

# Seat Planning and Seat Assignment with Social Distancing

Zikang LI

Department of Industrial Engineering and Decision Analytics  
The Hong Kong University of Science and Technology

Supervisors: Prof. Xiangtong QI  
Prof. Qian LIU

Aug 1, 2024

# Table of Contents

- 1 Introduction
- 2 Literature Review
- 3 Problem Description
- 4 Seat Planning with Stochastic Demand
- 5 Seat Assignment with Dynamic Demand
- 6 Conclusion and Future Work

# Introduction

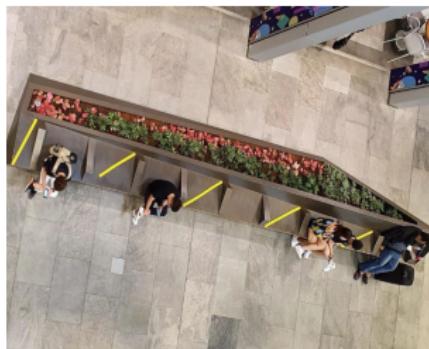
# Social Distancing under Pandemic

- Social distancing measures



# Social Distancing under Pandemic

- Social distancing in seating areas

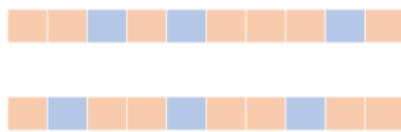


# Seat Planning and Seat Assignment

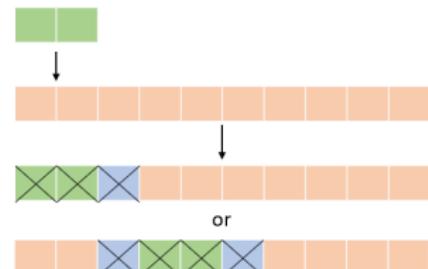
Social distancing requirement:

- The size of a group is confined.
- People in the same group sit together.
- Different groups should keep distance.

Seat planning:



Seat assignment:



# Overview

## ■ Seat planning

- **Deterministic demand**: make the seat planning with the known specific demand for groups.
- **Stochastic demand**: make the seat planning with the known demand distribution before the realization of demand.

## ■ Seat assignment with dynamic demand

- Assign seats to each group under **flexible** seat planning.
  - **Real-time seat assignment**
  - Accept or reject each group after its realization, but assign them later.
- Assign seats to each group under **fixed** seat planning.

# Contributions

- Seat planning:

- New model and technique for stochastic demand
- Provide seat planning as a guidance for seat assignment

- Seat assignment:

- New model for seat assignment problem
- Provide practical solution and insights

# Literature Review

# Seat Planning with Social Distancing

- Applications:

Airplanes seat planning (Ghorbani et.al 2020)

Classroom layout planning (Bortolete et al. 2022)

Trains seat planning (Haque & Hamid 2022).

- Group-based seat planning for known groups:

Amphitheaters (Haque & Hamid 2022)

Airplanes (Salari et al. 2022)

Theaters (Blom et al. 2022).

Our work considers seat planning with stochastic demand.

# Dynamic Seat Assignment

- Dynamic multiple knapsack problem:
  - Multiple knapsack problem (Pisinger et al. 1999)
  - Dynamic knapsack problem (Kleywegt et al. 1998)
- Network revenue management: Group-based (Talluri et al. 2006).
- Dynamic capacity control: Assign-to-seat for selling high-speed train tickets (Zhu et al. 2023)

Our work considers the group-based seat assignment.

# Problem Description

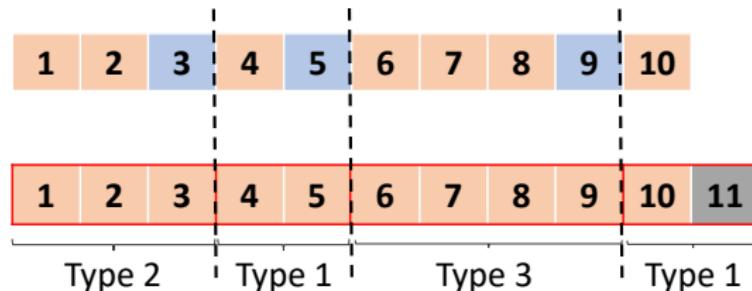
# Seat Planning with Social Distancing

- Group type  $\mathcal{M} = \{1, \dots, M\}$ .
- Row  $\mathcal{N} = \{1, \dots, N\}$ .
- Number of seats in row  $j$ :  $L_j^0, j \in \mathcal{N}$ .
- Social distancing:  $\delta$  seat(s).
  - $n_i = i + \delta$ : the size of type  $i$  group for each  $i \in \mathcal{M}$ .
  - $L_j = L_j^0 + \delta$ : the size of row  $j$  for each  $j \in \mathcal{N}$ .



