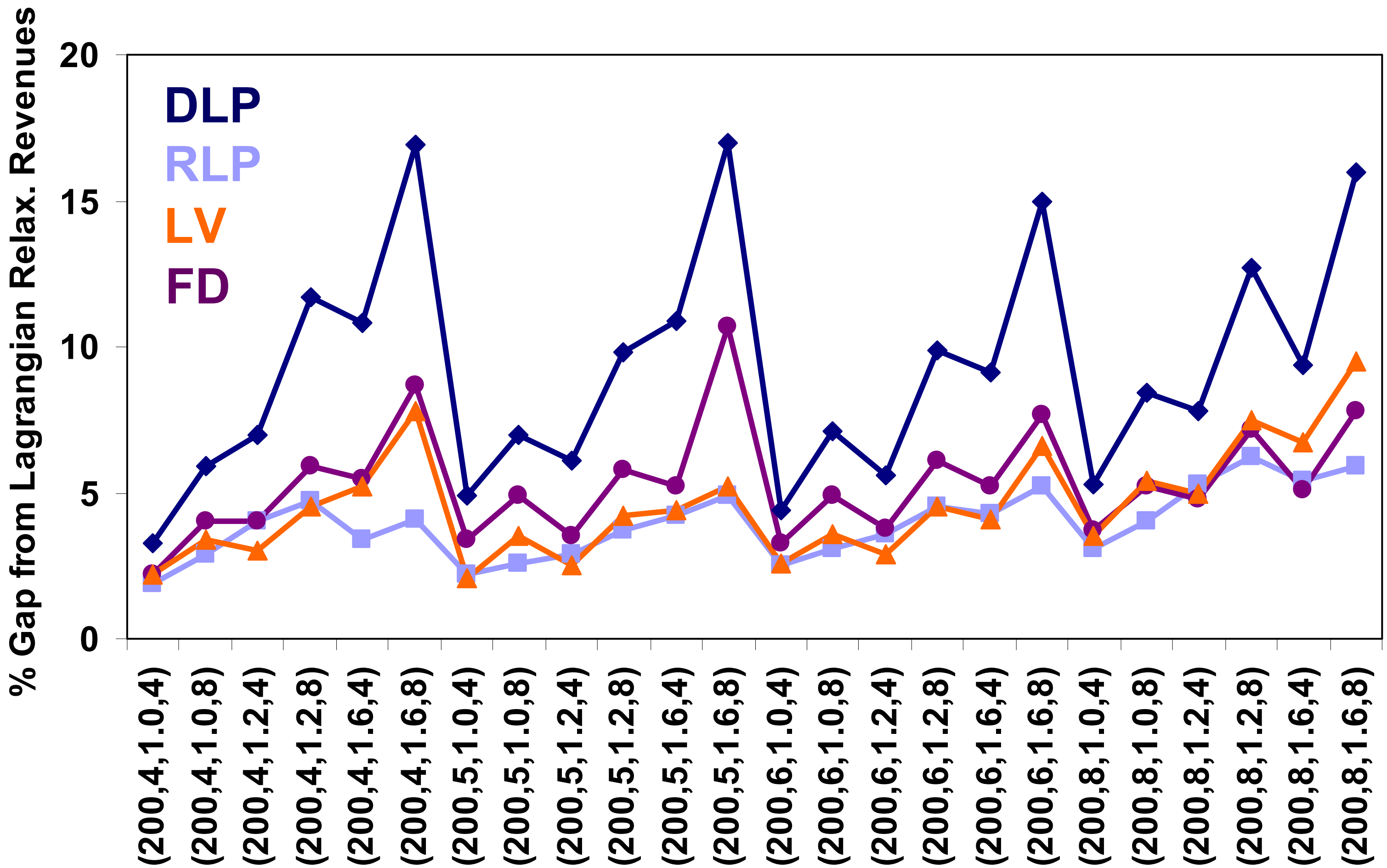

(no time periods, no spokes, congestion, fare difference)

