# Dynamic Seat Assignment With Social Distancing

Dis· count

August 14, 2023


## Abstract

This study addresses the dynamic seat assignment problem with social distancing, which arises when groups arrive at a venue and need to be seated together while respecting minimum physical distance requirements. To tackle this challenge, we develop a scenario-based method for generating seat plans and propose a seat assignment policy for accepting or denying arriving groups. We also explore a relaxed setting where seat assignments can be made after the booking period. We found that the Dynamic Seat Assignment (DSA) approach performs well compared with the offline optimal solution, achieving an occupancy rate of over 70% when total demand exceeds the number of seats and there are at least 2 people in each group. The results provide insights for policymakers and venue managers on seat utilization rates and offer a practical tool for implementing social distancing measures while optimizing seat assignments and ensuring group safety.


Keywords: Social Distancing, Scenario-based Stochastic Programming, Seat Assignment, Dynamic Arrival.

## 1 Introduction

Governments worldwide have been faced with the challenge of reducing the spread of Covid-19 while minimizing the economic impact. Social distancing has been widely implemented as the most effective non-pharmaceutical treatment to reduce the health effects of the virus. This website records a timeline of Covid-19 and the relevant epidemic prevention measures [18]. For instance, in March 2020, the Hong Kong government implemented restrictive measures such as banning indoor and outdoor gatherings of more than four people, requiring restaurants to operate at half capacity. As the epidemic worsened, the government tightened measures by limiting public gatherings to two people per group in July 2020. As the epidemic subsided, the Hong Kong government gradually relaxed social distancing restrictions, allowing public group gatherings of up to four people in September 2020. In October 2020, pubs were allowed to serve up to four people per table, and restaurants could serve up to six people per table.

Specifically, the Hong Kong government also implemented different measures in different venues [14]. For example, the catering businesses will have different social distancing requirements depending on their mode of operation for dine-in services. They can operate at 50%, 75%, or 100% of their normal seating capacity at any one time, with a maximum of 2, 2, or 4 people per table, respectively. Bars and pubs may open with a maximum of 6 persons per table and a total number of patrons capped at 75% of their capacity. The restrictions on the number of persons allowed in premises such as cinemas, performance venues, museums, event premises, and religious premises will remain at 85% of their capacity.

The measures announced by the Hong Kong government mainly focus on limiting the number of people in each group and the seat occupancy rate. However, implementing these policies in operations can be challenging, especially for venues with fixed seating layouts. In our study, we will focus on addressing this challenge in commercial premises, such as cinemas and music concert venues.

We aim to provide a practical tool for venues to optimize seat assignments while ensuring the safety of groups by proposing a seat assignment policy that takes into account social distancing requirements and the given seating layout. Additionally, we will offer guidance on setting appropriate occupancy rates and group sizes. We strive to enable venues to implement social distancing measures effectively by providing a tailored solution that accommodates their specific seating arrangements and operational constraints.

Some papers use seat assignment to express seat planning, to avoid confusion about seat planning and seat assignment, we clarify the difference between them. Seat planning involves determining the best layout and arrangement of seats in a venue or space based on factors such as the size of the room, the number of attendees, and the type of event. This can include deciding on the number of rows and columns, the spacing between seats, and any special requirements such as seats partition with social distancing. Seat assignment, on the other hand, involves assigning specific seats to attendees based on factors such as ticket type and availability. This is typically done closer to the event date and can involve a variety of methods such as manual seat selection by the attendee or an automated system that assigns seats based on predetermined criteria. For deterministic demand, seat planning equals seat assignment.

When purchasing tickets for movies or concerts, there are generally two approaches to seat assignment: seat assignment after all groups arrived and seat assignment for each group arrival.

The seat assignment after all groups arrived involves delaying seat assignment until the reservation deadline has passed. This means that the organizer does not need to immediately allocate seats to customers, but has to make the decision to accept or deny, so implementing social distancing restrictions will not affect the booking process. After the reservation deadline, the seller will inform customers of the seat layout information before admission. For instance, in venues such as singing concert halls, where there is high ticket demand and numerous seats available, organizers usually do not determine the seats during booking. Instead, they will inform customers of the seat information after the overall demands are determined. This approach allows for more flexibility in seat assignments and can accommodate changes in group sizes or preferences. However, in other venues where seating options are limited, it may be necessary to assign seats immediately upon accepting a group.

On the other hand, the seat assignment for each group arrival typically involves the cinema releasing the seating charts online, which show the available and unavailable seats, when there are no social distancing requirements. Customers can then choose their desired seats and reserve them by paying for their tickets. After successful payment, the seats are allocated to the customers. However, due to social distancing requirements, this approach needs to be modified. Seat assignments are still arranged when groups book their tickets, but the seller will provide the seat information directly. For example, in movie theaters with relatively few seats, the demands for tickets are usually low enough to allow for free selection of seats directly online. Early seat planning can satisfy the requirement of social distancing and save costs without changing seat allocation. The seat allocation could remain for one day because the same film genre will likely attract similar groups with similar seating preferences.

Our study mainly focuses on the latter situation where customers come dynamically, and the seat assignment needs to be made immediately without knowing the number and composition of future customers. In Section 6, we also consider the situation where the seat assignment can be made after the booking period.

This paper focuses on addressing the dynamic seating assignment problem with a given set of seats in the context of a pandemic. The government issues a maximum number of people allowed in each group and a maximum capacity percentage, which must be implemented in the seat planning. The problem becomes further complicated by the existence of groups of guests who can sit together.

To address this challenge, we have developed a mechanism for seat planning. Our proposed algorithm includes a solution approach to balance seat utilization rates and the associated risk of infection. Our goal is to obtain the final seating plan that satisfies social distancing constraints and implement the seat assignment when groups arrive.

Our approach provides a practical tool for venues to optimize seat assignments while ensuring the safety of their customers. The proposed algorithm has the potential to help companies and governments optimize seat assignments while maintaining social distancing measures and ensuring the safety of groups. Overall, our study offers a comprehensive solution for dynamic seat assignment with social distancing in the context of a pandemic.

Our main contributions in this paper are summarized as follows:

First, this study presents the first attempt to consider the arrangement of seat assignments with social distancing under dynamic arrivals. While many studies in the literature highlight the importance of social distancing in controlling the spread of the virus, they often focus too much on the model and do not provide much insight into the operational significance behind social distancing [1, 11]. Recent studies have explored the effects of social distancing on health and economics, mainly in the context of aircraft [13, 27, 28]. Our study provides a new perspective to help the government adopt a mechanism for setting seat assignments to protect people in the post-pandemic era.

Second, we establish a deterministic model to analyze the effects of social distancing when the demand is known. Due to the medium size of the problem, we can solve the IP model directly. We then consider the stochastic demand situation where the demands of different group types are random. By using two-stage stochastic programming and Benders decomposition methods, we obtain the optimal linear solution.

Third, to address the dynamic scenario problem, we first obtain a feasible seating plan using scenario-based stochastic programming. We then make a decision for each incoming group based on a nested policy, either accepting or rejecting the group. Our results demonstrate a significant improvement over a first-come first-served baseline strategy and provide guidance on how to develop attendance policies.

The rest of this paper is structured as follows. The following section reviews relevant literature. We describe the motivating problem in Section 3. In Section 4, we establish the stochastic model,analyze its properties and give the seating planning. Section 5 demonstrates the dynamic seat assignment during booking period and after booking period. Section 6 gives the results. The conclusions are shown in Section 7.

# 2 Literature Review

The present study is closely connected to the following research areas – seat planning with social distancing and dynamic seat assignment. The subsequent sections review literature pertaining to each perspective and highlight significant differences between the present study and previous research.

## 2.1 Seat Planning with Social Distancing

Since the outbreak of covid-19, social distancing is a well-recognized and practiced method for containing the spread of infectious diseases [25]. An example of operational guidance is ensuring social distancing in seating plans.

Social distancing in seat planning has attacted considerable attention from the research area. The applications include the allocation of seats on airplanes [13], classroom layout planning [4], seat planning in long-distancing trains [16]. The social distancing can be implemented in various forms, such as fixed distances or seat lengths. Fischetti et al. [11] consider how to plant positions with social distancing in restaurants and beach umbrellas. Different venues may require different forms of social distancing; for instance, on an airplane, the distancing between seats and the aisle must be considered [27], while in a classroom, maximizing social distancing between students is a priority [4].

These researchs focus on the static version of the problem. This typically involves creating an IP model with social distancing constraints( [4, 13, 16]), which is then solved either heuristically or directly. The seat allocation of the static form is useful for fixed people, for example, the students in one class. But it is not be practical for the dynamic arrivals in commercial events.

The recent pandemic has shed light on the benefits of group reservations, as they have been shown to increase revenue without increasing the risk of infection [24]. In our specific setting, we require that groups be accepted on an all-or-none basis, meaning that members of the same family or group must be seated together. However, the group seat reservation policy poses a significant challenge when it comes to determining the seat assignment policy.

This group seat reservation policy has various applications in industries such as hotels [23], working spaces [11], public transport [9], sports arenas [21], and large-scale events [22]. This policy has significant impacts on passenger satisfaction and revenue, with the study [31] showing that passenger groups increase revenue by filling seats that would otherwise be empty. Traditional works [6,9]in transportation focus on maximizing capacity utilization or reducing total capacity needed for passenger rail, typically modeling these problems as knapsack or binpacking problems.

Some related literature mentioned the seat planning under pandemic for groups are represented below. Fischetti et al. [11] proposed a seating planning for known groups of customers in amphitheaters. Haque and Hamid [16] considers grouping passengers with the same origin-destination pair of travel and assigning seats in long-distance passenger trains. Salari et al. [27] performed group seat assignment in airplanes during the pandemic and found that increasing passenger groups can yield greater social distancing than single passengers. Haque and Hamid [17] aim to optimize seating assignments on trains by minimizing the risk of virus spread while maximizing revenue. The specific number of groups in their models is known in advance. But in our study, we only know the arrival probabilities of different groups.

This paper [3] discusses strategies for filling a theater by considering the social distancing and group arrivals, which is similar to ours. However, unlike our project, it only focuses on a specific location layout and it is still based on a static situation by giving the proportion of different groups.