# Pattern

- Pattern:  $\mathbf{h} = (h_1, \dots, h_M)$ , the seat planning for one row, where  $h_i$  is the number of type  $i$  groups.
- Feasible pattern:  $\sum_{i=1}^M n_i h_i \leq L$ .
- Maximum number of people accommodated:  $|\mathbf{h}| = \sum_{i=1}^M i h_i$ .



$$L = 11, \delta = 1, M = 4.$$

$$\mathbf{h} = (2, 1, 1, 0), |\mathbf{h}| = 7.$$

# Largest and Full Patterns

- **Largest** pattern:

$\mathbf{h}$  is a largest pattern if  $|\mathbf{h}| \geq |\mathbf{h}'|$  for any feasible  $\mathbf{h}'$ .

$|\mathbf{h}| = qM + \max\{r - \delta, 0\}$ , where  $q = \lfloor L/n_M \rfloor$ ,  $r = L - q \cdot n_M$ .

- **Full** pattern:

$\mathbf{h}$  is a full pattern if  $\sum_{i=1}^M n_i h_i = L$ .

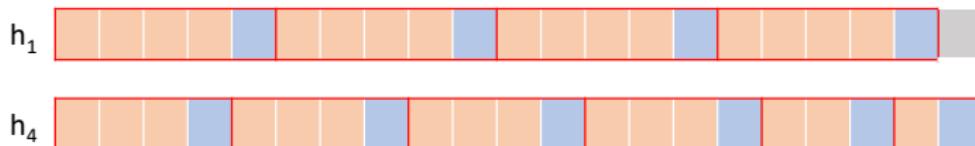
## Example:

$\delta = 1$ ,  $M = 4$ ,  $L = 21$ .

Largest patterns:  $h_1 = (0, 0, 0, 4)$ ,  $h_2 = (0, 0, 4, 1)$ ,  $h_3 = (0, 2, 0, 3)$ ,  
 $|h_1| = |h_2| = |h_3| = 16$ .

Largest but not full:  $h_1 = (0, 0, 0, 4)$ .  $\sum_{i=1}^M n_i h_i \neq L$

Full but not largest:  $h_4 = (1, 1, 4, 0)$ .  $|h_4| = 15 < 16$



# Problem Formulation

Goal: obtain the seat planning composed of full or largest patterns while maximizing the number of people accommodated.

Seat planning problem with given demand  $\mathbf{d} = (d_1, \dots, d_M)$ :

$$\begin{aligned}
 \max \quad & \sum_{i=1}^M \sum_{j=1}^N (n_i - \delta) x_{ij} \\
 \text{s.t.} \quad & \sum_{j=1}^N x_{ij} \leq d_i, \quad i \in \mathcal{M}, \\
 & \sum_{i=1}^M n_i x_{ij} \leq L_j, \quad j \in \mathcal{N}, \\
 & x_{ij} \in \mathbb{N}, \quad i \in \mathcal{M}, j \in \mathcal{N}.
 \end{aligned} \tag{1}$$

Objective: maximize the number of people accommodated.

$x_{ij}$ : the number of type  $i$  groups in row  $j$ .

# Property

In the LP relaxation of problem (1), there is a threshold  $\tilde{i} \in \{0, 1, \dots, M\}$  such that the optimal solutions satisfy the following conditions:

- For  $i = 1, \dots, \tilde{i} - 1$  and all  $j$ ,  $x_{ij}^* = 0$ .
- For  $i = \tilde{i} + 1, \dots, M$ ,  $\sum_j x_{ij}^* = d_i$ .
- For  $i = \tilde{i}$ ,  $\sum_j x_{ij}^* = \frac{L - \sum_{i=\tilde{i}+1}^M d_i n_i}{n_{\tilde{i}}}$

The seat planning obtained from problem (1) may not utilize all seats.

We aim to obtain the **seat planning composed of full or largest patterns**.

# Generate the Full or Largest Patterns

Original seat planning:  $\mathbf{H}$ .

Desired feasible seat planning:  $\mathbf{H}'$ .

