Dynamic Seat Assignment

With Social Distancing

IEDA
The Hong Kong University of Science and Technology

Table of contents

Literature Review

Seat Planning with Social Distancing

- Seat planning on airplanes, classrooms, trains.
- Group seat assignment in amphitheaters, airplanes, theater.

Dynamic Seat Assignment

- Multiple knapsack problem
- Revenue management + Assign-to-seat

Problem Definition

Seat Planning with Social Distancing

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

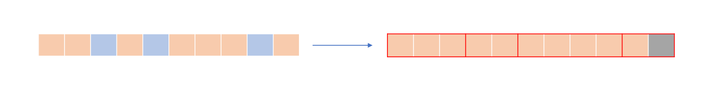


Figure: Problem Conversion with One Seat as Social Distancing

Some Definitions

- Pattern refers to the seat planning for one row.
- For each pattern k, α_k, β_k indicate the number of groups and the left seats, respectively.
- Denote by $\alpha_k\delta + \beta_k - \delta$ the loss for pattern k, $l(k)$. The loss represents the number of people lost compared to the situation without social distancing.
- Let I_1 be the set of patterns with the minimal loss. We call the patterns from I_1 are the largest. The patterns with zero left seat are called full patterns.
- Suppose there are n groups in a row, for ease of brevity, we use a descending form $P_k = (t_1, t_2, \ldots, t_n)$ to denote pattern k, where t_h is the new group size, $h = 1, \ldots, n$.

Example

- Suppose the social distancing is one seat and there are four types of groups. Then the new sizes of groups are $2, 3, 4, 5$, respectively.
- The length of one row is $L = 21$.
- Then these patterns, $(5,5,5,5), (5,4,4,4,4), (5,5,5,3,3)$, belong to I_1.
- Pattern $(5,5,5,5)$ is not full because there is one left seat.

Property

- Select the maximal group size, n_M, as many as possible and the left space is occupied by the group with the corresponding size.
- Let $L = n_M \cdot q + r, 0 \leq r < n_M$. The loss of the largest pattern is $q\delta - \delta + f(r)$, where $f(r) = 0$ if $r > \delta$; $f(r) = r$ if $r \leq \delta$.
- For a seat layout, $\{S_1, S_2, \ldots, S_N\}$, the minimal total loss is $\sum_j (\lfloor \frac{S_j+\delta}{n_M} \rfloor - \delta + f((S_j + \delta) \mod n_M))$. The maximal number of people assigned is $\sum_j (S_j - \lfloor \frac{S_j+\delta}{n_M} \rfloor + \delta - f((S_j + \delta) \mod n_M))$.

Dynamic Programming

Dynamic seat assignment can be characterized by DP:

$$V_t(\mathbf{L}) = \mathbb{E}_{i\sim p}\left[\max_{\substack{j\in\mathcal{N}:\\ L_j \geqslant n_i}} \left\{V_{t+1}\left(\mathbf{L} - n_i \mathbf{e}_j^{\intercal}\right) + i, V_{t+1}(\mathbf{L})\right\}\right]$$

$$V_{T+1}(\mathbf{L}) = 0,$$

- $\mathbf{L} = (L_1^r, L_2^r, \ldots, L_N^r)$, remaining capacity. L_j^r represents the number of remaining seats in row j.
- p_i: the probability of an arrival of group type i.

Scenario-based Stochastic Programming

Scenario-based Stochastic Programming

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

For any i, ω, at most one of $y^+_{i\omega}$ and $y^-_{i\omega}$ can be positive.

Two-stage

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

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

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

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

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

Solve the Second Stage Problem

The dual of problem (3) is

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

Let $P = \{\alpha | \alpha^\intercal V \geq \mathbf{f}^\intercal\}$. The feasible region of problem (4), P, is bounded. In addition, all the extreme points of P are integral.

Delayed Constraint Generation

LP of problem (1) can be obtained by solving following restricted benders master problem(RBMP):

$$
\begin{aligned}
\max \quad & \mathbf{c}^\intercal x + \sum_{\omega \in \Omega} p_\omega z_\omega \\
\text{s.t.} \quad & \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} \\
& (\alpha^k)^\intercal (\mathbf{d}_\omega - \mathbf{x1}) \geq z_\omega, \alpha^k \in \mathcal{O}^t, \forall \omega \\
& \mathbf{x} \geq 0
\end{aligned} \tag{5}
$$

Constraints will be generated from problem (4) until the value of RBMP converges.

Benders Decomposition Algorithm

Step 1. Solve LP (5) with all $\alpha_\omega^0 = \mathbf{0}$ for each scenario. Then, obtain the solution $(\mathbf{x}_0, \mathbf{z}^0)$.

Step 2. Set the upper bound $UB = c'\mathbf{x}_0 + \sum_{\omega\in\Omega} p_\omega z_\omega^0$.

