Some Analyses

June 14, 2025

1 Seat Planning Problem with Social Distancing

1.1 Concepts

Consider a seat layout comprising N rows, with each row j containing L_j^0 seats, for $j \in \mathcal{N} := \{1, 2, \ldots, N\}$. The venue will hold an event with multiple seat requests, where each request includes a group of multiple people. There are M distinct group types, where each group type i, $i \in \mathcal{M} := \{1, 2, \ldots, M\}$, consists of i individuals requiring i consecutive seats in one row. The request of each group type is represented by a demand vector $\mathbf{d} = (d_1, d_2, \ldots, d_M)^\intercal$, where d_i is the number of groups of type i.

We now formulate the seat planning with deterministic requests (SPDR) problem as an integer programming, where x_{ij} represents the number of groups of type i planned in row j.

$$
\begin{aligned}
\text{(SPDR)} \quad \max \quad & \sum_{i=1}^{M} \sum_{j=1}^{N} (n_i - \delta) x_{ij} && (1) \\
\text{s.t.} \quad & \sum_{j=1}^{N} x_{ij} \leq d_i, \quad i \in \mathcal{M}, && (2) \\
& \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N}, && (3) \\
& x_{ij} \in \mathbb{N}, \quad i \in \mathcal{M}, j \in \mathcal{N}.
\end{aligned}
$$

The objective function (1) is to maximize the number of individuals accommodated. Constraint (2) ensures the total number of accommodated groups does not exceed the number of requests for each group type. Constraint (3) stipulates that the number of seats allocated in each row does not exceed the size of the row.

The increasing nature of the ratio $\frac{i}{n_i}$ with respect to group size i leads to preferential inclusion of larger groups in the optimal fractional seat plan. This intuitive property is illustrated in Proposition 1.

Proposition 1. *For the LP relaxation of the* SPDR *problem, there exists an index $\tilde{i}$ such that the optimal solutions satisfy the following conditions: $x^*_{ij} = 0$ for all j, $i = 1, \ldots, \tilde{i} - 1$; $\sum_{j=1}^{N} x^*_{ij} = d_i$ for $i = \tilde{i} + 1, \ldots, M$; $\sum_{j=1}^{N} x^*_{ij} = \frac{L - \sum_{i=\tilde{i}+1}^{M} d_i n_i}{n_{\tilde{i}}}$ for $i = \tilde{i}$.*

2 Seat Assignment with Dynamic Requests

In many commercial situations, requests arrive sequentially over time, and the seller must immediately decide whether to accept or reject each request upon arrival while ensuring compliance with the required spacing constraints. If a request is accepted, the seller must also determine the specific seats to assign. Importantly, each request must be either fully accepted or entirely rejected; once seats are assigned to a group, they cannot be altered or reassigned to other requests.

To model this problem, we formulate it using dynamic programming approach in a discrete-time framework. Time is divided into T periods, indexed forward from 1 to T. We assume that in each period, at most one request arrives and the probability of an arrival for a group type i is denoted as p_i, where $i \in \mathcal{M}$. The probabilities satisfy the constraint $\sum_{i=1}^{M} p_i \leq 1$, indicating that the total probability of any group arriving in a single period does not exceed one. We introduce the probability $p_0 = 1 - \sum_{i=1}^{M} p_i$ to represent the probability of no arrival in each period. To simplify the analysis, we assume that the arrivals of different group types are independent and the arrival probabilities remain constant over time. This assumption can be extended to consider dependent arrival probabilities over time if necessary.