Original group type requirement:  $\sum_{k=i}^M \sum_{j=1}^N H'_{ji} \geq \sum_{k=i}^M \sum_{j=1}^N H_{ji}, \forall i \in \mathcal{M}$ .

$$\begin{aligned}
 \max \quad & \sum_{i=1}^M \sum_{j=1}^N (n_i - \delta) x_{ij} \\
 \text{s.t.} \quad & \sum_{j=1}^N \sum_{k=i}^M x_{kj} \geq \sum_{k=i}^M \sum_{j=1}^N H_{ji}, i \in \mathcal{M} \\
 & \sum_{i=1}^M n_i x_{ij} \leq L_j, j \in \mathcal{N} \\
 & x_{ij} \in \mathbb{N}, i \in \mathcal{M}, j \in \mathcal{N}
 \end{aligned} \tag{2}$$

Property:  $\mathbf{H}'$  corresponding to the optimal solution to problem (2) is composed of **full or largest patterns**.

# Seat Planning with Stochastic Demand

# Method Flow

We aim to obtain a seat planning with known demand scenarios.

- Build the formulation of Scenario-based Stochastic Programming (SSP).
  - Consider the nested relation: a smaller group can take the seats planned for the larger group.
- Reformulate SSP to the Benders Master Problem (BMP) and subproblem.
- The optimal solution can be obtained by solving BMP iteratively.

# Scenario-based Stochastic Programming (SSP)

Objective: maximize the expected number of people

$y_{i\omega}^+$ : excess supply for  $i, \omega$ .  $y_{i\omega}^-$ : shortage of supply for  $i, \omega$ .

$d_{i\omega}$ : demand of type  $i$  groups for scenario  $\omega$

$$\begin{aligned}
 \max \quad & E_\omega \left[ \sum_{i=1}^{M-1} (n_i - \delta) \left( \sum_{j=1}^N x_{ij} + y_{i+1,\omega}^+ - y_{i\omega}^+ \right) + (n_M - \delta) \left( \sum_{j=1}^N x_{Mj} - y_{M\omega}^+ \right) \right] \\
 \text{s.t.} \quad & \sum_{j=1}^N x_{ij} - y_{i\omega}^+ + y_{i+1,\omega}^+ + y_{i\omega}^- = d_{i\omega}, \quad i = 1, \dots, M-1, \omega \in \Omega \\
 & \sum_{j=1}^N x_{ij} - y_{i\omega}^+ + y_{i\omega}^- = d_{i\omega}, \quad i = M, \omega \in \Omega \\
 & \sum_{i=1}^M n_i x_{ij} \leq L_j, j \in \mathcal{N} \\
 & y_{i\omega}^+, y_{i\omega}^- \in \mathbb{N}, \quad i \in \mathcal{M}, \omega \in \Omega \\
 & x_{ij} \in \mathbb{N}, \quad i \in \mathcal{M}, j \in \mathcal{N}.
 \end{aligned} \tag{3}$$

$$\begin{aligned}
 & E_\omega \left[ \sum_{i=1}^{M-1} (n_i - \delta) \left( \sum_{j=1}^N x_{ij} + y_{i+1,\omega}^+ - y_{i\omega}^+ \right) + (n_M - \delta) \left( \sum_{j=1}^N x_{Mj} - y_{M\omega}^+ \right) \right] = \\
 & \sum_{j=1}^N \sum_{i=1}^M i \cdot x_{ij} - \sum_{\omega=1}^{|\Omega|} p_\omega \sum_{i=1}^M y_{i\omega}^+
 \end{aligned}$$

# Reformulation

$$\max \quad \mathbf{c}^T \mathbf{x} + \sum_{\omega \in \Omega} p_{\omega} \mathbf{f}^T \mathbf{y}_{\omega}$$

$$\text{s.t.} \quad \sum_{j=1}^N x_{ij} - y_{i\omega}^+ + y_{i+1,\omega}^+ + y_{i\omega}^- = d_{i\omega}, \\ i = 1, \dots, M-1, \omega \in \Omega$$

$$\sum_{j=1}^N x_{ij} - y_{i\omega}^+ + y_{i\omega}^- = d_{i\omega}, \\ i = M, \omega \in \Omega$$

$$\sum_{i=1}^M n_i x_{ij} \leq L_j, j \in \mathcal{N}$$

$$y_{i\omega}^+, y_{i\omega}^- \in \mathbb{N}, \quad i \in \mathcal{M}, \omega \in \Omega \\ x_{ij} \in \mathbb{N}, \quad i \in \mathcal{M}, j \in \mathcal{N}.$$

$$\mathbf{c}^T \mathbf{x} = \sum_{j=1}^N \sum_{i=1}^M i x_{ij}, \quad \mathbf{f}^T \mathbf{y}_{\omega} = - \sum_{i=1}^M y_{i\omega}^+.$$

