# Dynamic Seat Assignment with Social Distancing

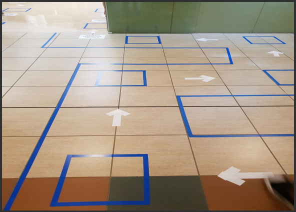
IEDA
The Hong Kong University of Science and Technology

# Table of Contents

# Introduction

# Social Distancing under Pandemic

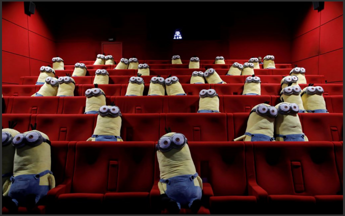
- Social distancing measures.

# Social Distancing under Pandemic

- Social distancing in seating areas.



# Literature Review

# Seat Planning with Social Distancing

- Seat planning on airplanes, classrooms, trains.
  Allocation of seats on airplanes (Ghorbani et.al 2020), classroom layout planning (Bortolete et al. 2022), seat planning in long-distancing trains (Haque & Hamid 2022).

- Group seat assignment in amphitheaters, airplanes, theater.

- Social distancing can be enforced in different groups (Moore et al. 2021).

- Seating planning for known groups in amphitheaters (Haque & Hamid 2022), airplanes (Salari et al. 2022), theater (Blom et al. 2022).

# Dynamic Seat Assignment

- Related to multiple knapsack problem [4] and dynamic knapsack problem [3].
- Dynamic seat assignment on airplane [2], train [1, 5].
- Assign-to-seat: dynamic capacity control for selling high-speed train tickets. [5]

# Problem Definition

# Seat Planning with Social Distancing

- Group type $\mathcal{M} = \{1, \ldots, M\}$.
- Row $\mathcal{N} = \{1, \ldots, N\}$.
- The social distancing: $\delta$ seat(s).
- $n_i = i + \delta$: the new size of group type $i$ for each $i \in \mathcal{M}$.
- The number of seats in row $j$: $S_j, j \in \mathcal{N}$.
- $L_j = S_j + \delta$: the length of row $j$ for each $j \in \mathcal{N}$.

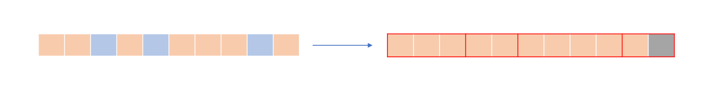


Figure: Problem Conversion with One Seat as Social Distancing

## Basic Concepts

- $P_k = (t_1, \ldots, t_M)$, pattern, the seat planning for each group type $i, i \in \mathcal{M}$ in one row.
- For each pattern $k$, $\alpha_k, \beta_k$ indicate the number of groups and the left seats, respectively.
- Loss for pattern $k$: $\alpha_k \delta + \beta_k - \delta$.
- The largest patterns: patterns with the minimal loss.
- Full patterns: $\beta_k = 0$.

- Example:
  $\delta = 1$, $M = 4$, $n_i = i + 1, i \in \mathcal{M}$, $L = 21$.
  Largest patterns: $(0, 0, 0, 4), (0, 0, 4, 1), (0, 2, 0, 3)$.
  Not a Full pattern: $(0, 0, 0, 4)$.

# Loss of The Largest Patterns

- Find a largest pattern: consider $L = n_M \cdot q + r, 0 \leq r < n_M$, where $q$ is the qutionet and $r$ is remainder. If the remainder $r$ is greater than $\delta$, the seats can be occupied by a group of size $(r - \delta)$. However, if $r$ is less than or equal to $\delta$, the seats should be left empty.
- Example: $\delta = 1$, $M = 4$, $L = 21$, $n_M = 5$, $q = 4$, $(0, 0, 0, 4)$ is a largest pattern.

* Loss of the largest patterns: $l(L) = \lfloor \frac{L}{n_M} \rfloor \delta - \delta + f((L \mod n_M))$, where $f(r) = 0$ if $r > \delta$; $f(r) = r$ if $r \leq \delta$.
* For a original seat layout, $\{S_1, S_2, \ldots, S_N\}$, the minimal total loss: $\sum_j l(S_j + \delta)$.