The remaining capacity in each row is represented by a vector $\mathbf{L} = (l_1, l_2, \ldots, l_N)$, where l_j denotes the number of remaining seats in row j. Upon the arrival of a group type i at time t, the seller needs to make a decision denoted by $u_{i,j}^t$, where $u_{i,j}^t = 1$ indicates acceptance of group type i in row j during period t, while $u_{i,j}^t = 0$ signifies rejection of that group type in row j. The feasible decision set is defined as

$$U^t(\mathbf{L}) = \left\{ u_{i,j}^t \in \{0,1\}, \forall i \in \mathcal{M}, \forall j \in \mathcal{N} \,\middle|\, \sum_{j=1}^{N} u_{i,j}^t \leq 1, \forall i \in \mathcal{M}; n_i u_{i,j}^t \mathbf{e}_j \leq \mathbf{L}, \forall i \in \mathcal{M}, \forall j \in \mathcal{N} \right\}.$$

Here, $\mathbf{e}_j$ represents an N-dimensional unit column vector with the j-th element being 1, i.e., $\mathbf{e}_j = (\underbrace{0, \cdots, 0}_{j-1}, 1, \underbrace{0, \cdots, 0}_{N-j})$. The decision set $U^t(\mathbf{L})$ consists of all possible combinations of acceptance and rejection decisions for each group type in each row, subject to the constraints that at most one group of each type can be accepted in any row, and the number of seats occupied by each accepted group must not exceed the remaining capacity of the row.

Let $V^t(\mathbf{L})$ denote the maximum expected revenue earned by the optimal decision regarding group seat assignments at the beginning of period t, given the remaining capacity $\mathbf{L}$. Then, the dynamic programming formulation for this problem can be expressed as:

$$V^t(\mathbf{L}) = \max_{u_{i,j}^t \in U^t(\mathbf{L})} \left\{ \sum_{i=1}^{M} p_i \Big(\sum_{j=1}^{N} i u_{i,j}^t + V^{t+1}(\mathbf{L} - \sum_{j=1}^{N} n_i u_{i,j}^t \mathbf{e}_j) \Big) + p_0 V^{t+1}(\mathbf{L}) \right\} \tag{4}$$

with the boundary conditions $V^{T+1}(\mathbf{L}) = 0, \forall \mathbf{L}$, which implies that the revenue at the last period is 0 under any capacity. The initial capacity is denoted as $\mathbf{L}_0 = (L_1, L_2, \ldots, L_N)$. Our objective is to determine group assignments that maximize the total expected revenue during the horizon from period 1 to T, represented by $V^1(\mathbf{L}_0)$.

Solving the dynamic programming problem in equation (4) presents computational challenges due to the curse of dimensionality that arises from the large state space. To address this, we develop a relaxed dynamic programming formulation and propose the Seat-Plan-Based Assignment (SPBA) policy. This policy combines the relaxed DP for preliminary acceptance decisions with the seat plan that serves as the basis for the final assignment decision.

2.1 Relaxed Dynamic Programming

To simplify the complexity of the dynamic programming formulation in (4), we employ a relaxed dynamic programming (RDP) approach by aggregating all rows into a single row with the total capacity $\tilde{L} = \sum_{j=1}^{N} L_j$. This relaxation yields preliminary seat assignment decisions for each group arrival, where the rejection by the RDP is final (no further evaluation is needed), the acceptance by the RDP is tentative and must be validated according to the current seat plan in the subsequent group-type control.

Let $u_i^t \in$ denote the RDP's decision variable for accepting ($u_i^t = 1$) or rejecting ($u_i^t = 0$) a type i request in period t. The value function of the relaxed DP with the total capacity l in period t, denoted by $V^t(l)$, is the following:

$$V^t(l) = \max_{u_i^t \in \{0,1\}} \left\{ \sum_{i=1}^{M} p_i \left[V^{t+1}(l - n_i u_i^t) + i u_i^t \right] + p_0 V^{t+1}(l) \right\} \tag{5}$$

with the boundary conditions $V^{T+1}(l) = 0, \forall l \geq 0$ and $V^t(0) = 0, \forall t$.

To make the initial decision, we compute the value function $V^t(l)$ and compare the values of accepting versus rejecting the request. Preliminarily accepted requests are then verified and assigned in the subsequent group-type control stage.

References