SSP is equivalent to the following master problem

$$\begin{aligned} \max \quad & \mathbf{c}^T \mathbf{x} + z(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x}^T \mathbf{n} \leq \mathbf{L} \\ & \mathbf{x} \in \mathbb{N}^{M \times N, +}, \end{aligned} \quad (4)$$

where  $z(\mathbf{x})$  is defined as

$$z(\mathbf{x}) := E(z_{\omega}(\mathbf{x})) = \sum_{\omega \in \Omega} p_{\omega} z_{\omega}(\mathbf{x}),$$

and for each scenario  $\omega \in \Omega$ , the subproblem

$$\begin{aligned} z_{\omega}(\mathbf{x}) := \max \quad & \mathbf{f}^T \mathbf{y}_{\omega} \\ \text{s.t.} \quad & \mathbf{x} \mathbf{1} + \mathbf{V} \mathbf{y}_{\omega} = \mathbf{d}_{\omega} \\ & \mathbf{y} \geq 0. \end{aligned} \quad (5)$$

# Solution to Subproblem

Problem (5) is easy to solve with a given  $\mathbf{x}$  from the perspective of the dual problem:

$$\begin{aligned} \min \quad & \alpha_{\omega}^T (\mathbf{d}_{\omega} - \mathbf{x} \mathbf{1}) \\ \text{s.t.} \quad & \alpha_{\omega}^T \mathbf{V} \geq \mathbf{f}^T \end{aligned} \tag{6}$$

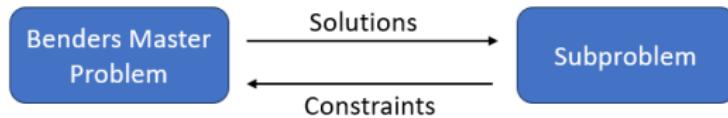
- The feasible region of problem (6),  $P = \{\alpha | \alpha^T \mathbf{V} \geq \mathbf{f}^T\}$ , is bounded.  
In addition, all the extreme points of  $P$  are integral.
- The optimal solution can be obtained directly according to the complementary slackness property.

# Benders Decomposition Procedure

Let  $z_\omega$  be the lower bound of problem (6), SSP can be obtained by solving following benders master problem:

$$\begin{aligned}
 \max \quad & \mathbf{c}^\top \mathbf{x} + \sum_{\omega \in \Omega} p_\omega z_\omega \\
 \text{s.t.} \quad & \mathbf{x}^\top \mathbf{n} \leq \mathbf{L} \\
 & (\alpha^k)^\top (\mathbf{d}_\omega - \mathbf{x} \mathbf{1}) \geq z_\omega, \alpha^k \in \mathcal{O}, \forall \omega \\
 & \mathbf{x} \in \mathbb{N}^{M \times N, +}
 \end{aligned} \tag{7}$$

Constraints will be generated from problem (6) until an optimal solution is found.



# Obtain Seat Planning Composed of Full or Largest Patterns

Challenge: it is hard to solve SSP in some cases.

Property: there exists an optimal solution to SSP such that the patterns associated with this **optimal solution** are composed of the **full or largest patterns** under any given scenarios.

We aim to obtain a seat planning composed of full or largest patterns by the optimal solution to the LP relaxation of SSP.

- Obtain the solution to the relaxed SSP,  $\mathbf{x}^*$ , by benders decomposition. Aggregate  $\mathbf{x}^*$  to the number of each group type,  $s_i = \sum_j x_{ij}^*, i \in \mathbf{M}$ .
- Obtain the optimal solution,  $\mathbf{x}^1$ , by solving problem (1) with  $d_i = s_i$ .
- Obtain full or largest patterns by solving problem (2) with  $\mathbf{H} = \mathbf{x}^1$ .

# Seat Assignment with Dynamic Demand

# Real-time Seat Assignment

- There is one and only one group arrival at each period,  $t = 1, \dots, T + 1$ .
- The probability of an arrival of group type  $i$ :  $p_i$ .
- $\mathbf{L}^t = (l_1, l_2, \dots, l_N)$ , where  $l_j = 0, \dots, L_j, j \in \mathcal{N}$ : Remaining capacity.
- $u_{i,j}^t$ : Decision. Assign group type  $i$  to row  $j$  at period  $t$ ,  $u_{i,j}^t = 1$ .
- $U^t(\mathbf{L}^t) = \{u_{i,j}^t \in \{0, 1\}, \forall i, j | \sum_{j=1}^N u_{i,j}^t \leq 1, \forall i, n_i u_{i,j}^t \mathbf{e}_j \leq \mathbf{L}^t, \forall i, j\}$ .
- $\mathbf{e}_j$ : Unit column vector with  $j$ -th element being 1.
- $V^t(\mathbf{L}^t)$ : Value function at period  $t$ , given remaining capacity,  $\mathbf{L}^t$ .