# Comparing the Upper Bounds



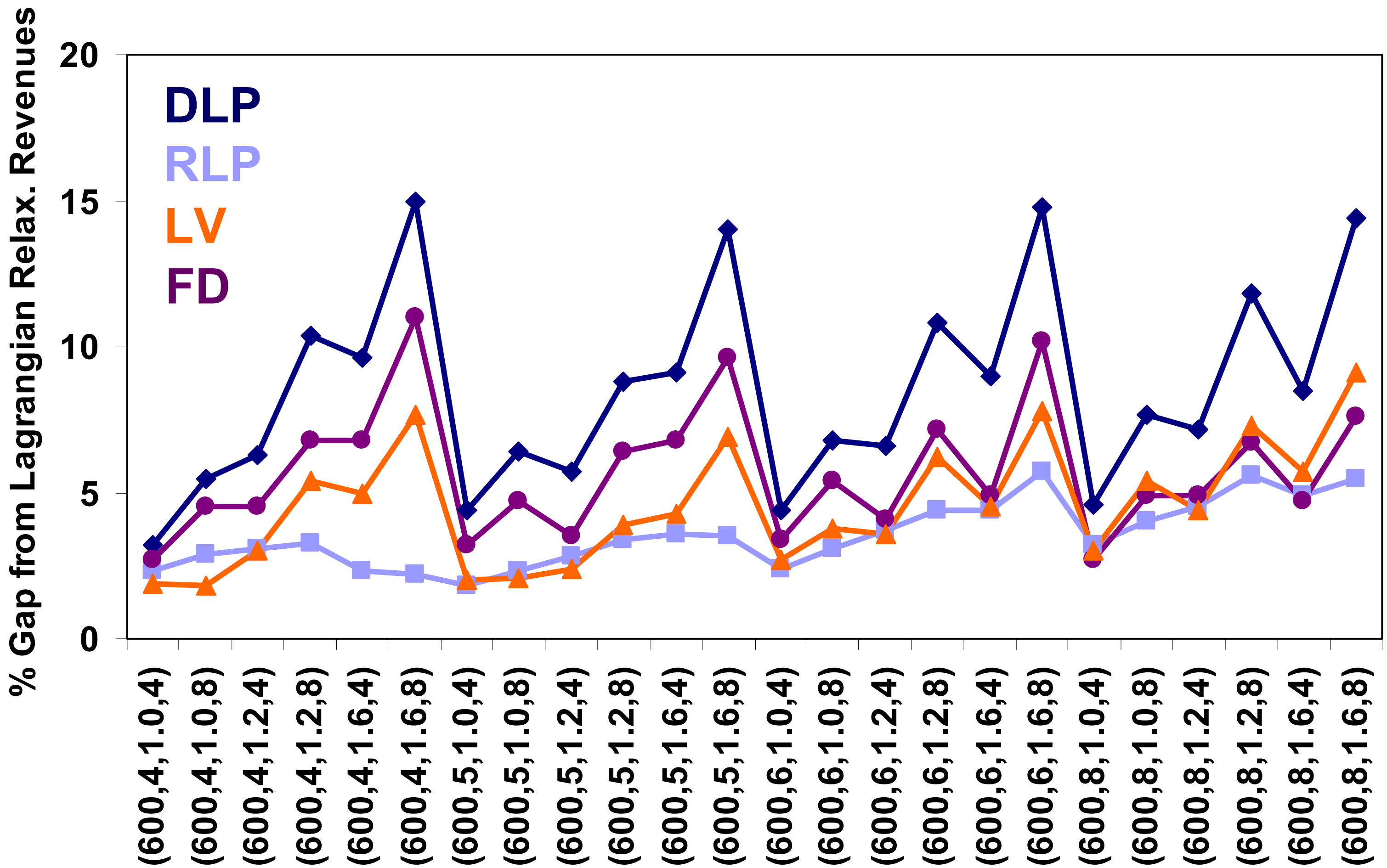

(no time periods, no spokes, congestion, fare difference)

# Comparing the Performance




(no time periods, no spokes, congestion, fare difference)