## 2.2 Dynamic Seat Assignment

Our model in its static form can be viewed as a specific instance of the multiple knapsack problem [26], where we aim to assign a subset of groups to some distinct rows. In our dynamic form, the decision to accept or reject groups is made at each stage as they arrive. The related problem can be dynamic knapsack problem [20], where there is one knapsack.

Dynamic seat assignment is a process of assigning seats to passengers on a transportation vehicle, such as an airplane, train, or bus, in a way that maximizes the efficiency and convenience of the seating arrangements [2, 15, 32].

Our problem is closely related to the network revenue management (RM) problem [30], which is typically formulated as a dynamic programming (DP) problem. However, for large-scale problems, the exponential growth of the state space and decision set makes the DP approach computationally intractable. To address this challenge, we propose using scenario-based programming [5, 10, 19] to determine the seat planning. In this approach, the aggregated supply can be considered as a protection level for each group type. Notably, in our model, the supply of larger groups can also be utilized by smaller groups. This is because our approach focuses on group arrival rather than individual unit, which sets it apart from traditional partitioned and nested approaches [7, 29].

Traditional revenue management focuses on decision-making issues, namely accepting or rejecting a request [12].However, our paper not only addresses decision-making, but also emphasizes the significance of assignment, particularly in the context of seat assignment. This sets it apart from traditional revenue management methods and makes the problem more challenging.

Similarly, the assign-to-seat approach introduced by Zhu et al. [32] also highlights the importance of seat assignment in revenue management. This approach addresses the challenge of selling high-speed train tickets in China, where each request must be assigned to a single seat for the entire journey and takes into account seat reuse. This further emphasizes the significance of seat assignment and sets it apart from traditional revenue management methods.

# 3 Problem Description

In this section, to incorporate the social distancing into seat planning, we first give the description of the seat planning problem with social distancing. Then we introduce the dynamic seat assignment problem with social distancing.

## 3.1 Seat Planning Problem with Social Distancing

We consider a set of groups, each of which consists of no more than $M$ people, to be assigned to a set of seats. There are $M$ different group types, with group type $i$ containing $i$ people, where $i \in \mathcal{M} := \{1, 2, \ldots, M\}$. (We use $\mathcal{M} = \{1, \ldots, M\}$ to denote the set of all positive integers that are no larger than $M$.)

These groups can be represented by a demand vector, denoted by $\mathbf{d} = [d_1, \ldots, d_M]$. Each element $d_i$, where $i \in \mathcal{M}$, indicates the number of group type $i$. For illustration, we consider a layout consisting of $N$ rows, each containing $S_j$ seats, where $j \in \mathcal{N}$.

In accordance with epidemic prevention requirements, customers from the same group are allowed to sit together, while different groups must maintain social distancing. Let $\delta$ denote the social distancing, which can be one seat or more seats.

Specifically, each group must leave empty seat(s) to maintain social distancing from adjacent groups. Additionally, different rows do not affect each other, meaning that a person from one group can sit directly behind a person from another group.

To achieve the social distancing requirements in the seat planning process, we add $\delta$ to the original size of each group to create the new size of the group. Let $n_i = i + \delta$ denote the new size of group type $i$ for each $i \in \mathcal{M}$. Construct new seat layout by adding $\delta$ to each row, i.e., let $L_j = S_j + \delta$ denote the length of row $j$ for each $j \in \mathcal{N}$, where $S_j$ represents the number of seats in row $j$.

Then we can illustrate the seat planning for one row below.

Figure 1: Problem Conversion

The social distancing here is one seat. On the left side of the diagram, the blue squares represent the empty seats required for social distancing, while the orange squares represent the seats occupied by groups. On the right side, we have added one dummy seat at the end of each row. The orange squares surrounded by the red line represent the seats taken by groups in this row, which includes two groups of 1, one group of 2, and one group of 3.

By incorporating the additional seat and designating certain seats for social distancing, we can integrate social distancing measures into the seat planning problem.

Now, we analyse the effect of introducing social distancing for each row. At first, we consider the types of pattern, which refers to the seat planning for each row. For each pattern $k$, we use $\alpha_k, \beta_k$ to indicate the number of groups and the left seats, respectively. Denote by $l(k) = \alpha_k \delta + \beta_k - \delta$ the loss for pattern $k$. 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. Then we call the patterns from $I_1$ are the largest. Similarly, the patterns from $I_2$ are the second largest, so forth and so on. 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 1.** *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* $I_1$*.*

We can use the following greedy way to generate the largest pattern. Select the maximal group size, $n_M$, as many as possible and the left space is assigned to the group with the corresponding size. Let $L = n_M \cdot q + r, 0 \leq r < n_M$, where $q$ is the quotient representing the number of times $n_M$ selected and $r$ is the remainder representing the number of remaining seats. The loss of the largest pattern is $q\delta - \delta + f(r)$, where $f(r) = 0$ if $r > \delta$, and $f(r) = r$ if $r \leq \delta$.

**Proposition 1.** *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* $\sum_j (S_j - \lfloor \frac{S_j+\delta}{n_M} \rfloor + \delta - f((S_j + \delta) \mod n_M))$*.*

## 3.2 Dynamic Seat Assignment with Social Distancing

Consider a scenario in which groups arrive dynamically and a decision-maker must determine whether to accept or reject each group and assign them to empty seats in some row while ensuring that the social distancing constraint is met. Once seats are confirmed and assigned to a group, they cannot be changed. To keep track of the remaining capacity of rows, we use a vector $\mathbf{L} = (L_1^r, L_2^r, \ldots, L_N^r)$, where $L_j^r$ represents the number of remaining seats in row $j$. There are $T$ periods, and exactly one group request for each period. We assume that arrivals of different group types are independent, and the arrival probability of group type $i$ in each period is $p_i$. Let $V_t(\mathbf{L})$ denote the maximal expected value at period $t$ with the capacity of rows.

In every period, a decision is made on whether to accept or reject a group and which row to assign the group to. Let $u_{i,j}$ denote the decision, where $u_{i,j}(t) = 1$ if we accept a group type $i$ in row $j$ at period $t$, and $u(t) = 0$ otherwise.

The dynamic programming formulation for this problem is

$$V_t(\mathbf{L}) = \max_{u \in U(\mathbf{L})} \{\sum_{i=1}^{M} p_i (\sum_{j=1}^{N} i u_{i,j} + V_{t+1}(\mathbf{L} - \sum_{j=1}^{N} n_i u_{i,j} \mathbf{e}_j^\top))\}, \mathbf{L} \geq 0, V_{T+1}(\mathbf{L}) = 0, \forall \mathbf{L}$$

The decision set $U(\mathbf{L}) = \{u_{i,j} \in \{0,1\}, \forall i,j | \sum_{j=1}^N u_{i,j} \leq 1, \forall i, n_i u_{i,j} \mathbf{e}_j^\top \leq \mathbf{L}, \forall i,j\}$ and $\mathbf{e}_j$ is the N-dimensional unit row vector with $j$-th element being 1.

The compact form can be written as:

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

Initially, we have $\mathbf{L}_T = (L_1, L_2, \ldots, L_N)$. As we can observe, the dynamic programming algorithm has to make a decision on which row to assign group type $i$. This leads to the curse of dimensionality due to the numerous seat planning combinations. To avoid this complexity, we propose an approach that directly targets the final seat planning and then formulate a policy for assigning groups. To obtain the final seat planning firstly, we develop the scenario-based stochastic programming.

# 4 Scenario-based Stochastic Programming

This section focuses on seat planning in scenarios with varying demand. We begin by introducing a scenario-based stochastic programming formulation. However, due to its time-consuming nature, we transform it into a two-stage problem and implement benders decomposition to obtain the optimal linear solution. Using this solution, we generate a feasible seat planning.

## 4.1 Formulation

Now suppose the demand of groups is stochastic, the stochastic information can be obtained from scenarios through historical data. Use $\omega$ to index the different scenarios, each scenario $\omega \in \Omega$. A particular realization of the demand vector can be represented as $\mathbf{d}_\omega = (d_{1\omega}, d_{2\omega}, \ldots, d_{M,\omega})^\intercal$. Let $p_\omega$ denote the probability of any scenario $\omega$, which we assume to be positive. To maximize the expected value of people over all the scenarios, we propose a scenario-based stochastic programming.

Consider the decision makers who give the seat planning based on the scenarios then assign the groups to seats according to the realized true demand.

The seat planning can be denoted by decision variables $\mathbf{x} \in \mathbb{Z}_+^{M\times N}$. Let $x_{i,j}$ stand for the number of group type $i$ in row $j$. The supply for group type $i$ can be represented by $\sum_{j=1}^N x_{ij}$. Regarding the nature of the obtained information, we assume that there are $S = |\Omega|$ possible scenarios. There is a scenario-dependent decision variable, $\mathbf{y}$, to be chosen. It includes two vectors of decisions, $\mathbf{y}^+ \in \mathbb{Z}_+^{M\times S}$ and $\mathbf{y}^- \in \mathbb{Z}_+^{M\times S}$. Each component of $\mathbf{y}^+$, $y_i^{\omega(+)}$, represents the number of surplus seats for group type $i$. Similarly, $y_i^{\omega(-)}$ represents the number of inadequate seats for group type $i$. Considering that the group can take the seats planned for the larger group type, we assume that the surplus seats for group type $i$ can be occupied by smaller group type $j < i$ in the descending order of the group size. That is, for any $\omega$, $i \leq M-1$, $y_{i\omega}^+ = \left(\sum_{j=1}^N x_{ij} - d_{i\omega} + y_{i+1,\omega}^+\right)^+$ and $y_{i\omega}^- = \left(d_{i\omega} - \sum_{j=1}^N x_{ij} - y_{i+1,\omega}^+\right)^+$, where $(x)^+$ equals $x$ if $x > 0$, 0 otherwise. Specially, for the largest group type $M$, we have $y_{M\omega}^+ = (\sum_{j=1}^N x_{ij} - d_{i\omega})^+$, $y_{M\omega}^- = (d_{i\omega} - \sum_{j=1}^N x_{ij})^+$.