$$V^t(\mathbf{L}^t) = \max_{u_{i,j}^t \in U^t(\mathbf{L}^t)} \left\{ \sum_{i=1}^M p_i \left( \sum_{j=1}^N i u_{i,j}^t + V^{t+1}(\mathbf{L}^t - \sum_{j=1}^N n_i u_{i,j}^t \mathbf{e}_j) \right) + p_0 V^{t+1}(\mathbf{L}^t) \right\}$$

# Seat Assignment under The Seat Planning

- Assign the seats under the **flexible seat planning**

Situation 1: **real-time seat assignment**

Situation 2: accept or reject each arriving group, then assign the seats after all groups arrive.

- Assign the seats under the **fixed seat planning**

The seats will be dismantled before people arrive to prevent them from occupying those seats.

# Proposed Methods to Real-time Seat Assignment

- Suppose the supply associated with the seat planning is  $[X_1, \dots, X_M]$ . ( $X_i = \sum_j x_{ij}, \forall i$ )

For the arriving group type  $i$ ,

- if  $X_i > 0$ , accept the group, assign it by the tie-breaking rule (full pattern first);
- if  $X_i = 0$ , two methods:
  1. Based on the modified SSP.
  2. Based on the seat planning from the LP relaxation of SSP.
    - Determine the possible group type  $\hat{i}^* > i$  by group-type control
    - Decision on assigning the group to a specific row

# Method 1: Modified SSP

Introduce the decision variables  $I_j, j \in \mathcal{N}$  indicating whether we accept the arriving group type  $i'$  in row  $j$ .

$$\begin{aligned}
 \max \quad & \sum_j i' I_j + E_\omega \left[ \sum_{i=1}^{M-1} (n_i - \delta) \left( \sum_{j=1}^N x_{ij} + y_{i+1,\omega}^+ - y_{i\omega}^+ \right) + (n_M - \delta) \left( \sum_{j=1}^N x_{Mj} - y_{M\omega}^+ \right) \right] \\
 \text{s.t.} \quad & \sum_{j=1}^N x_{ij} - y_{i\omega}^+ + y_{i+1,\omega}^+ + y_{i\omega}^- = d_{i\omega}, \quad i = 1, \dots, M-1, \omega \in \Omega \\
 & \sum_{j=1}^N x_{ij} - y_{i\omega}^+ + y_{i\omega}^- = d_{i\omega}, \quad i = M, \omega \in \Omega \\
 & \sum_{i=1}^M n_i x_{ij} \leq L_j - \textcolor{red}{n_{i'} I_j}, \quad j \in \mathcal{N} \\
 & \sum_{j=1}^N I_j \leq 1 \\
 & x_{ij} \in \mathbb{N}, \quad i \in \mathcal{M}, j \in \mathcal{N}, y_{i\omega}^+, y_{i\omega}^- \in \mathbb{N}, \quad i \in \mathcal{M}, \omega \in \Omega, I_j \in \{0, 1\}, j \in \mathcal{N}.
 \end{aligned} \tag{8}$$

Intuitive but not efficient.

## Method 2: Dynamic Seat Assignment (DSA)

For the arriving group type  $i$ , if  $X_i = 0$ :

Step 1. Determine the group type  $\hat{i}^* > i$  by the group-type control.

Step 2. Decision on assigning the group type  $i$  to a specific row.

## Method 2: Dynamic Seat Assignment (DSA)

1. **Group-type control:** determine the group type  $\hat{i}^*$  to place arriving group type  $i$ .

Recall that the supply is  $[X_1, \dots, X_M]$ .

$$d^t(i, \hat{i}) = \underbrace{i + (\hat{i} - i - \delta)P(D_{\hat{i}-i-\delta}^{T-t} \geq X_{\hat{i}-i-\delta} + 1)}_{\text{acceptance}} - \underbrace{\hat{i}P(D_{\hat{i}}^{T-t} \geq X_{\hat{i}})}_{\text{rejection}}$$

$P(D_i^{T-t} \geq X_i)$  is the probability that the demand of group type  $i$  in  $(T-t)$  periods is no less than  $X_i$ .

