ORIE 4154 - Pricing and Market Design

Module 2: Network RM and Approximate DP
(The Network RM Dynamic Program)

Instructor: Sid Banerjee, ORIE

Cornell University

From Last Class: The Network RM Problem

Pricing/admission control for products using interlinked resources.

Connecting flights, multi-day hotel bookings, project teams, etc.

- Resources
  - Perishable units of capacity managed by firm
    E.g. Seats on a flight, hotel room nights, employee hours
  - Constrained (C_i units of resource i)
  - Perishable (each resource expires at a certain time)
- Product
  - Bundle of resources for selling to customer
    E.g. multi-leg flight, multiple days stay at hotel
  - Each product has a specified set of resources and price

What we need

- Compact representation for DP formulation

From Last Class: The Network RM Problem

Pricing/admission control for products using interlinked resources.

Connecting flights, multi-day hotel bookings, project teams, etc.

- Resources
  - Perishable units of capacity managed by firm
    E.g. Seats on a flight, hotel room nights, employee hours
  - Constrained (C_i units of resource i)
  - Perishable (each resource expires at a certain time)
- Product
  - Bundle of resources for selling to customer
    E.g. multi-leg flight, multiple days stay at hotel
  - Each product has a specified set of resources and price

What we need

- Compact representation for DP formulation
- Good approximations for solving the DP

Formal Model for Network RM

Basic Setting

- Time periods $\{1,2,\ldots,T\}$
- m resources $\{1,2,\ldots,m\}$
- Resource i has initial capacity c_i; expires at T

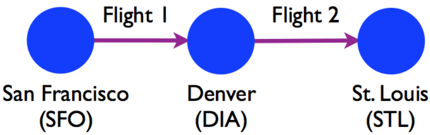
Products

- n unique products $\{0,1,\ldots,n\}$
- Product $j \equiv$ (price p_j, resource reqt A_j)
- Incidence vector $A_j = \{a_{ij}\}$, where $a_{ij} = \mathbb{1}_{\{j \text{ uses resource } i\}}$

More generally, $a_{ij} = \#$ of units of resource i used by product j
Incidence Matrix: $A = [A_1, A_2, \ldots, A_n]$ ($m \times n$ matrix)

Incidence Matrix: Example 1

- 2 resources
  - Flight 1 from SFO to DIA
  - Flight 2 from DIA to STL
- Suppose we have 3 products and two fare classes: 6 ODFs
  - SFO to DIA full fare
  - SFO to DIA discount fare
  - DIA to STL full fare
  - DIA to STL discount fare
  - SFO to STL full fare
  - SFO to STL discount fare



Incidence Matrix

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

Courtesy: Paat Rusmevichentong

Incidence Matrix: Example 2

- Suppose a hotel offers 1-night, 2-night, and 3-night stays only.
- Resources: Room nights
- Products: Combinations of arrival night and length of stay.
  - Assume one fare class for each product, that is, number of ODFs is the same as the number of products.
- Incidence matrix for 1 week of resources.

Resources/ Length of Stay	Arrival Date														
	Sunday			Monday			Tuesday			Wednesday			Thursday		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
Sunday	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Monday	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Tuesday	0	0	1	0	1	1	1	1	1	0	0	0	0	0	0
Wednesday	0	0	0	0	0	1	0	1	1	1	1	1	0	0	0
Thursday	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1
Friday	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
Saturday	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Incidence Matrix: Example 2 (some caveats)

- In theory, the network management problem for a hotel stretches out indefinitely into the future.
- In practice, a hotel will only accept booking for some limited period into the future (often a year), limiting the number of resources and products that need to be managed.
- Moreover, in theory a hotel offers and infinite number of products, since customers can buy any length of stay.
- In practice, lengths of stay longer than 14 nights are extremely rare at most hotels, and hotels usually managed them as a single product.