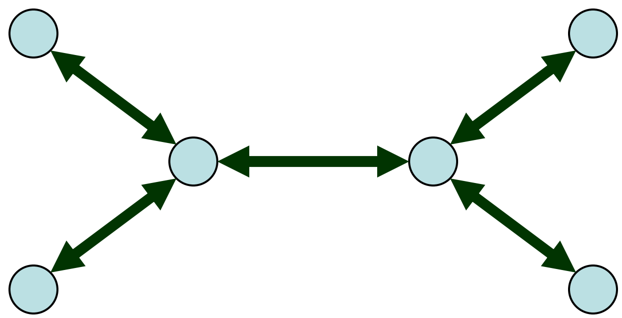
# Comparing the Performance




(no time periods, no spokes, congestion, fare difference)

# Comments

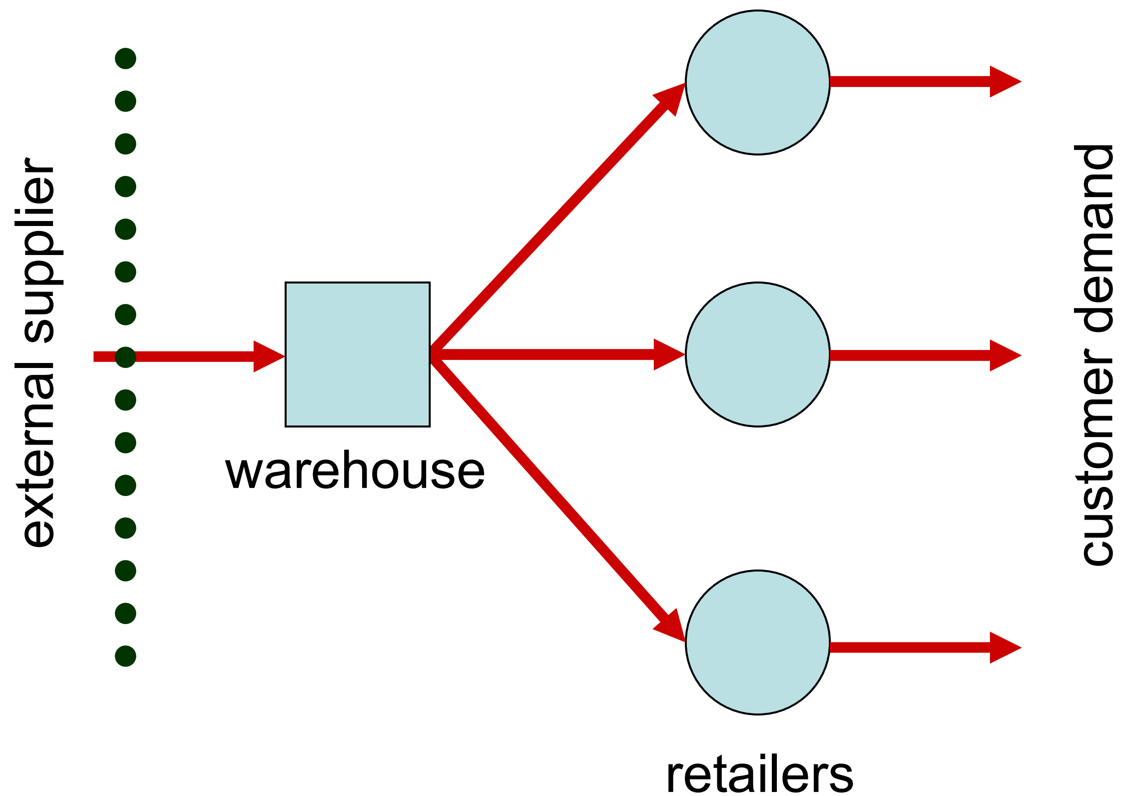
- Concrete foundation for decomposing the network revenue management problem by flight legs



- Provides good performance
- Requires considerably more computational power and storage
- Storage requirements can be alleviated through a stochastic implementation

# Comments

- Applicable to other problem classes



# Comments

- Applicable to other problem classes