$D_j^t$  is a random variable indicating the number of group type  $j$  in  $t$  periods.

For all  $\hat{i} > i$ , find the maximum value denoted as  $d^t(i, \hat{i}^*)$ .

If  $d^t(i, \hat{i}^*) \geq 0$ , we place the group of  $i$  in  $(\hat{i}^* + \delta)$ -size seats. Otherwise, reject the group.

## Method 2: Dynamic Seat Assignment (DSA)

### 2. Decision on assigning the group to a specific row.

- Determine the specific row  $\hat{j}$  including group type  $i^*$  by tie-breaking rule (Non-full pattern first).
- Whether to assign the group in row  $\hat{j}$ 
  - Value of Acceptance (VoA): value of LP relaxation of SSP with  $(\mathbf{L} - n_i \mathbf{e}_{\hat{j}})$  plus  $i$ .
  - Value of Rejection (VoR): value of LP relaxation of SSP with  $\mathbf{L}$ .
  - If VoA is no less than VoR, accept group type  $i$ ; otherwise, reject it.

# Compared with Other Policies

We compared DSA with the following policies

- Bid-price control
- Dynamic programming based heuristic
- Booking limit control
- First come first served
- Benchmark:  
Offline optimal solution with knowing demands before decision

# Bid-price Control

The dual problem of LP relaxation of problem (1) is:

$$\begin{aligned}
 \min \quad & \sum_{i=1}^M d_i z_i + \sum_{j=1}^N L_j \beta_j \\
 \text{s.t.} \quad & z_i + \beta_j n_i \geq (n_i - \delta), \quad i \in \mathcal{M}, j \in \mathcal{N} \\
 & z_i \geq 0, i \in \mathcal{M}, \beta_j \geq 0, j \in \mathcal{N}.
 \end{aligned} \tag{9}$$

The optimal solution to problem (9) is given by  $z_1, \dots, z_{\tilde{i}} = 0, z_i = \frac{\delta(n_i - n_{\tilde{i}})}{n_{\tilde{i}}}$ , for  $i = \tilde{i} + 1, \dots, M$ ,  $\beta_j = \frac{n_{\tilde{i}} - \delta}{n_{\tilde{i}}}$  for all  $j$ .

For the group type  $i$ , if  $i - \beta_j n_i \geq 0$ , accept it.

# Dynamic Programming Based Heuristic

- Relax all rows to one row with the same capacity by  $L = \sum_{j=1}^N L_j$ .
  - Deterministic problem:  
 $\{\max \sum_{i=1}^M (n_i - \delta)x_i : x_i \leq d_i, i \in \mathcal{M}, \sum_{i=1}^M n_i x_i \leq L, x_i \in \mathbb{Z}_0^+\}$ .
- Decision:  $u^t$ . If we accept a request in period  $t$ ,  $u^t = 1$ ; otherwise,  $u^t = 0$ .
  - DP with one row can be expressed as:

$$V^t(l^t) = \max_{u^t \in \{0,1\}} \left\{ \sum_i p_i [V^{t+1}(l^t - n_i u^t) + i u^t] + p_0 V^{t+1}(l^t) \right\}$$

$$V^{T+1}(l) = 0, \forall l.$$

- After accepting one group, assign it in some row arbitrarily when the capacity of the row allows.

# Booking limit Control

Basic idea: for each type of requests, we only allocate a fixed amount according to the static solution and reject all other exceeding requests.

- 1 Observe the arrival group type  $i$ .
- 2 Solve problem (1) with  $d_i = p_i \times T, i \in \mathcal{M}$ .
- 3 Obtain the optimal solution,  $x_{ij}^*$  and the aggregate optimal solution,  $\mathbf{X}$ .
- 4 If  $X_i > 0$ , accept the arrival and assign the group to row  $k$  where  $x_{ik} > 0$ , update  $\mathbf{L}^{t+1} = \mathbf{L}^t - n_i \mathbf{e}_k$ ; otherwise, reject it, let  $\mathbf{L}^{t+1} = \mathbf{L}^t$ .

## Parameters Description

$$M = 4, \delta = 1, |\Omega| = 1000, N = 10, L_j = 21, j \in \mathcal{N}.$$

Consider three sets of probability distributions with the same expectation of demand each period:

D1: [0.25, 0.25, 0.25, 0.25]

D2: [0.25, 0.35, 0.05, 0.35]

D3: [0.15, 0.25, 0.55, 0.05]

Each entry in the result columns is the average of 100 instances.