# Dynamic Seat Assignment Problem

- There is one and only one group arrival at each period, $t = 1, \dots, T+1$.
- The probability of an arrival of group type $i$: $p_i$.
- $\mathbf{L} = (l_1, l_2, \dots, l_N)$, where $l_j = 0, \dots, L_j, j \in \mathcal{N}$: Remaining capacity.
- $u_{i,j}$: Decision. Assign group type $i$ to row $j$, $u_{i,j} = 1$.
- $V_t(\mathbf{L})$: Value function with $t$ periods to go.

$$V_t(\mathbf{L}) = \max_{u \in U(\mathbf{L})} \{\sum_{i=1}^{M} p_i (\sum_{j=1}^{N} i u_{i,j} + V_{t+1}(\mathbf{L} - \sum_{j=1}^{N} n_i u_{i,j} \mathbf{e}_j^\top))\}$$

- DP is computationally complex caused by the curse of dimensionality.
- We give a seat planning by stochastic programming firstly, then apply stochastic planning policy to make the decision.

# Seat Planning by Stochastic Programming

# Method Flow

- The formulation of scenario-based stochastic programming(SSP).
- Reformulate (SSP) to the benders master problem(BMP) and subproblem.
- The optimal solution can be obtained by solving (BMP) iteratively.
- To avoid solving IP directly, we consider the linear relaxation form.
- Obtain integral seat planning by deterministic model.

# Scenario-based Stochastic Programming

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

# Reformulation

$$
\begin{aligned}
\max \quad & \mathbf{c}^\intercal \mathbf{x} + z(\mathbf{x}) \\
\text{s.t.} \quad & \mathbf{n}\mathbf{x} \leq \mathbf{L} \\
& \mathbf{x} \in \mathbb{Z}_+^{M \times N},
\end{aligned} \tag{2}
$$

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 $\omega \in \Omega$,

$$
\begin{aligned}
z_\omega(\mathbf{x}) := \max \quad & \mathbf{f}^\intercal \mathbf{y} \\
\text{s.t.} \quad & \mathbf{x}\mathbf{1} + \mathbf{V}\mathbf{y} = \mathbf{d}_\omega \\
& \mathbf{y} \geq 0.
\end{aligned} \tag{3}
$$

# Solution to Subproblem

Problem (3) is easy to solve with a given $\mathbf{x}$ which can be seen by the dual problem:

$$\begin{aligned} \min \quad & \alpha_\omega^\intercal (\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \\ \text{s.t.} \quad & \alpha_\omega^\intercal \mathbf{V} \geq \mathbf{f}^\intercal \end{aligned} \tag{4}$$

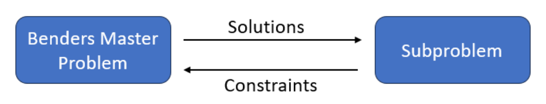
- The feasible region of problem (4), $P = \{\alpha | \alpha^\intercal V \geq \mathbf{f}^\intercal\}$, is bounded. In addition, all the extreme points of $P$ are integral.
- The optimal solution to this problem can be obtained directly according to the complementary slackness property.

# Benders Decomposition Procedure

Let $z_\omega$ be the lower bound of problem (4), (SSP) can be obtained by solving following restricted benders master problem(BMP):

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} p_\omega z_\omega \\ \text{s.t.} \quad & \mathbf{n}\mathbf{x} \leq \mathbf{L} \\ & (\alpha^k)^\intercal (\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega, \alpha^k \in \mathcal{O}, \forall \omega \\ & \mathbf{x} \in \mathbb{Z}_+ \end{aligned} \tag{5}$$

Constraints will be generated from problem (4) until an optimal solution is found.




To avoid solving IP directly, we consider the linear relaxation of Problem (5).

# Deterministic Formulations

To obtain an integral seat planning, we consider the following two deterministic formulations.

$$\begin{aligned} \max \quad & \sum_{i=1}^{M}\sum_{j=1}^{N}(n_i - s)x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{N} x_{ij} \leq s_i^0, \quad i \in \mathcal{M}, \\ & \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} \\ & x_{ij} \in \mathbb{Z}_+, \quad i \in \mathcal{M}, j \in \mathcal{N}. \end{aligned} \tag{6}$$

$$\begin{aligned} \max \quad & \sum_{i=1}^{M}\sum_{j=1}^{N}(n_i - s)x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{N} x_{ij} \geq s_i^1, \quad i \in \mathcal{M}, \\ & \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} \\ & x_{ij} \in \mathbb{Z}_+, \quad i \in \mathcal{M}, j \in \mathcal{N}. \end{aligned} \tag{7}$$

Problem (6) can generate a feasible seat planning.
Problem (7) can generate a seat planning no inferior than given feasible seat planning.