Then we have the deterministic equivalent form(DEF) of the scenario-based stochastic programming:

$$
\begin{aligned}
(DEF) \quad \max \quad & E_\omega \left[ \sum_{i=1}^{M-1} (n_i - \delta)(\sum_{j=1}^{N} x_{ij} + y^+_{i+1,\omega} - y^+_{i\omega}) + (n_M - \delta)(\sum_{j=1}^{N} x_{Mj} - y^+_{M\omega}) \right] && (1)\\
\text{s.t.} \quad & \sum_{j=1}^{N} x_{ij} - y^+_{i\omega} + y^+_{i+1,\omega} + y^-_{i\omega} = d_{i\omega}, \quad i = 1, \ldots, M-1, \omega \in \Omega && (2)\\
& \sum_{j=1}^{N} x_{ij} - y^+_{i\omega} + y^-_{i\omega} = d_{i\omega}, \quad i = M, \omega \in \Omega && (3)\\
& \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} && (4)\\
& y^+_{i\omega}, y^-_{i\omega} \in \mathbb{Z}_+, \quad i \in \mathcal{M}, \omega \in \Omega && (5)\\
& x_{ij} \in \mathbb{Z}_+, \quad i \in \mathcal{M}, j \in \mathcal{N}. && (6)
\end{aligned}
$$

The objective function contains two parts, the number of the largest group type that can be accommodated is $\sum_{j=1}^{N} x_{Mj} - y^+_{M\omega}$. The number of group type $i$ that can be accommodated is $\sum_{j=1}^{N} x_{ij} + y^+_{i+1,\omega} - y^+_{i\omega}$. $E_\omega$ is the expectation with respect to the scenario set.

By reformulating the objective function, we have

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

The last equality holds because of $n_i - \delta = i, i \in \mathcal{M}$.

**Remark 1.** *For any $i, \omega$, at most one of $y^+_{i\omega}$ and $y^-_{i\omega}$ can be positive. Suppose there exist $i_0$ and $\omega_0$ such that $y^+_{i_0\omega_0}$ and $y^-_{i_0\omega_0}$ are positive. Substracting $\min\{y^+_{i_0,\omega_0}, y^-_{i_0,\omega_0}\}$ from these two values will still satisfy constraints (2) and (3) but increase the objective value when $p_{\omega_0}$ is positive. Thus, at most one of $y^+_{i\omega}$ and $y^-_{i\omega}$ can be positive.*

Let $\mathbf{n} = (n_1, \ldots, n_M)$, $\mathbf{L} = (L_1, \ldots, L_N)$ where $s_i$ is the size of seats taken by group type $i$ and $L_j$ is the length of row $j$ as we defined above. Then the constraint (4) can be expressed as $\mathbf{nx} \leq \mathbf{L}$.

The linear constraints associated with scenarios, i.e., constraints (2) and (3), can be written in a matrix form as

$$\mathbf{x1} + \mathbf{V}\mathbf{y}_\omega = \mathbf{d}_\omega, \omega \in \Omega,$$

where $\mathbf{1}$ is a column vector of size $N$ with all 1s, $\mathbf{V} = [\mathbf{W},\ \mathbf{I}]$.

$$\mathbf{W} = \begin{bmatrix} -1 & 1 & \dots & 0 \\ & \ddots & \ddots & \vdots \\ & & & 1 \\ 0 & & & -1 \end{bmatrix}_{M \times M}$$

and $\mathbf{I}$ is the identity matrix. For each scenario $\omega \in \Omega$,

$$\mathbf{y}_\omega = \begin{bmatrix} \mathbf{y}_\omega^+ \\ \mathbf{y}_\omega^- \end{bmatrix}, \mathbf{y}_\omega^+ = \begin{bmatrix} y_{1\omega}^+ & y_{2\omega}^+ & \cdots & y_{M\omega}^+ \end{bmatrix}^\intercal, \mathbf{y}_\omega^- = \begin{bmatrix} y_{1\omega}^- & y_{2\omega}^- & \cdots & y_{M\omega}^- \end{bmatrix}^\intercal.$$

As we can find, this deterministic equivalent form is a large-scale problem even if the number of possible scenarios $\Omega$ is moderate. However, the structured constraints allow us to simplify the problem by applying Benders decomposition approach. Before using this approach, let us write this problem in the form of the two-stage stochastic programming.

Let $\mathbf{c}^\intercal \mathbf{x} = \sum_{j=1}^{N} \sum_{i=1}^{M} i \cdot x_{ij}$, $\mathbf{f}^\intercal \mathbf{y}_\omega = -\sum_{i=1}^{M} y_{i\omega}^+$. Then the DEF formulation can be expressed as below,

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal \mathbf{x} + z(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{n}\mathbf{x} \leq \mathbf{L} \\ & \mathbf{x} \in \mathbb{Z}_+^{M \times N}, \end{aligned} \tag{7}$$

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 $\omega \in \Omega$,

$$\begin{aligned} z_\omega(\mathbf{x}) := \max \quad & \mathbf{f}^\intercal \mathbf{y}_\omega \\ \text{s.t.} \quad & \mathbf{x}\mathbf{1} + \mathbf{V}\mathbf{y}_\omega = \mathbf{d}_\omega \\ & \mathbf{y}_\omega \geq 0. \end{aligned} \tag{8}$$

Problem (8) stands for the second-stage problem and $z_\omega(\mathbf{x})$ is the optimal value of problem (8), together with the convention $z_\omega(\mathbf{x}) = \infty$ if the problem is infeasible.

Solving the above problem directly can be challenging, so we first obtain the optimal solution to the relaxation of problem (7). From this solution, we generate a seat planning.

## 4.2 Solve the Scenario-based Two-stage Problem

At first, we generate a closed-form solution to the second-stage problem in section 4.2.1. Then we obtain the solution to the linear relaxation of problem (7) by the delayed constraint generation. Finally, we obtain a feasible seat planning from the linear solution.

#### 4.2.1 Solve the Second Stage Problem

Consider a $\mathbf{x}$ such that $\mathbf{n}\mathbf{x} \leq \mathbf{L}$ and $\mathbf{x} \geq 0$ and suppose that this represents the seat planning. Once $\mathbf{x}$ is fixed, the optimal decisions $\mathbf{y}_\omega$ can be determined by solving problem (8) for each $\omega$.

Notice that the feasible region of the dual of problem (8) does not depend on $\mathbf{x}$. Let $\boldsymbol{\alpha}$ be the vector of dual variable. For each $\omega$, we can form its dual problem, which is

$$\begin{aligned} \min \quad & \boldsymbol{\alpha}^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \\ \text{s.t.} \quad & \boldsymbol{\alpha}^\intercal \mathbf{V} \geq \mathbf{f}^\intercal \end{aligned} \tag{9}$$

Let $\mathbb{P} = \{\boldsymbol{\alpha} \in \mathbb{R}^M | \boldsymbol{\alpha}^\intercal \mathbf{V} \geq \mathbf{f}^\intercal\}$. We assume that $\mathbb{P}$ is nonempty and has at least one extreme point. Then, either the dual problem (9) has an optimal solution and $z_\omega(\mathbf{x})$ is finite, or the primal problem (8) is infeasible and $z_\omega(\mathbf{x}) = \infty$.

Let $\mathcal{O}$ be the set of all extreme points of $\mathbb{P}$ and $\mathcal{F}$ be the set of all extreme rays of $\mathbb{P}$. Then $z_\omega > -\infty$ if and only if $\boldsymbol{\alpha}^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq 0, \boldsymbol{\alpha} \in \mathcal{F}$, which stands for the feasibility cut.

**Lemma 1.** *The feasible region of problem (9), $\mathbb{P}$, is bounded. In addition, all the extreme points of $\mathbb{P}$ are integral.*

(Proof of lemma 1)**.** *Notice that $\mathbf{f}^\intercal = [-\mathbf{1},\ \mathbf{0}], V = [W,\ I]$, $W$ is a totally unimodular matrix. Then, we have $\boldsymbol{\alpha}^\intercal W \geq -\mathbf{1}, \boldsymbol{\alpha}^\intercal I \geq \mathbf{0}$. Thus, the feasible region is bounded. Furthermore, let $\alpha_0 = 0$, then we have $0 \leq \alpha_i \leq \alpha_{i-1} + 1$, $i \in \mathcal{M}$, so the extreme points are all integral.* □

Because the feasible region is bounded, then feasibility cuts are not needed. Let $z_\omega$ be the lower bound of $z_\omega(x)$ such that $\boldsymbol{\alpha}^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega, \boldsymbol{\alpha} \in \mathcal{O}$, which is the optimality cut.

**Corollary 1.** *Only the optimality cuts, $\alpha^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega$, will be included in the decomposition approach.*

**Corollary 2.** *The optimal value of the problem (8), $z_\omega(x)$, is finite and will be attained at extreme points of the set $P$. Thus, we have $z_\omega(x) = \min_{\boldsymbol{\alpha} \in \mathcal{O}} \boldsymbol{\alpha}^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1})$.*

When we are given $\mathbf{x}^*$, the demand that can be satisfied by the assignment is $\mathbf{x}^*\mathbf{1} = \mathbf{d}_0 = (d_{1,0}, \ldots, d_{M,0})^\intercal$. Then plug them in the subproblem (8), we can obtain the value of $y_{i\omega}$ recursively:

$$\begin{aligned} y^-_{M\omega} &= (d_{M\omega} - d_{M0})^+ \\ y^+_{M\omega} &= (d_{M0} - d_{M\omega})^+ \\ y^-_{i\omega} &= \left(d_{i\omega} - d_{i0} - y^+_{i+1,\omega}\right)^+, i = 1, \ldots, M-1 \\ y^+_{i\omega} &= \left(d_{i0} - d_{i\omega} + y^+_{i+1,\omega}\right)^+, i = 1, \ldots, M-1 \end{aligned} \tag{10}$$

The optimal value for scenario $\omega$ can be obtained by $\mathbf{f}^\intercal y_\omega$, then we need to find the dual optimal solution.