Courtesy: Paat Rusmevichentong

The Network RM Dynamic Program

- m resources, n products, time periods $\{1, 2, \ldots, T\}$
- Resource i has initial capacity c_i
- Product $j \equiv$ (price p_j, resource reqt A_j), where $a_{ij} = \mathbb{1}_{\{j \text{ uses resource } i\}}$

Dynamics and actions

- In period t, state of system is $\underline{\mathbf{x}} = [x_1, x_2, \ldots, x_n]$
- *At most one* request arrives in each period:
  - No request arrives with probability λ_0
  - Request for product j arrives with probability λ_j
  - $\sum_{j=0}^{n} \lambda_j = 1$
- Action/policy vector: $\underline{\mathbf{u}}(t, \underline{\mathbf{x}}) = [u_1(t, \underline{\mathbf{x}}), u_2(t, \underline{\mathbf{x}}), \ldots, u_n(t, \underline{\mathbf{x}})]$,

$$u_j(t, \underline{\mathbf{x}}) = \mathbb{1}_{\{\text{Sell product } j \text{ in period } t \text{ with initial state } \underline{\mathbf{x}}\}}$$

The Network RM Dynamic Program

- m resources, n products, time periods $\{1,2,\ldots,T\}$
- Resource i has initial capacity c_i
- Product $j \equiv (p_j, A_j)$, where $a_{ij} = \mathbb{1}_{\{j \text{ uses resource } i\}}$
- Request for product j arrives w.p. λ_j; no request w.p λ_0
- $u_j(t,\underline{\mathbf{x}}) = \mathbb{1}_{\{\text{Sell product } j \text{ in period } t \text{ with initial state } \underline{\mathbf{x}}\}}$

For any $t, \underline{\mathbf{x}}$, value function $V_t(\underline{\mathbf{x}})$ denotes the maximum expected revenue that can be obtained from period t until T given a remaining capacity $\underline{\mathbf{x}}$ among the m resources

The Bellman Equation

$$V_t(\underline{\mathbf{x}}) = \lambda_0 V_{t+1}(\underline{\mathbf{x}}) + \sum_{j=1}^{n} \lambda_j \max_{u \in \{0,1\}: u.A_j \leq \underline{\mathbf{x}}} \left\{ u \cdot p_j + V_{t+1}(\underline{\mathbf{x}} - u \cdot A_j) \right\}$$

where $u \cdot A_j \leq \underline{\mathbf{x}}$ is equivalent to $u \cdot a_{ij} \leq x_i \forall i$

The Network RM Dynamic Program: Optimal Policy

The Bellman Equation

$$V_t(\underline{\mathbf{x}}) = \lambda_0 V_{t+1}(\underline{\mathbf{x}}) + \sum_{j=1}^{n} \lambda_j \max_{u \in \{0,1\} : u \cdot A_j \leq \underline{\mathbf{x}}} \left\{ u \cdot p_j + V_{t+1}(\underline{\mathbf{x}} - u \cdot A_j) \right\}$$

Suppose we are given $V_{t+1}(\underline{\mathbf{x}})$ for all states $\underline{\mathbf{x}}$. Then we can solve for the optimal policy $\underline{\mathbf{u}}^{\star}(t, \underline{\mathbf{x}})$ to get for each product j:

$$u_j^{\star}(t, \underline{\mathbf{x}}) = \begin{cases} 1 & \text{if } p_j > V_{t+1}(\underline{\mathbf{x}}) - V_{t+1}(x - A_j), \text{ AND } A_j \leq \underline{\mathbf{x}} \\ 0 & \text{otherwise} \end{cases}$$

Intuition: Sell a product only if its price exceeds the opportunity cost of losing future sales due to reduction in resource capacities.

The Curse of Dimensionality

Thus, we have the optimal policy $\underline{\mathbf{u}}^\star(t, \underline{\mathbf{x}})$

$$u_j^\star(t, \underline{\mathbf{x}}) = \begin{cases} 1 & \text{if } p_j > V_{t+1}(\underline{\mathbf{x}}) - V_{t+1}(x - A_j), \text{ AND } A_j \leq \underline{\mathbf{x}} \\ 0 & \text{otherwise} \end{cases}$$

Thus, given $V_t(\underline{\mathbf{x}})$ for all $t, \underline{\mathbf{x}}$, we can find the optimal policy.

- However, $\underline{\mathbf{x}}$ can take $\prod_{i=1}^n (c_i + 1) \approx (c+1)^m$ states!
- This is the curse of dimensionality: infeasible to store $V_t(\underline{\mathbf{x}})$ exactly for moderately sized problems

To get around this, we want good approximations of $V_t(\underline{\mathbf{x}})$