# Obtain The Feasible Seat Planning

Step 1. Obtain the solution, $\mathbf{x}^*$, by benders decomposition. Aggregate $\mathbf{x}^*$ to the number of each group type, $s_i^0 = \sum_j x_{ij}^*, i \in \mathbf{M}$.

Step 2. Solve problem (6) to obtain the optimal solution, $\mathbf{x}^1$. Aggregate $\mathbf{x}^1$ to the number of each group type, $s_i^1 = \sum_j x_{ij}^1, i \in \mathbf{M}$.

Step 3. Solve problem (7) to obtain the optimal solution, $\mathbf{x}^2$. Aggregate $\mathbf{x}^2$ to the number of each group type, $s_i^2 = \sum_j x_{ij}^2, i \in \mathbf{M}$.

Step 4. For each row, construct a full or largest pattern.

# Dynamic Seat Assignment for Each Group Arrival

# Policies

- Stochastic planning policy
- Bid-price control
- Dynamic programming based heuristic
- Booking limit control
- First come first served

# Stochastic Planning Policy(SPP)

- Group-type control
  - Feasible Seat planning from stochastic programming.
  - When there is no small group, decide which group type to be assigned. Let $d(i, j)$ be the difference of expected number of accepted people between acceptance and rejection on group $i$ occupying $(j + \delta)$-size seats.
    Find the largest $d(i, j)$, denoted as $d(i, j^*)$. If $d(i, j^*) > 0$, we will place the group of $i$ in $(j^* + \delta)$-size seats. Otherwise, reject the group.

# Stochastic Planning Policy(SPP)

- Value of Acceptance and Rejection
  - Compare the value of stochastic programming when assigning in the row versus not assigning.

  Value of Acceptance: use the value of stochastic programming as the approximation of $V_t(\mathbf{L} - \mathbf{n_i}\mathbf{e_j}^\top) + i$.
  Value of Rejection: approximation of $V_t(\mathbf{L})$.

# Bid-price Control

The dual problem of linear relaxation of problem (6) is:

$$\begin{aligned} \min \quad & \sum_{i=1}^{M} d_i z_i + \sum_{j=1}^{N} L_j \beta_j \\ \text{s.t.} \quad & z_i + \beta_j n_i \geq (n_i - \delta), \quad i \in \mathcal{M}, j \in \mathcal{N} \\ & z_i \geq 0, i \in \mathcal{M}, \beta_j \geq 0, j \in \mathcal{N}. \end{aligned} \tag{8}$$

There exists $h$ such that the aggregate optimal solution to relaxation of problem (6) takes the form $xe_h + \sum_{i=h+1}^{M} d_i e_i$, $x = (L - \sum_{i=h+1}^{M} d_i n_i)/n_h$.

# Dynamic Programming Based Heuristic

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

$$V_t(L) = \mathbb{E}_{i \sim p}[\max_{u \in \{0,1\}} \{[V_{t+1}(L - n_i u) + iu]\}], L \geq 0$$

$$V_{T+1}(x) = 0, \forall x.$$

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

# Booking limit Control

Solve problem (6) using the expected demand. Then for every type of requests, we only allocate a fixed amount according to the static solution and reject all other exceeding requests.

When we solve the linear relaxation of problem (6), the aggregate optimal solution is the limits for each group type. Interestingly, the bid-price control policy is found to be equivalent to the booking limit control policy.