**Theorem 1.** *The optimal solutions to problem* (9) *are given by*

$$\begin{aligned}
&\alpha_i = 0, i \in \mathcal{M} \quad \textit{if } y^-_{i\omega} > 0, y^+_{i\omega} = 0\\
&\alpha_i = \alpha_{i-1} + 1, i \in \mathcal{M} \quad \textit{if } y^+_{i\omega} > 0, y^-_{i\omega} = 0\\
&\alpha_i = 0, i = 1, \ldots, M-1 \quad \textit{if } y^-_{i\omega} = y^+_{i\omega} = 0, y^+_{i+1,\omega} > 0\\
&0 \le \alpha_i \le \alpha_{i-1} + 1, i = 1, \ldots, M-1 \quad \textit{if } y^-_{i\omega} = y^+_{i\omega} = 0, y^+_{i+1,\omega} = 0\\
&0 \le \alpha_i \le \alpha_{i-1} + 1, i = M \quad \textit{if } y^-_{i\omega} = y^+_{i\omega} = 0
\end{aligned} \tag{11}$$

(Proof of Theorem 1). *According to the complementary slackness property, we can obtain the following equations*

$$\begin{aligned}
&\alpha_i(d_{i0} - d_{i\omega} - y^+_{i\omega} + y^+_{i+1,\omega} + y^-_{i\omega}) = 0, i = 1, \ldots, M-1\\
&\alpha_i(d_{i0} - d_{i\omega} - y^+_{i\omega} + y^-_{i\omega}) = 0, i = M\\
&y^+_{i\omega}(\alpha_i - \alpha_{i-1} - 1) = 0, i = 1, \ldots, M\\
&y^-_{i\omega}\alpha_i = 0, i = 1, \ldots, M.
\end{aligned}$$

*When $y^-_{i\omega} > 0$, we have $\alpha_i = 0$; when $y^+_{i\omega} > 0$, we have $\alpha_i = \alpha_{i-1} + 1$. Let $\Delta d = d_\omega - d_0$, then the elements of $\Delta d$ will be a negative integer, positive integer and zero. When $y^+_{i\omega} = y^-_{i\omega} = 0$, if $i = M$, $\Delta d_M = 0$, the value of objective function associated with $\alpha_M$ is always 0, thus we have $0 \le \alpha_M \le \alpha_{M-1} + 1$; if $i < M$, we have $y^+_{i+1,\omega} = \Delta d_i \ge 0$. If $y^+_{i+1,\omega} > 0$, the objective function associated with $\alpha_i$ is $\alpha_i \Delta d_i = \alpha_i y^+_{i+1,\omega}$, thus to minimize the objective value, we have $\alpha_i = 0$; if $y^+_{i+1,\omega} = 0$, we have $0 \le \alpha_i \le \alpha_{i-1} + 1$.* □

We can use the forward method, calculating from $\alpha_{1\omega}$ to $\alpha_{M\omega}$, to obtain the value of $\alpha_\omega$ instead of solving the original large-scale linear programming.

#### 4.2.2 Delayed Constraint Generation

Benders decomposition works with only a subset of those exponentially many constraints and adds more constraints iteratively until the optimal solution of Benders Master Problem(BMP) is attained. This procedure is known as delayed constraint generation.

Use the characterization of $z_\omega(x)$ in the problem (7) and take into account the optimality cuts, we can conclude the BMP will have the form:

$$\begin{aligned}
\max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} p_\omega z_\omega\\
\text{s.t.} \quad & \mathbf{n}\mathbf{x} \le \mathbf{L}\\
& \boldsymbol{\alpha}^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \ge z_\omega, \boldsymbol{\alpha} \in \mathcal{O}, \forall \omega\\
& \mathbf{x} \ge 0, z_\omega \text{ is free}
\end{aligned} \tag{12}$$

When substituting $\mathcal{O}$ with its subset, $\mathcal{O}^t$, the problem (12) becomes the Restricted Benders Master

Problem(RBMP).

To determine the initial $\mathcal{O}^t$, we have the following lemma.

**Lemma 2.** *RBMP is always bounded with at least any one optimality cut for each scenario.*

(Proof of lemma 2)**.** *Suppose we have one extreme point $\boldsymbol{\alpha}_\omega^0$ for each scenario. Then we have the following problem.*

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} p_\omega z_\omega \\ s.t. \quad & \mathbf{n}\mathbf{x} \le \mathbf{L} \\ & (\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{d}_\omega \ge (\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{x}\mathbf{1} + z_\omega, \forall \omega \\ & \mathbf{x} \ge 0 \end{aligned} \tag{13}$$

*Problem (13) reaches its maximum when $(\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{d}_\omega = (\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{x}\mathbf{1} + z_\omega, \forall \omega$. Substitute $z_\omega$ with these equations, we have*

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal \mathbf{x} - \sum_{\omega} p_\omega (\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{x}\mathbf{1} + \sum_{\omega} p_\omega (\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{d}_\omega \\ s.t. \quad & \mathbf{n}\mathbf{x} \le \mathbf{L} \\ & \mathbf{x} \ge 0 \end{aligned} \tag{14}$$

*Notice that $\mathbf{x}$ is bounded by $\mathbf{L}$, then the problem (13) is bounded. Adding more optimality cuts will not make the optimal value larger. Thus, RBMP is bounded.* □

Given the initial $\mathcal{O}^t$, we can have the solution $\mathbf{x}_0$ and $\mathbf{z}^0 = (z_1^0, \ldots, z_S^0)$. Then $c^\intercal \mathbf{x}_0 + \sum_{\omega \in \Omega} p_\omega z_\omega^0$ is an upper bound of problem (12).

When $\mathbf{x}_0$ is given, the optimal solution, $\boldsymbol{\alpha}_\omega^1$, to problem (9) can be obtained according to Theorem 1. $z_\omega^{(0)} = \boldsymbol{\alpha}_\omega^1 (d_\omega - \mathbf{x}_0 \mathbf{1})$ and $(\mathbf{x}_0, \mathbf{z}^{(0)})$ is a feasible solution to problem (12) because it satisfies all the constraints. Thus, $\mathbf{c}^\intercal \mathbf{x}_0 + \sum_{\omega \in \Omega} p_\omega z_\omega^{(0)}$ is a lower bound of problem (12).

If for every scenario $\omega$, the optimal value of the corresponding problem (9) is larger than or equal to $z_\omega^0$, all contraints are satisfied, we have an optimal solution, $(\mathbf{x}_0, \mathbf{z}_\omega^0)$, to the BMP. Otherwise, add one new constraint, $(\boldsymbol{\alpha}_\omega^1)^\intercal (\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \ge z_\omega$, to RBMP.

The steps of the algorithm are described as below,

**Algorithm 1** The benders decomposition algorithm

**Step 1.** Solve LP (13) 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 $UB = c^\intercal \mathbf{x}_0 + \sum_{\omega\in\Omega} p_\omega z_\omega^0$.

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

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

**Remark 2.** *From the Lemma 2, we can set* $\boldsymbol{\alpha}_\omega^0 = \mathbf{0}$ *initially in* **Step 1**.

**Remark 3.** *Notice that only contraints are added in each iteration, thus* $LB$ *and* $UB$ *are both monotone. Then we can use* $UB - LB < \epsilon$ *to terminate the algorithm in* **Step 6**.

After the algorithm terminates, we obtain the optimal $\mathbf{x}^*$. The demand that can be satisfied by the arrangement is $\mathbf{x}^*\mathbf{1} = \mathbf{d}_0 = (d_{1,0}, \ldots, d_{M,0})$. Then we can obtain the value of $\mathbf{y}_\omega$ from equation (10).

### 4.3 Obtain the Feasible Seat Planning

The decomposition method only gives a fractional solution and the stochastic model does not provide an appropriate seat planning when the number of people in scenario demands is way smaller than the number of the seats. Thus, we change the linear solution from the decomposition method to obtain a feasible seat planning. Before that, we will discuss the deterministic model that can help to achieve the goal.

When $|\Omega| = 1$ in DEF formulation, the stochastic programming will be

$$\begin{aligned}
\max \quad & \sum_{i=1}^{M}\sum_{j=1}^{N}(n_i - \delta)x_{ij} - \sum_{i=1}^{M} y_i^+ \\
\text{s.t.} \quad & \sum_{j=1}^{N} x_{ij} - y_i^+ + y_{i+1}^+ + y_i^- = d_i, \quad i = 1, \ldots, M-1, \\
& \sum_{j=1}^{N} x_{ij} - y_i^+ + y_i^- = d_i, \quad i = M, \\
& \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} \\
& y_i^+, y_i^- \in \mathbb{Z}_+, \quad i \in \mathcal{M} \\
& x_{ij} \in \mathbb{Z}_+, \quad i \in \mathcal{M}, j \in \mathcal{N}.
\end{aligned} \tag{15}$$

To maximize the objective function, we can take $y_i^+ = 0$. Notice that $y_i^- \geq 0$, thus the constraints $\sum_{j=1}^{N} x_{ij} + y_i^- = d_i, i \in \mathcal{M}$ can be rewritten as $\sum_{j=1}^{N} x_{ij} \leq d_i, i \in \mathcal{M}$, then we have

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{M}\sum_{j=1}^{N}(n_i-\delta)x_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{N} x_{ij} \leq d_i, \quad i \in \mathcal{M}, \\
& \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} \\
& x_{ij} \in \mathbb{Z}_+, \quad i \in \mathcal{M}, j \in \mathcal{N}.
\end{aligned} \tag{16}
$$

Problem (16) represents the deterministic model. Demand, $d_i, i \in \mathcal{M}$ is known in advance, our goal is to accommodate as many as people possible in the fixed rows.

**Lemma 3.** *For the linear relaxation of problem* (16)*, there exists $h$ such that the optimal solutions $x_{ij}^* = 0$ when $i < h$; $\sum_j x_{ij}^* = d_i$, when $i > h$; $\sum_j x_{ij}^* = (L - \sum_{i=h+1}^{M} d_i n_i)/n_h$, when $i = h$.*

Let $\sum_{j=1}^{N} x_{ij}$ represent the supply for group type $i$. We define $\mathbf{X} = (\sum_{j=1}^{N} x_{1j}, \ldots, \sum_{j=1}^{N} x_{Mj})$ as the aggregate solution to the linear relaxation of problem (16). Furthermore, let $e_i$ denote the unit size of the $i$-th element of $\mathbf{X}$.