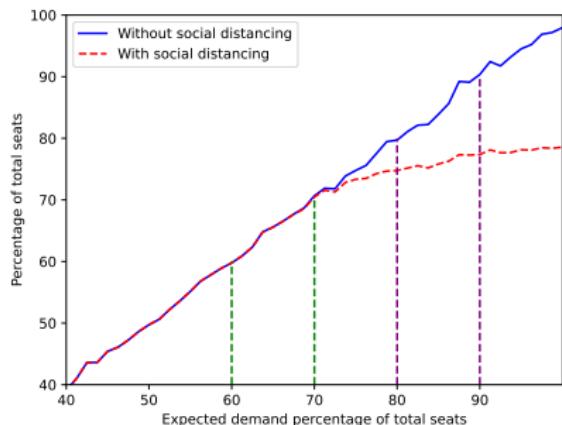
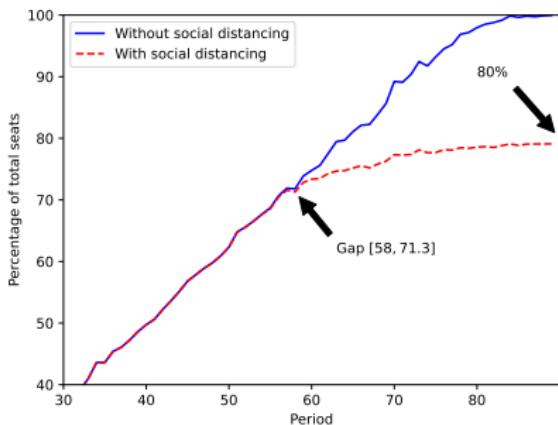
# Performances of Different Policies

Distribution	T	DSA (%)	DP1 (%)	Bid (%)	Booking (%)	FCFS (%)
D1	60	99.12	98.42	98.38	96.74	98.17
	70	98.34	96.87	96.24	97.18	94.75
	80	98.61	95.69	96.02	98.00	93.18
	90	99.10	96.05	96.41	98.31	92.48
	100	99.58	95.09	96.88	98.70	92.54
D2	60	98.94	98.26	98.25	96.74	98.62
	70	98.05	96.62	96.06	96.90	93.96
	80	98.37	96.01	95.89	97.75	92.88
	90	99.01	96.77	96.62	98.42	92.46
	100	99.23	97.04	97.14	98.67	92.00
D3	60	99.14	98.72	98.74	96.61	98.07
	70	99.30	96.38	96.90	97.88	96.25
	80	99.59	97.75	97.87	98.55	95.81
	90	99.53	98.45	98.69	98.81	95.50
	100	99.47	98.62	98.94	98.90	95.25

DSA has better performance than other policies under different demands.

# Impact of Social Distancing As Demand Increases

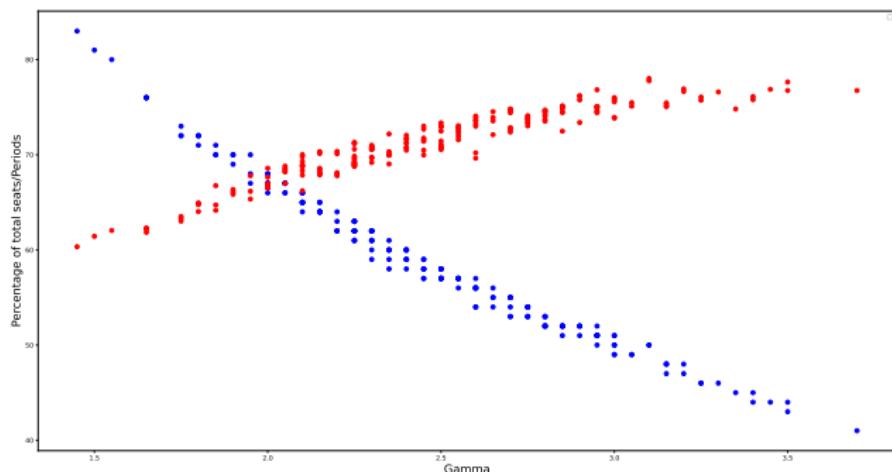
When the probability distribution is  $[0.25, 0.25, 0.25, 0.25]$



- Gap point: the first period when there is difference
- Demand **less than** 71.3% total seats: no difference
- Demand **larger than** 71.3% total seats: difference becomes larger
- The largest occupancy rate is 80% by the property of the largest pattern.

# Gap Points and Occupancy Rates under Different Probability Distributions

$\gamma = p_1 * 1 + p_2 * 2 + p_3 * 3 + p_4 * 4$ : the expected number of people at each period.