# Numerical Results

# Running time of Benders Decomposition and IP

| # of scenarios | demands | running time of IP(s) | Benders (s) | # of rows | # of groups | # of seats |
|---|---|---|---|---|---|---|
| 1000 | (150, 350) | 5.1 | 0.13 | 30 | 8 | (21, 50) |
| 5000 | | 28.73 | 0.47 | 30 | 8 | |
| 10000 | | 66.81 | 0.91 | 30 | 8 | |
| 50000 | | 925.17 | 4.3 | 30 | 8 | |
| 1000 | (1000, 2000) | 5.88 | 0.29 | 200 | 8 | (21, 50) |
| 5000 | | 30.0 | 0.62 | 200 | 8 | |
| 10000 | | 64.41 | 1.09 | 200 | 8 | |
| 50000 | | 365.57 | 4.56 | 200 | 8 | |
| 1000 | (150, 250) | 17.15 | 0.18 | 30 | 16 | (41, 60) |
| 5000 | | 105.2 | 0.67 | 30 | 16 | |
| 10000 | | 260.88 | 1.28 | 30 | 16 | |
| 50000 | | 3873.16 | 6.18 | 30 | 16 | |

# Feasible Seat Planning versus IP Solution

| # samples | T | probabilities | # rows | people served by FSP | IP |
|---|---|---|---|---|---|
| 1000 | 45 | [0.4,0.4,0.1,0.1] | 8 | 85.30 | 85.3 |
| 1000 | 50 | [0.4,0.4,0.1,0.1] | 8 | 97.32 | 97.32 |
| 1000 | 55 | [0.4,0.4,0.1,0.1] | 8 | 102.40 | 102.40 |
| 1000 | 60 | [0.4,0.4,0.1,0.1] | 8 | 106.70 | NA |
| 1000 | 65 | [0.4,0.4,0.1,0.1] | 8 | 108.84 | 108.84 |
| 1000 | 35 | [0.25,0.25,0.25,0.25] | 8 | 87.16 | 87.08 |
| 1000 | 40 | [0.25,0.25,0.25,0.25] | 8 | 101.32 | 101.24 |
| 1000 | 45 | [0.25,0.25,0.25,0.25] | 8 | 110.62 | 110.52 |
| 1000 | 50 | [0.25,0.25,0.25,0.25] | 8 | 115.46 | NA |
| 1000 | 55 | [0.25,0.25,0.25,0.25] | 8 | 117.06 | 117.26 |
| 5000 | 300 | [0.25,0.25,0.25,0.25] | 30 | 749.76 | 749.76 |
| 5000 | 350 | [0.25,0.25,0.25,0.25] | 30 | 866.02 | 866.42 |
| 5000 | 400 | [0.25,0.25,0.25,0.25] | 30 | 889.02 | 889.44 |
| 5000 | 450 | [0.25,0.25,0.25,0.25] | 30 | 916.16 | 916.66 |

Each entry of people served is the average of 50 instances. IP will spend more than 2 hours in some instances, as 'NA' showed in the table.