In the aggregate optimal solution, denoted as $xe_h + \sum_{i=h+1}^{M} d_i e_i$, the following components are present: $xe_h$: This term represents the allocation of resources for group type $h$. The value of $x$ is calculated as $(L - \sum_{i=h+1}^{M} d_i n_i)/n_h$, indicating the remaining capacity after satisfying the demands of indices greater than $h$, divided by the unit size $n_h$. $\sum_{i=h+1}^{M} d_i e_i$: This term accounts for the allocation of resources for group types $h+1$ to $M$. It represents the total demand for these group types, where $d_i$ denotes the demand of group type $i$, and $e_i$ represents the unit size of the corresponding element in $\mathbf{X}$. Together, the aggregate optimal solution combines the allocation of resources for group type $h$ with the aggregated demands for group types $h+1$ to $M$ to achieve an optimal solution to the linear relaxation of the problem.

Let the optimal solution to the relaxation of DEF be $x_{ij}^*$. Aggregate $\mathbf{x}^*$ to the number of each group type, $s_i^0 = \sum_j x_{ij}^*, i \in \mathbf{M}$. Replace the vector $\mathbf{d}$ with $\mathbf{s}^0$, we have the following problem,

$$
\{\max \sum_{j=1}^{N}\sum_{i=1}^{M}(n_i-\delta)x_{ij} : \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N}; \sum_{j=1}^{N} x_{ij} \leq s_i^0, i \in \mathcal{M}; x_{ij} \in Z^+\} \tag{17}
$$

then solve the resulting problem (17) to obtain the optimal solution, $\mathbf{x}^1$, which represents a feasible seat planning. Aggregate $\mathbf{x}^1$ to the number of each group type, $s_i^1 = \sum_j x_{ij}^1, i \in \mathbf{M}$, which represents the supply for each group type.

To fully utilize the seats, we should set the supply $\mathbf{s}^1$ as the lower bound, then re-solve a seat planning problem. We substitute the constraint $\sum_{j=1}^{N} x_{ij} \leq d_i, i \in \mathcal{M}$ in problem (16) with the new constraint $\sum_{j=1}^{N} x_{ij} \geq s_i^1, i \in \mathcal{M}$.

$$
\{\max \sum_{j=1}^{N}\sum_{i=1}^{M}(n_i-\delta)x_{ij} : \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N}; \sum_{j=1}^{N} x_{ij} \geq s_i^1, i \in \mathcal{M}; x_{ij} \in Z^+\} \tag{18}
$$

Notice that the number of unoccupied seats in the seat planning obtained from problem (18) is at most $\delta$ for each row, given any feasible supply, $\mathbf{s}^1$. To maximize the utilization of seats, we should assign full or largest patterns to each row. This procedure can be described in **Step 4** of the following algorithm.

---

**Algorithm 2** Feasible seat planning algorithm

---

**Step 1.** Obtain the solution, $\mathbf{x}^*$, from stochatic linear programming 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 (17) 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.** Obtain the optimal solution, $\mathbf{x}^2$, from problem (18) with supply $\mathbf{s}^1$. 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.** Check if row $j$ is full for all $j$. When row $j^0$ is not full, i.e., $\sum_i n_i x_{ij} < L_{j^0}$, let $\beta = L_{j^0} - \sum_i n_i x_{ij}$. Find the smallest group size in row $j^0$ and mark it as $i^0$. If the smallest group is exactly the largest, then the row corresponds to the largest pattern and check next row. Otherwise, reduce the number of group type $i^0$ by one and increase the number of group type $\min\{(i^0+\beta), M\}$ by one. Continue this procedure until this row is full.

---

**Remark 4.** **Step 2** *can give a feasible seat planning.* **Step 4** *can give the full or largest patterns for each row.*

**Remark 5.** *For a feasible seat planning, we provide a full or largest pattern for each row. The sequence of groups within each pattern can be arranged arbitrarily, allowing for a flexible seat planning that can accommodate realistic operational constraints. Therefore, any fixed sequence of groups within each pattern can be used to construct a seat planning that meets practical needs.*

# 5 Dynamic Seat Assignment(DSA) for Each Group Arrival

In this section, we discuss policies for assigning seats dynamically. We need to make decisions not only on whether to accept or reject an arrival but also on how to assign seats to each group if we decide to accept it. We first present the dynamic seat assignment based on stochastic planning policy, which incorporates group-type control to optimize resource utilization. We also compare this approach with the bid-price control policy. Finally, we establish the FCFS policy as the benchmark.

We can estimate the arrival rate from the historical data, $p_i = \frac{N_i}{N_0}, i \in \mathcal{M}$, where $N_0$ is the number of total groups, $N_i$ is the number of group type $i$. Recall that we assume there are $T$ independent periods, with one group arriving in each period. There are $M$ different group types. Let $\mathbf{y}$ be a discrete random variable indicating the number of people in the group, and let $\mathbf{p}$ be a vector probability, where $p(y = i) = p_i$, $i \in \mathcal{M}$ and $\sum_i p_i = 1$.

## 5.1 Seat Assignment Based on Stochastic Planning Policy

### 5.1.1 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.

In the following section, we will demonstrate how to decide whether to accept the current group to occupy larger-size seats when there is no corresponding supply available.

When the number of remaining periods is $T_r$, for any $j > i$, we can use one supply of group type $j$ to accept a group of $i$. In that case, when $i+\delta \leq j$, $(j-i-\delta)$ seats can be provided for one group of $j-i-\delta$ with $\delta$ seats of social distancing. Let $D_j$ be the random variable indicates the number of group type $j$ in $T_r$ periods. The expected number of accepted people is $i + (j-i-\delta)P(D_{j-i-\delta} \geq x_{j-i-\delta}+1; T_r)$, where $P(D_i \geq x_i; T_r)$ is the probability of that the demand of group type $i$ in $T_r$ periods is no less than $x_i$, the remaining supply of group type $i$. Thus, the term, $P(D_{j-i-\delta} \geq x_{j-i-\delta}+1; T_r)$, indicates the probability that the demand of group type $(j-i-\delta)$ in $T_r$ periods is no less than its current remaining supply plus 1. When $i < j < i+\delta$, the expected number of accepted people is $i$.

Similarly, when we retain the supply of group type $j$ by rejecting a group of $i$, the expected number of accepted people is $jP(D_j \geq x_j; T_r)$. $P(D_j \geq x_j; T_r)$ indicates the probability that the demand of group type $j$ in $T_r$ periods is no less than its current remaining supply.

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. If $j \geq i+\delta$, $d(i,j)$ equals $i + (j-i-\delta)P(D_{j-i-\delta} \geq x_{j-i-\delta} + 1) - jP(D_j \geq x_j)$, otherwise, $d(i,j)$ equals $i - jP(D_j \geq x_j)$. One intuitive decision is to choose the largest difference. For all $j > i$, 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.

**Remark 6.** *This control is based on the current feasible seat planning. We can decide which row to place according to the pattern instead of placing it when the capacity is sufficient. When a new seat planning can be regenerated, we can use the objective value of stochastic programming to make the decision.*

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. This approach takes into account the uncertain nature of the decision-making environment and allows for a more optimal decision to be made.

### 5.1.2 Algorithm Based on Stocahstic Planning Policy

The feasible seat planning can be obtained from Algorithm 2. In accordance with the group-type control discussed in the previous section, we determine whether to accept or reject group arrivals and which row to assign the group in.

The algorithm is shown below:

---

**Algorithm 3** Dynamic seat assignment algorithm

---

**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.

---

### 5.1.3 Break Tie for Stochastic Planning

A tie occurs when a small group is accepted by a larger planned group. To accept the smaller group, check if the current row contains at least two planned groups, including one larger group. If so, accept the smaller group in that row. If not, move on to the next row and repeat the check. If no available row is found after checking all rows, place the smaller group in the first row that contains the larger group.

By following this approach, the number of unused seats in each row can be reduced, leading to better capacity utilization.

## 5.2 Bid-price Control

Bid-price control is a classical approach discussed extensively in the literature on network revenue management. It involves setting bid prices for different group types, which determine the eligibility of groups to take the seats. Bid-prices refer to the opportunity costs of taking one seat. As usual, we estimate the bid price of a seat by the shadow price of the capacity constraint corresponding to some row. In this section, we will demonstrate the implementation of the bid-price control policy.

The dual problem of linear relaxation of problem (16) is:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{M} d_i z_i + \sum_{j=1}^{N} L_j \beta_j \\
\text{s.t.} \quad & z_i + \beta_j n_i \geq (n_i - \delta), \quad i \in \mathcal{M}, j \in \mathcal{N} \\
& z_i \geq 0, i \in \mathcal{M}, \beta_j \geq 0, j \in \mathcal{N}.
\end{aligned} \tag{19}
$$

In (19), $\beta_j$ can be interpreted as the bid-price for a seat in row $j$. A request is only accepted if the revenue it generates is above the sum of the bid prices of the seats it uses. Thus, if its revenue is more than its opportunity costs, i.e., $i - \beta_j n_i > 0$, we will accept the group type $i$. And choose $j^* = \arg\max_j \{i - \beta_j n_i\}$ as the row to allocate that group.

**Lemma 4.** *According to Lemma 3, there exists $h$ such that the aggregate optimal solution to relaxation of problem (16) takes the form $xe_h + \sum_{i=h+1}^{M} d_i e_i$, $x = (L - \sum_{i=h+1}^{M} d_i n_i)/n_h$. Then the optimal solution to probelm (19) is given by $z_1, \ldots, z_h = 0$, $z_i = \frac{\delta(n_i - n_h)}{n_h}$ for $i = h+1, \ldots, M$ and $\beta_j = \frac{n_h - \delta}{n_h}$ for all $j$.*

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. It coincides with the booking limit control policy. However, we can find that $\beta_j$ does not vary with $j$, which means bid-price control cannot decide which row to assign the group to.