Step 3. For x_0, we can obtain α_ω^1 and $z_\omega^{(0)}$ for each scenario, set the lower bound $LB = c'x_0 + \sum_{\omega\in\Omega} p_\omega z_\omega^{(0)}$

Step 4. For each ω, if $(\alpha_\omega^1)'(\mathbf{d}_\omega - \mathbf{x}_0\mathbf{1}) < z_\omega^0$, add one new constraint, $(\alpha_\omega^1)'(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega$, to RBMP.

Step 5. Solve the updated RBMP, obtain a new solution (x_1, z^1) and update UB.

Step 6. Repeat step 3 until $UB - LB < \epsilon$.(In our case, UB converges.)

Deterministic Formulation

When $|\Omega| = 1$ in problem (1), the stochastic programming will be

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

Deterministic Formulation

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

Substitute the first constraint with $\sum_{j=1}^{N} x_{ij} \geq s_i, i \in \mathcal{M}$, we can obtain the problem with lower bound supply.

Obtain the Feasible Seat Planning

Step 1. Obtain the solution, $\mathbf{x}^*$, from linear stochastic programming by benders decomposition.

Step 2. Aggregate the solution to the supply, $s_i^0 = \sum_j x_{ij}^*$.

Step 3. Obtain the optimal solution, $\mathbf{x}^1$, from problem (7) by setting the supply $\mathbf{s}^0$ as the upper bound.

Step 4. Aggregate the solution to the supply, $s_i^1 = \sum_j x_{ij}^1$.

Step 5. Obtain the optimal solution, $\mathbf{x}^2$, from problem (19) by setting the supply $\mathbf{s}^1$ as the lower bound.

Step 6. Aggregate the solution to the supply, $s_i^2 = \sum_j x_{ij}^2$, which is the feasible seat planning.

Dynamic Seat Assignment for Each Group Arrival

Group-type(Supply) Control

Feasible seat planning represents the supply for each group type. We can use supply control to determine whether to accept a group. Specifically, if there is a supply available for an arriving group, we will accept the group. However, if there is no corresponding supply for the arriving group, we need to decide whether to use a larger group supply to meet the group's needs. When a group is accepted to occupy larger-size seats, the remaining empty seat(s) can be reserved for future demand.

The difference of expected number of accepted people between acceptance and rejection on group i occupying $(j+\delta)$-size seats:
$d(i,j) = i + (j-i-\delta)P(D_{j-i-\delta} \geq x_{j-i-\delta}+1) - jP(D_j \geq x_j)$ if $j \geq i+\delta$; otherwise, $d(i,j) = i - jP(D_j \geq x_j)$. Find $d(i,j^*) = \max_j d(i,j)$, if $d(i,j^*) > 0$, accept group type i in $(j^*+\delta)$-size seats; otherwise, reject it.

Stochastic Planning Policy

Stochastic planning policy involves using the objective value of accepting or rejecting an arrival to make a decision. To determine this objective value, we need to consider the potential outcomes that could result from accepting the current arrival (i.e., the Value of Acceptance), as well as the potential outcomes that could result from rejecting it (i.e., the Value of Rejection).

The Value of Acceptance considers the scenarios that could arise if we accept the current arrival, while the Value of Rejection considers the same scenarios if we reject it. By comparing the Value of Acceptance and the Value of Rejection, we can make an informed decision about whether to accept or reject the arrival based on which option has the higher objective value.

Dynamic Seat Assignment for Each Group Arrival

Step 1. Obtain the set of patterns, $\mathbf{P} = \{P_1, \ldots, P_N\}$, from the feasible seat planning algorithm. The corresponding aggregate supply is $\mathbf{X} = [x_1, \ldots, x_M]$.

Step 2. For the arrival group type i at period T', If $\exists k \in \mathcal{N}$ such that $i \in P_k$, accept the group, update $P_k = P_k/(i)$ and $x_i = x_i - 1$. Go to step 4. Otherwise, go to step 3.

Step 3. Calculate $d(i, j^*)$. If $d(i, j^*) > 0$, find the first $k \in \mathcal{N}$ such that $j^* \in P_k$. If value of acceptance is larger than value of rejection, accept group type i and update $P_k = P_k/(j^*)$, $x_{j^*} = x_{j^*} - 1$. Then update $x_{j^*-i-\delta} = x_{j^*-i-\delta} + 1$ and $P_k = P_k \cup (j^* - i - \delta)$ when $j^* - i - \delta > 0$. If $d(i, j^*) \leq 0$, reject group type i.

Step 4. If $T' \leq T$, move to next period, set $T' = T' + 1$, go to step 2. Otherwise, terminate this algorithm.

Bid-price Control

The dual problem of linear relaxation of problem (7) 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 (7) takes the form $xe_h + \sum_{i=h+1}^{M} d_i e_i$, $x = (L - \sum_{i=h+1}^{M} d_i n_i)/n_h$.
The bid-price policy will make the decision to accept group type i, where i is greater than or equal to h, if the capacity allows.