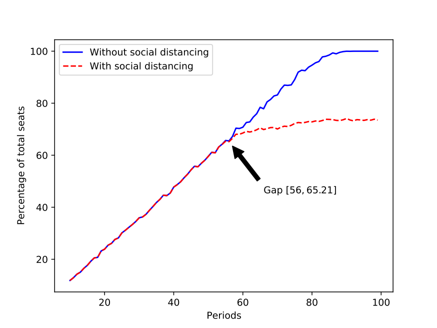
# Performances of Different Policies

| T | Probabilities | SPP(%) | DP1(%) | Bid(%) | Booking(%) | FCFS(%) |
|---|---|---|---|---|---|---|
| 60 | [0.25, 0.25, 0.25, 0.25] | 99.12 | 98.42 | 98.38 | 96.74 | 98.17 |
| 70 | [0.25, 0.25, 0.25, 0.25] | 98.34 | 96.87 | 96.24 | 97.18 | 94.75 |
| 80 | [0.25, 0.25, 0.25, 0.25] | 98.61 | 95.69 | 96.02 | 98.00 | 93.18 |
| 90 | [0.25, 0.25, 0.25, 0.25] | 99.10 | 96.05 | 96.41 | 98.31 | 92.48 |
| 100 | [0.25, 0.25, 0.25, 0.25] | 99.58 | 95.09 | 96.88 | 98.70 | 92.54 |
| 60 | [0.25, 0.35, 0.05, 0.35] | 98.94 | 98.26 | 98.25 | 96.74 | 98.62 |
| 70 | [0.25, 0.35, 0.05, 0.35] | 98.05 | 96.62 | 96.06 | 96.90 | 93.96 |
| 80 | [0.25, 0.35, 0.05, 0.35] | 98.37 | 96.01 | 95.89 | 97.75 | 92.88 |
| 90 | [0.25, 0.35, 0.05, 0.35] | 99.01 | 96.77 | 96.62 | 98.42 | 92.46 |
| 100 | [0.25, 0.35, 0.05, 0.35] | 99.23 | 97.04 | 97.14 | 98.67 | 92.00 |
| 60 | [0.15, 0.25, 0.55, 0.05] | 99.14 | 98.72 | 98.74 | 96.61 | 98.07 |
| 70 | [0.15, 0.25, 0.55, 0.05] | 99.30 | 96.38 | 96.90 | 97.88 | 96.25 |
| 80 | [0.15, 0.25, 0.55, 0.05] | 99.59 | 97.75 | 97.87 | 98.55 | 95.81 |
| 90 | [0.15, 0.25, 0.55, 0.05] | 99.53 | 98.45 | 98.69 | 98.81 | 95.50 |
| 100 | [0.15, 0.25, 0.55, 0.05] | 99.47 | 98.62 | 98.94 | 98.90 | 95.25 |

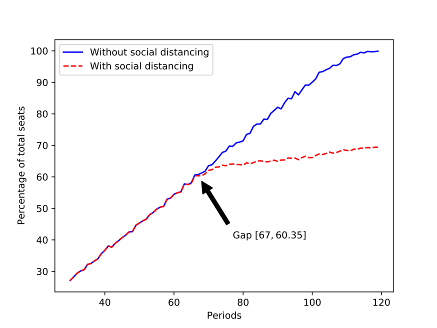
SPP has better performance than other policies under different demands.

# Impact of Social Distance as Demand Increases

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




(a) When $\gamma = 2.5$




(b) When $\gamma = 1.9$

The gap point represents the first period where the number of people without social distancing is larger than that with social distancing and the gap percentage is the corresponding percentage of total seats.

# When Supply and Demand Are Close

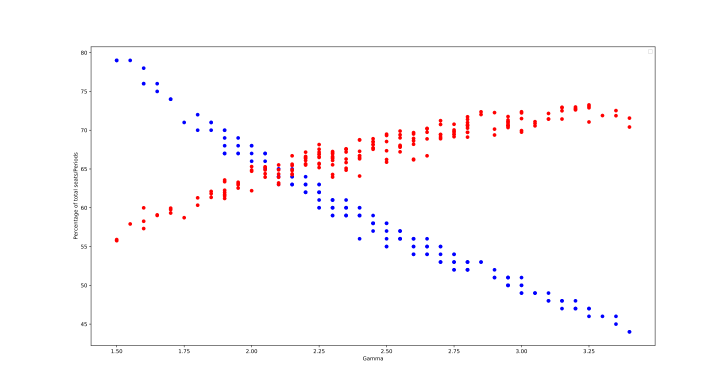



Figure: Gap points under 200 probabilities

Blue points: period of the gap point. Red points: occupancy rate of the gap point. Gap points can be estimated.

# Conclusion

# Conclusion

- Our approach, stochastic planning policy, provides a comprehensive solution for optimizing seat assignments while ensuring the safety of customers under dynamic situation.
-

# References


[1] Matthew E Berge and Craig A Hopperstad. "Demand driven dispatch: A method for dynamic aircraft capacity assignment, models and algorithms". In: *Operations research* 41.1 (1993), pp. 153–168.

[2] Younes Hamdouch et al. "Schedule-based transit assignment model with vehicle capacity and seat availability". In: *Transportation Research Part B: Methodological* 45.10 (2011), pp. 1805–1830.

[3] Anton J Kleywegt and Jason D Papastavrou. "The dynamic and stochastic knapsack problem". In: *Operations research* 46.1 (1998), pp. 17–35.

[4] David Pisinger. "An exact algorithm for large multiple knapsack problems". In: *European Journal of Operational Research* 114.3 (1999), pp. 528–541.

[5] Feng Zhu et al. "Assign-to-seat: Dynamic capacity control for selling high-speed train tickets". In: *Manufacturing & Service Operations Management* (2023).

# The End