The bid-price control policy based on the static model is stated below.

**Algorithm 4** Bid-price control algorithm

**Step 1.** Observe the arrival group type $i$ at period $t = 1, \ldots, T$.

**Step 2.** Solve the linear relaxation of problem (16) with $d_i^t = (T - t) \cdot p_i$ and $\mathbf{L}^t$, obtain the aggregate optimal solution $xe_h + \sum_{i=h+1}^{M} d_i e_i$.

**Step 3.** If $i \geq h$, accept the arrival and assign the group to row $k$ arbitrarily, update $\mathbf{L}^{t+1} = \mathbf{L}^t - n_i \mathbf{e}_k^\top$; otherwise, reject it, let $\mathbf{L}^{t+1} = \mathbf{L}^t$.

**Step 4.** If $t \leq T$, move to next period, set $t = t + 1$, go to step 2. Otherwise, terminate this algorithm.

## 5.3 Booking limit Control

The booking limit control policy involves setting a maximum number of reservations that can be accepted for each group type. By controlling the booking limits, revenue managers can effectively manage demand and allocate inventory to maximize revenue.

In this policy, we replace the real demand by the expected one and solve the corresponding static problem 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 (16), 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.

When we solve problem (16) directly, we can develop the booking limit control policy.

**Algorithm 5** Booking limit control algorithm

**Step 1.** Observe the arrival group type $i$ at period $t = 1, \dots, T$.

**Step 2.** Solve problem (16) with $d_i^t = (T - t) \cdot p_i$ and $\mathbf{L}^t$, obtain the optimal solution, $x_{ij}^*$ and the aggregate optimal solution, $\mathbf{X}$.

**Step 3.** If $X_i > 0$, accept the arrival and assign the group to row $k$ where $x_{ik} > 0$, update $\mathbf{L}^{t+1} = \mathbf{L}^t - n_i \mathbf{e}_k^\top$; otherwise, reject it, let $\mathbf{L}^{t+1} = \mathbf{L}^t$.

**Step 4.** If $t \leq T$, move to next period, set $t = t + 1$, go to step 2. Otherwise, terminate this algorithm.

## 5.4 Dynamic Programming Base-heuristic

Since the original dynamic programming problem is too complex to solve directly, we can instead consider a simplified version of the problem, known as the relaxation problem. By solving the relaxation problem, we can make decisions for each group arrival based on the dynamic programming approach.

Relax all rows to one row with the same capacity by $L = \sum_{j=1}^{N} L_j$. The deterministic problem is

$$\begin{aligned} \max \quad & \sum_{i=1}^{M} (n_i - \delta) x_i \\ \text{s.t.} \quad & x_i \leq d_i, \quad i \in \mathcal{M}, \\ & \sum_{i=1}^{M} n_i x_i \leq L \\ & x_i \in \mathbb{Z}_+, \quad i \in \mathcal{M}. \end{aligned} \tag{20}$$

**Lemma 5.** *Let* $\mathbf{X}$ *be the solution of linear relaxation of problem* (20). $\mathbf{X}$ *is the same as the aggregate solution of linear relaxation of problem* (16).

Let $u$ denote the decision, where $u(t) = 1$ if we accept a request in period $t$, $u(t) = 0$ otherwise. Similar to the DP in section 3.2, the DP with one row can be expressed as:

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

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

## 5.5 First Come First Served(FCFS) Policy

For dynamic seat assignment for each group arrival, the intuitive but trivial method will be on a first-come-first-served basis. Each accepted request will be assigned seats row by row. If the capacity of a row is insufficient to accommodate a request, we will allocate it to the next available row. If a subsequent request can fit exactly into the remaining capacity of a partially filled row, we will assign it to that row immediately. Then continue to process requests in this manner until the capacity of all rows is fully utilized.

## 6 Results

We carried out several experiments, including comparing the running time of decomposition and Integer programming, comparing the number of people served using the feasible seat planning and Integer programming methods, analyzing different policies under the two booking situations, evaluating the results under varying demands, assessing the results for different numbers of people in each period and finally investigating the impact of seat layout on the number of served people.

### 6.1 Running time of Benders Decomposition and IP

The running times of solving DEF directly and solving the relaxation of DEF with Benders decomposition are shown in Table 1.

Table 1: Running time of Decomposion 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 | |

The parameters in the columns of the table are the number of scenarios, the range of demands, running time of integer programming, running time of Benders decomposition method, the number of rows, the number of group types and the number of seats for each row, respectively.

Take the first experiment as an example, the scenarios of demands are generated from (150, 350) randomly, the number of seats for each row is generated from (21, 50) randomly.

### 6.2 Feasible Seat Planning versus IP Solution

An arrival sequence can be expressed as $\{y_1, y_2, \ldots, y_T\}$. Let $N_i = \sum_t I(y_t = i)$, i.e., the number of times group type $i$ arrives during $T$ periods. Then the scenarios, $(N_1, \ldots, N_M)$, follow a multinomial distribution,

$$p(N_1, \ldots, N_M \mid \mathbf{p}) = \frac{T!}{N_1!, \ldots, N_M!} \prod_{i=1}^{M} p_i^{N_i}, T = \sum_{i=1}^{M} N_i.$$

It is clear that the number of different sequences is $M^T$. The number of different scenarios is $O(T^{M-1})$ which can be obtained by the following DP.

Use $D(T, M)$ to denote the number of scenarios, which equals the number of different solutions to $x_1 + \ldots + x_M = T, \mathbf{x} \geq 0$. Then, we know the recurrence relation $D(T, M) = \sum_{i=0}^{T} D(i, M-1)$ and boundary condition, $D(i, 1) = 1$. So we have $D(T, 2) = T + 1$, $D(T, 3) = \frac{(T+2)(T+1)}{2}$, $D(T, M) =$

$O(T^{M-1})$. The number of scenarios is too large to enumerate all possible cases. Thus, we choose to sample some sequences from the multinomial distribution.

Then, we will show the feasible seat assignment has a close performance with IP when considering group-type control policy.

Table 2: Feasible seat planning versus IP solution

| # samples | T | probabilities | # rows | people served by decomposition | people served by 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. The number of seats is 20 when the number of rows is 8, the number of seats is 40 when the number of rows is 30.

## 6.3 Performances of Different Policies to Optimal

In this section, we compare the performance of four dynamic seat assignment policies to the optimal value, which can be obtained by solving the deterministic model after observing all arrivals. The policies under examination are the stochastic planning policy, DP Base-heuristic, bid-price policy and FCFS policy. The seat layout consists of 10 rows, each with 21 seats (including one dummy seat), and the group size can range up to 4 people. We conducted experiments over 60 to 100 periods to demonstrate the policies' performance under varying demand levels. We selected three probabilities to ensure that the expected number of people for each period is consistent. The table below displays the average of 200 instances for each number.

We can find that the stochastic planning policy are better than DP Base-heuristic and bid-price policy consistently, and FCFS policy works worst. As we mentioned previously, DP Base-heuristic and bid-price policy can only make the decision to accept or deny, cannot decide which row to assign the group to. FCFS accepts groups in sequential order until the capacity cannot accommodate more.

For the first three policies, their performance tends to initially drop and then increase as the number of periods increases. When the number of periods is small, the demand for capacity is relatively low, and the policies can achieve relatively optimal performance. However, as the number of periods increases, the policies may struggle to always obtain a perfect allocation plan, leading to a decrease in performance. Nevertheless, when the number of periods continue to become larger, these policies tend to accept larger

Table 3: Performances of Different Policies to Optimal

| 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 |
| 90 | [0.25, 0.25, 0.25, 0.25] | 99.10 | 96.05 | 96.41 | 92.48 |
| 100 | [0.25, 0.25, 0.25, 0.25] | 99.58 | 95.09 | 96.88 | 92.54 |
| 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 |
| 90 | [0.25, 0.35, 0.05, 0.35] | 99.01 | 96.77 | 96.62 | 92.46 |
| 100 | [0.25, 0.35, 0.05, 0.35] | 99.23 | 97.04 | 97.14 | 92.00 |
| 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 |
| 90 | [0.15, 0.25, 0.55, 0.05] | 99.53 | 98.45 | 98.69 | 95.50 |
| 100 | [0.15, 0.25, 0.55, 0.05] | 99.47 | 98.62 | 98.94 | 95.25 |

groups, and as a result, narrow the gap with the optimal value, leading to an increase in performance.

### 6.4 Impact of Social Distance as Demands Increase

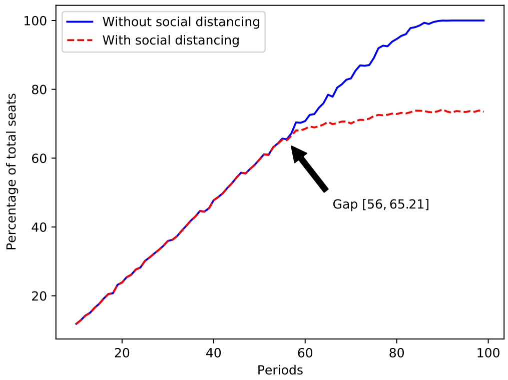
In this section, we will explore how social distancing affects the occupancy rate of a venue under different demands. Let $\gamma$ represent the average number of people who arrive in each period, and let $L$ represent the total number of seats available. We can calculate the expected number of people arriving using the formula $E(D) = \gamma T$, where $T$ is the number of periods.

If we assume that we accept all incoming groups within the $T'$ periods and they fill all the available seats, then we have $E(D) + T' = L$. Solving for $T'$, we get $T' = L/(\gamma + 1)$. $T'$ represents the ideal 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. But the actual first period will be smaller than the ideal one because we cannot always exactly fill the seats with accepted groups.