Dynamic Programming Base-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.

Results

Running time of Benders Decomposition and IP

# of scenarios	demands	running time of IP(s)	Benders (s)	# of rows	# of group
1000	(150, 350)	5.1	0.13	30	8
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
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
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.

Results of Different Policies

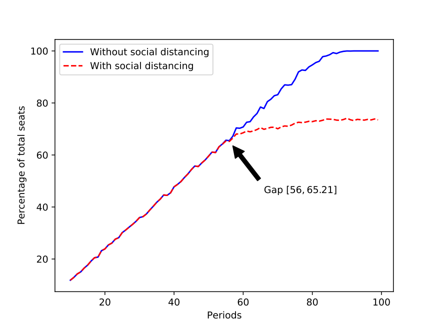
Table: Results of stochastic planning versus bid-price

T	probabilities	Sto(%)	DP1(%)	Bid-price(%)	FCFS(%)
60	[0.25, 0.25, 0.25, 0.25]	99.12	98.42	98.38	98.17
70	[0.25, 0.25, 0.25, 0.25]	98.34	96.87	96.24	94.75
80	[0.25, 0.25, 0.25, 0.25]	98.61	95.69	96.02	93.18
60	[0.25, 0.35, 0.05, 0.35]	98.94	98.26	98.25	98.62
70	[0.25, 0.35, 0.05, 0.35]	98.05	96.62	96.06	93.96
80	[0.25, 0.35, 0.05, 0.35]	98.37	96.01	95.89	92.88
60	[0.15, 0.25, 0.55, 0.05]	99.14	98.72	98.74	98.07
70	[0.15, 0.25, 0.55, 0.05]	99.30	96.38	96.90	96.25
80	[0.15, 0.25, 0.55, 0.05]	99.59	97.75	97.87	95.81

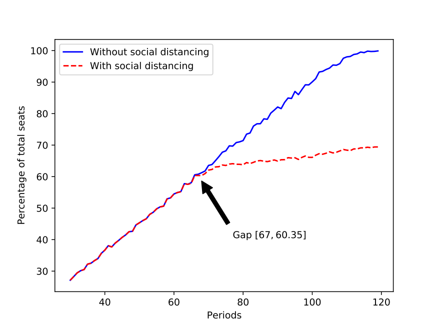
We compare the performance of different policies to the optimal value. Specifically, we consider two policies for seat assignment after all group arrivals: DSA and FCFS. In addition, we evaluate two policies for seat assignment for each group arrival: one based on dynamic programming (DP) and the other based on first-come, first-served (FCFS) scheduling.

Result of Different Demands

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



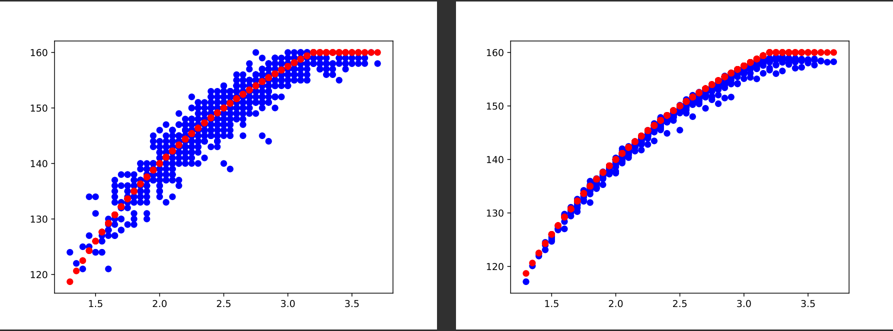
(a) When $c = 2.5$



(b) When $c = 1.9$

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.

Results of the Number of Arriving People per Period

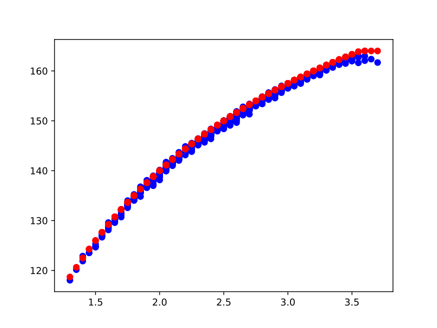


(c) One instance for each probability combination

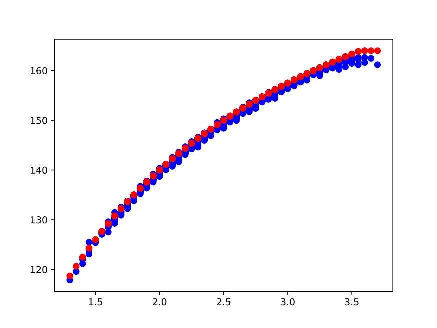
(d) Average of 50 instances for each probability combination

Figure: The number of people served versus c

Results of Different Seat Layouts



(a) Average of 50 instances for step-size seat layout



(b) Average of 50 instances for random seat layout

Figure: The number of people served versus c

The End