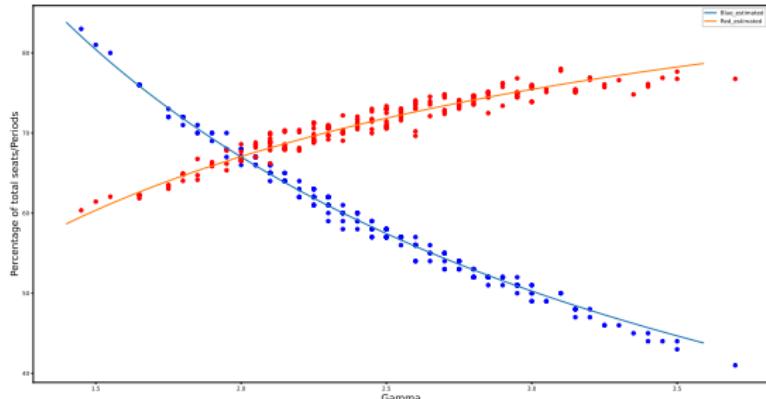
Blue points: period of the gap point.

Red points: occupancy rate of the gap point.

Difference is small under the same gamma.

Figure: Gap points and occupancy rates with 200 probability distributions

# Estimation of Gap Points



Ideal: the groups before gap point will occupy all seats.  $c_1, c_2$  can be seen as the discount factor compared to the ideal situation.

Gap points can be estimated with  $\gamma$ .

$$y_1 = \frac{c_1 \tilde{L}}{\gamma + \delta}, \quad y_2 = c_2 \frac{\gamma}{\gamma + \delta} \frac{\tilde{L}}{\tilde{L} - N\delta}$$

The larger  $c_1, c_2$ , the closer to the ideal situation.

Seat layout(# of rows × # of seats)	Fitting Values of $c_1$	Fitting Values of $c_2$
10 × 11	$0.909 \pm 0.013$	$89.89 \pm 1.436$
10 × 16	$0.948 \pm 0.008$	$94.69 \pm 0.802$
10 × 21	$0.955 \pm 0.004$	$95.44 \pm 0.571$
10 × 26	$0.966 \pm 0.004$	$96.23 \pm 0.386$
10 × 31	$0.965 \pm 0.003$	$96.67 \pm 0.434$
10 × 36	$0.968 \pm 0.003$	$97.04 \pm 0.289$

The larger the size of row is, the larger the values of  $c_1, c_2$  are.

# Seat Assignment After All Groups Arrive

DP1-A: Dynamic programming based heuristic policy after all groups arrive

Bid-A: Bid-price control policy after all groups arrive

Distribution	T	DP1-A (%)	Bid-A (%)	DP1 (%)	Bid (%)
D1	60	99.52	99.44	98.42	98.38
	70	99.32	98.97	96.87	96.24
	80	99.34	99.30	95.69	96.02
	90	99.55	99.49	96.05	96.41
	100	99.78	99.66	95.09	96.88
D2	60	99.50	99.37	98.26	98.25
	70	99.40	98.97	96.62	96.06
	80	99.46	99.24	96.01	95.89
	90	99.59	99.35	96.77	96.62
	100	99.77	99.61	97.04	97.14
D3	60	99.57	99.54	98.72	98.74
	70	99.46	99.39	96.38	96.90
	80	99.50	99.30	97.75	97.87
	90	99.34	99.44	98.45	98.69
	100	99.34	99.55	98.62	98.94

The performance is greatly improved without the assign-to-seat constraint.

# Seat Assignment under Fixed Seat Planning

The assignment is based on the fixed seat planning and the group-type control.

Distribution	T	# of rows	Compared to the optimal (%)
D1	70	10	94.97
	80		96.48
	90		97.94
	100		98.91
D2	70	10	95.90
	80		97.06
	90		98.58
	100		99.47
D3	70	10	97.41
	80		98.85
	90		98.73
	100		98.46
D1	140	20	95.83
	160		97.46
	180		99.05
	200		99.74

The performance declines with the fixed seat planning constraint.

# Conclusion and Future Work

# Conclusion and Future Work

## Conclusion

- Address the problem of seat planning and assignment with social distancing.
- Provide a comprehensive solution for optimizing seat assignments under dynamic situation.
- Provide different methods in different scenarios and elaborate on the role of social distancing.

## Future work

- Consider more flexible scenarios, people can choose the seats and leave randomly.
- Consider a scattered seat assignment when there are sufficient seats.

# Thank You!