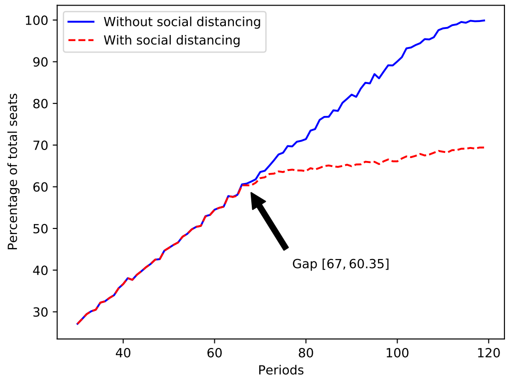
If we limit the number of group types to 4, we can express $\gamma$ as $\gamma = p_1 * 1 + p_2 * 2 + p_3 * 3 + p_4 * 4$, where $p_1$, $p_2$, $p_3$, and $p_4$ represent the probabilities of groups with one, two, three, and four people, respectively. Define each combination $(p_1, p_2, p_3, p_4)$ such that $p_1 + p_2 + p_3 + p_4 = 1$ as a probability combination. Specifically, we consider two situations: $\gamma = 2.5$ and $\gamma = 1.9$. When $\gamma = 2.5$, we set the parameters as follows: $T$ varies from 10 to 100, the step size is 1. When $\gamma = 1.9$, we set the parameters as follows: $T$ varies from 30 to 120, the step size is 1. The seat layout consists 10 rows and the number of seats per row is 21.

The figure below displays the outcomes of groups who were accepted under two different conditions: with social distancing measures and without social distancing measures. For the former case, we employ stochastic planning to obtain the results. For the latter case, we utilize a deterministic model with hindsight to generate the outcomes. Since the various probabilities with the same $\gamma$ exhibit similar patterns as shown in the figure, we present only one case of probabilities to illustrate the detailed figure.

The analysis comprises three stages. In the first stage, where the capacity is sufficient, social




(a) When $\gamma = 2.5$




(b) When $\gamma = 1.9$

Figure 2: The number of people served versus periods

distancing measures have no impact on the outcome. In the second stage, the gap between the outcomes with and without social distancing measures widens as $T$ increases. Finally, in the third stage, as $T$ continues to increase, the gap between the outcomes with and without social distancing measures converges when both situations accept the maximum number of people.

## 6.5 Comparison of Different Probabilities When Supply and Demand Are Close

When we set the number of periods $T$ to be equal to $T'$, representing a state where the expected demand and supply are in balance, we can observe differences among the outcomes for different probabilities. In this case, the estimated occupancy rate is given by $\frac{\gamma T'}{(\gamma+1)T'-N}$. It is important to note that this estimation is accurate only when all rows represent full patterns, as per our assumption.

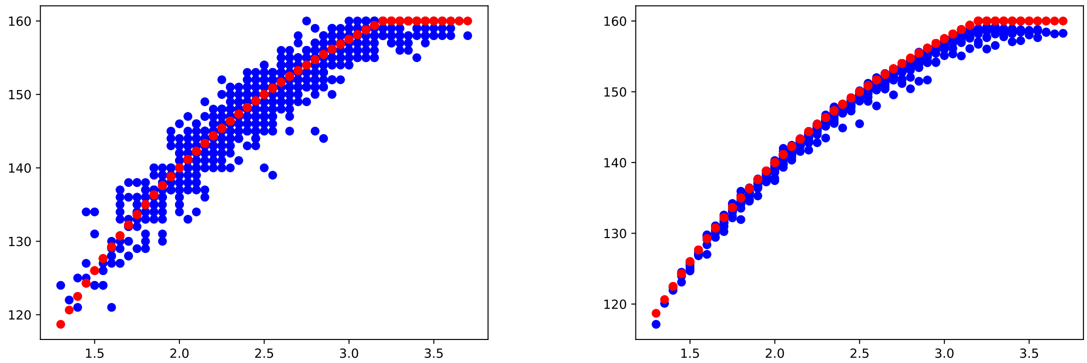
We sample $p_1$, $p_2$, and $p_3$ from 0.05 to 0.95 with an increment of 0.05. The seat layout still remains the same as the above experiments. The figure below shows the number of people served for each value of $\gamma$. For each probability combination, the blue point represents the average number of people served over 50 instances, and the red point represents the estimated number of people served.

If the largest pattern is assigned to each row, the resulting occupancy rate is $\frac{16}{21}$. The maximum number of people that can be accepted is $200 * \frac{16}{20} = 160$, which is the upper bound on the number of people that can be accepted. The estimated number of people accepted is given by $\frac{\gamma}{\gamma+1} * 200$, as indicated by the red points in the figure.

### 6.5.1 Analysis on The Difference Between Blue and Red Points

We can give the absolute difference between the blue point and red point for each probability combination as below.

Table 4 displays the absolute difference proportion for the average of 20, 50, and 100 instances, while Figure 4 shows the difference distribution for the average of 50 and 100 instances. These results suggest that we can estimate the attendance rate based on $\gamma$ for most probability combinations.




(a) One instance for each probability combination

(b) Average of 50 instances for each probability combination

Figure 3: The number of people accepted versus $\gamma$

Table 4: Absolute Difference Proportion

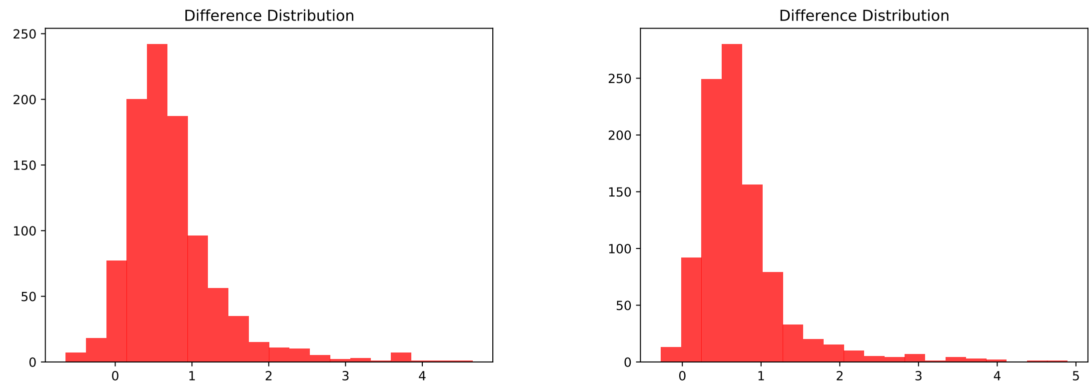
| # of instances | abs_diff $\geq 1$ | abs_diff $\geq 2$ | abs_diff $\geq 3$ | abs_diff $\geq 4$ |
|---|---|---|---|---|
| 20 | 32.92 % | 5.13 % | 1.74% | 0.51 % |
| 50 | 22.46 % | 4.31 % | 1.54 % | 0.31 % |
| 100 | 20.00 % | 4.21 % | 1.54 % | 0.31 % |

However, we have also observed that some blue points in the figure are significantly distant from their corresponding red points. This is evident when the probability combination is $[0.05, 0.05, 0.85, 0.05]$ (which corresponds to $\gamma = 2.9$), and the demands cannot be accommodated by constructing full patterns for every row, which does not satisfy our assumption. This leads to a considerable gap between the blue and red points in this case. For example, the demands could be $[4, 1, 45, 2]$ or $[2, 2, 47, 1]$ according to the probability combination, but the typical pattern that can be generated is $(4, 4, 4, 4)$, which is not full.

### 6.5.2 Results of Different Seat Layouts

If we modify the even seat layout, the differences between the red and blue points will decrease, and some outliers may be eliminated. To maintain the same total seating capacity, we compare two layouts with the even seat layout. The step-size seat layout consists of rows with 17, 18, 19, 20, 21, 21, 22, 23, 24, and 25 seats, while the random seat layout has rows with 19, 20, 21, 21, 23, 24, 26, 17, 19, and 20 seats. Both layouts can accommodate a maximum of 164 people when each row corresponds to the largest pattern.

The results suggest that a random seat layout may provide a more accurate estimation, implying that such a layout is more capable of accommodating uncertain demands and achieving a full pattern that can accept more people.




(a) Average of 50 instances for each probability combination (b) Average of 100 instances for each probability combination

Figure 4: The difference distribution

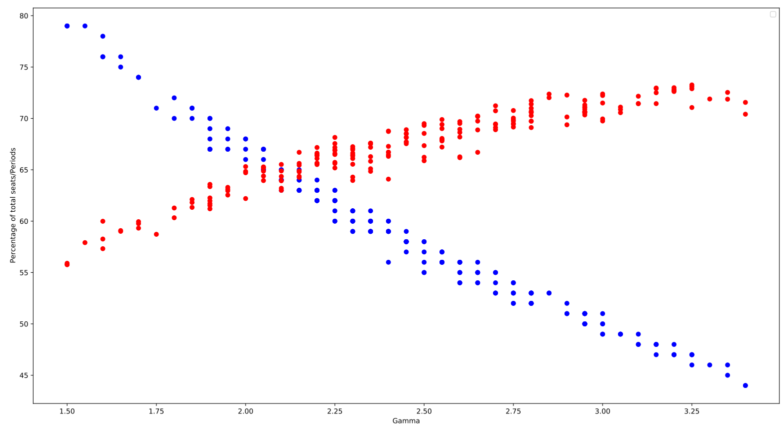
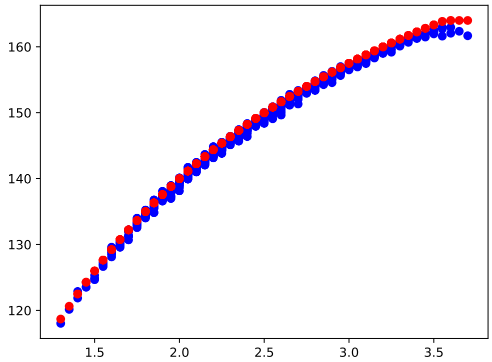


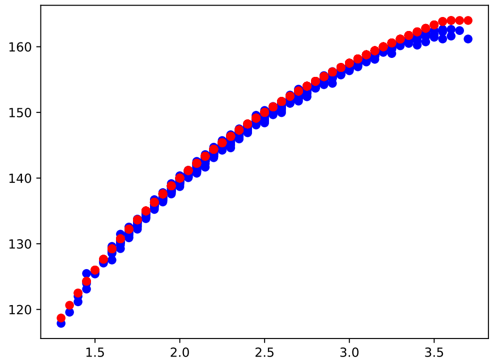

Figure 5: Gap points under 200 probabilities

# 7 Conclusion

In conclusion, this paper addresses the problem of dynamic seat assignment with social distancing in the context of a pandemic. We propose a practical algorithm that balances seat utilization rates and the associated risk of infection to obtain a final seating plan that satisfies social distancing constraints when groups arrive. Our approach provides a comprehensive solution for optimizing seat assignments while ensuring the safety of customers. Our contributions include establishing a deterministic model to analyze the effects of social distancing when demand is known, using two-stage stochastic programming and Benders decomposition methods to obtain the optimal linear solution for stochastic demand situations, and developing a feasible seating plan using scenario-based stochastic programming for the dynamic scenario problem. Our results demonstrate significant improvements over baseline strategies and provide guidance for developing attendance policies. Overall, our study highlights the importance of considering the operational significance behind social distancing and provides a new perspective for the government to adopt mechanisms for setting seat assignments to protect people in the post-pandemic era. Our study




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




(b) Average of 50 instances for random seat layout

Figure 6: The number of people served versus $\gamma$

demonstrates the efficiency of obtaining the final seating plan using our proposed algorithm. The results indicate that our policy yields a seating plan that is very close to the optimal result. Moreover, our analysis provides managerial guidance on how to set the occupancy rate and largest size of one group under the background of pandemic.

# References


[1] Michael Barry, Claudio Gambella, Fabio Lorenzi, John Sheehan, and Joern Ploennigs. Optimal seat allocation under social distancing constraints. *arXiv preprint arXiv:2105.05017*, 2021.

[2] Matthew E Berge and Craig A Hopperstad. Demand driven dispatch: A method for dynamic aircraft capacity assignment, models and algorithms. *Operations research*, 41(1):153–168, 1993.

[3] Danny Blom, Rudi Pendavingh, and Frits Spieksma. Filling a theater during the covid-19 pandemic. *INFORMS Journal on Applied Analytics*, 52(6):473–484, 2022.

[4] Juliano Cavalcante Bortolete, Luis Felipe Bueno, Renan Butkeraites, Antônio Augusto Chaves, Gustavo Collaço, Marcos Magueta, FJR Pelogia, LL Salles Neto, TS Santos, TS Silva, et al. A support tool for planning classrooms considering social distancing between students. *Computational and Applied Mathematics*, 41:1–23, 2022.

[5] Michael S Casey and Suvrajeet Sen. The scenario generation algorithm for multistage stochastic linear programming. *Mathematics of Operations Research*, 30(3):615–631, 2005.

[6] Tommy Clausen, Allan Nordlunde Hjorth, Morten Nielsen, and David Pisinger. The off-line group seat reservation problem. *European journal of operational research*, 207(3):1244–1253, 2010.

[7] Renwick E Curry. Optimal airline seat allocation with fare classes nested by origins and destinations. *Transportation science*, 24(3):193–204, 1990.

[8] George B Dantzig. Discrete-variable extremum problems. *Operations research*, 5(2):266–288, 1957.

[9] Igor Deplano, Danial Yazdani, and Trung Thanh Nguyen. The offline group seat reservation knapsack problem with profit on seats. *IEEE Access*, 7:152358–152367, 2019.

[10] Yonghan Feng and Sarah M Ryan. Scenario construction and reduction applied to stochastic power generation expansion planning. *Computers & Operations Research*, 40(1):9–23, 2013.

[11] Martina Fischetti, Matteo Fischetti, and Jakob Stoustrup. Safe distancing in the time of covid-19. *European Journal of Operational Research*, 2021.

[12] Guillermo Gallego and Garrett Van Ryzin. A multiproduct dynamic pricing problem and its applications to network yield management. *Operations research*, 45(1):24–41, 1997.

[13] Elaheh Ghorbani, Hamid Molavian, and Fred Barez. A model for optimizing the health and economic impacts of covid-19 under social distancing measures; a study for the number of passengers and their seating arrangements in aircrafts. *arXiv preprint arXiv:2010.10993*, 2020.

[14] GovHK. Government relaxes certain social distancing measures. https://www.info.gov.hk/gia/general/202209/30/P2022093000818.htm, 2022.

[15] Younes Hamdouch, HW Ho, Agachai Sumalee, and Guodong Wang. Schedule-based transit assignment model with vehicle capacity and seat availability. *Transportation Research Part B: Methodological*, 45(10):1805–1830, 2011.

[16] Md Tabish Haque and Faiz Hamid. An optimization model to assign seats in long distance trains to minimize sars-cov-2 diffusion. *Transportation Research Part A: Policy and Practice*, 162:104–120, 2022.

[17] Md Tabish Haque and Faiz Hamid. Social distancing and revenue management—a post-pandemic adaptation for railways. *Omega*, 114:102737, 2023.

[18] Healthcare. Covid-19 timeline. https://www.otandp.com/covid-19-timeline, 2023.

[19] Réne Henrion and Werner Römisch. Problem-based optimal scenario generation and reduction in stochastic programming. *Mathematical Programming*, pages 1–23, 2018.

[20] Anton J Kleywegt and Jason D Papastavrou. The dynamic and stochastic knapsack problem. *Operations research*, 46(1):17–35, 1998.

[21] Sungil Kwag, Woo Jin Lee, and Young Dae Ko. Optimal seat allocation strategy for e-sports gaming center. *International Transactions in Operational Research*, 29(2):783–804, 2022.

[22] Rhyd Lewis and Fiona Carroll. Creating seating plans: a practical application. *Journal of the Operational Research Society*, 67(11):1353–1362, 2016.

[23] Yihua Li, Bruce Wang, and Luz A Caudillo-Fuentes. Modeling a hotel room assignment problem. *Journal of Revenue and Pricing Management*, 12:120–127, 2013.

[24] Jane F Moore, Arthur Carvalho, Gerard A Davis, Yousif Abulhassan, and Fadel M Megahed. Seat assignments with physical distancing in single-destination public transit settings. *Ieee Access*, 9:42985–42993, 2021.

[25] Imad A Moosa. The effectiveness of social distancing in containing covid-19. *Applied Economics*, 52(58):6292–6305, 2020.

[26] David Pisinger. An exact algorithm for large multiple knapsack problems. *European Journal of Operational Research*, 114(3):528–541, 1999.

[27] Mostafa Salari, R John Milne, Camelia Delcea, and Liviu-Adrian Cotfas. Social distancing in airplane seat assignments for passenger groups. *Transportmetrica B: Transport Dynamics*, 10(1):1070–1098, 2022.

[28] Mostafa Salari, R John Milne, Camelia Delcea, Lina Kattan, and Liviu-Adrian Cotfas. Social distancing in airplane seat assignments. *Journal of Air Transport Management*, 89:101915, 2020.

[29] Garrett Van Ryzin and Gustavo Vulcano. Simulation-based optimization of virtual nesting controls for network revenue management. *Operations research*, 56(4):865–880, 2008.

[30] Elizabeth Louise Williamson. *Airline network seat inventory control: Methodologies and revenue impacts*. PhD thesis, Massachusetts Institute of Technology, 1992.

[31] Benson B Yuen. Group revenue management: Redefining the business process—part i. *Journal of Revenue and Pricing Management*, 1:267–274, 2002.

[32] Feng Zhu, Shaoxuan Liu, Rowan Wang, and Zizhuo Wang. Assign-to-seat: Dynamic capacity control for selling high-speed train tickets. *Manufacturing & Service Operations Management*, 2023.

# Proof

(Proof of Lemma 3). *Treat the groups as the items, the rows as the knapsacks. There are $M$ types of items, the total number of which is $K = \sum_i d_i$, each item $k$ has a profit $p_k$ and weight $w_k$.*

*Then this Integer Programming is a special case of the Multiple Knapsack Problem(MKP). Consider the solution to the linear relaxation of* (16). *Sort these items according to profit-to-weight ratios $\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \ldots \geq \frac{p_K}{w_K}$. Let the break item $b$ be given by $b = \min\{j : \sum_{k=1}^{j} w_k \geq L\}$, where $L = \sum_{j=1}^{N} L_j$ is the total size of all knapsacks. Then the Dantzig upper bound [8] becomes $u_{\mathrm{MKP}} = \sum_{j=1}^{b-1} p_j + \left(L - \sum_{j=1}^{b-1} w_j\right) \frac{p_b}{w_b}$. The corresponding optimal solution is to accept the whole items from 1 to $b-1$ and fractional $(L - \sum_{j=1}^{b-1} w_j)$ item $b$. Suppose the item $b$ belong to type $h$, then for $i < h$, $x_{ij}^* = 0$; for $i > h$, $x_{ij}^* = d_i$; for $i = h$, $\sum_j x_{ij}^* = (L - \sum_{i=h+1}^{M} d_i n_i)/n_h$.* □

(Proof of Lemma 4). *According to the Lemma 3, the aggregate optimal solution to relaxation of problem* (16) *takes the form $xe_h + \sum_{i=h+1}^{M} d_i e_i$, then according to the complementary slackness property, we know that $z_1, \ldots, z_h = 0$. This implies that $\beta_j \geq \frac{n_i - \delta}{n_i}$ for $i = 1, \ldots, h$. Since $\frac{n_i - \delta}{n_i}$ increases with $i$, we have $\beta_j \geq \frac{n_h - \delta}{n_h}$. Consequently, we obtain $z_i \geq n_i - \delta - n_i \frac{n_h - \delta}{n_h} = \frac{\delta(n_i - n_h)}{n_h}$ for $i = h+1, \ldots, M$.*

*Given that $d$ and $L$ are both no less than zero, the minimum value will be attained when $\beta_j = \frac{n_h - \delta}{n_h}$ for all $j$, and $z_i = \frac{\delta(n_i - n_h)}{n_h}$ for $i = h+1, \ldots, M$. Then the bid-price decision $i - \beta_j n_i = i - \frac{n_h - \delta}{n_h} n_i = \frac{\delta(i-h)}{n_h}$. When $i < h$, $i - \beta_j n_i < 0$. When $i \geq h$, $i - \beta_j n_i \geq 0$.* □