Seating Management under Social Distancing

April 1, 2025

Abstract

We consider the critical challenge of seat planning and assignment in the context of social distancing measures, which have become increasingly vital in today's environment. Ensuring that individuals maintain the required social distances while optimizing seating management requires careful consideration of several factors, including group sizes, venue layouts, and fluctuating demand patterns. Initially, we analyze seat planning with deterministic requests. Subsequently, we introduce a scenario-based stochastic programming approach to address seat planning with stochastic requests. The seat planning can serve as the foundation for the seat assignment. Furthermore, we explore a dynamic situation where groups arrive sequentially. Combining with relaxed dynamic programming, we propose a seat-plan-based assignment policy for either accommodating or rejecting incoming groups. Our method outperforms traditional bid-price and booking-limit strategies. The findings furnish valuable insights for policymakers and venue managers regarding seat occupancy rates and provide a practical framework for implementing social distancing protocols while optimizing seat allocations.

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

1 Introduction

Social distancing has proven effective in containing the spread of infectious diseases. For instance, during the recent COVID-19 pandemic, the fundamental requirement of social distancing involved establishing a minimum physical distance between individuals in public spaces. However, the principles of social distancing can also be applied to various industries beyond health prevention. For example, restaurants may adopt social distancing practices to enhance guest experience and satisfaction while fostering a sense of privacy. In event management, particularly for large gatherings, social distancing can improve comfort and safety, even in non-health-related contexts, by providing attendees with more personal space.

As a general principle, social distancing measures can be defined from different dimensions. The basic requirement of social distancing is the specification of a minimum physical distance between individuals in public areas. For example, the World Health Organization (WHO) suggests to"keep physical distance of at least 1 meter from others" WHO (2020). In the US, the Centers for Disease Control and Prevention (CDC) describes social distancing as "keeping a safe space between yourself and other people who are not from your household" CDC (2020). It's important to note that this requirement is typically applied with respect to groups of people. For instance, in Hong Kong, the government has implemented social distancing measures during the Covid-19 pandemic by limiting the size of groups in public gatherings to two, four, and six people per group over time. Moreover, the Hong Kong government has also established an upper limit on the total number of people in a venue; for example, restaurants were allowed to operate at 50% or 75% of their normal seating capacity.

From a company's perspective, social distancing can disrupt normal operations in certain sectors. For example, a restaurant needs to change or redesign the layout of its tables to comply with social distancing requirements. Such change often results in reduced capacity, fewer customers, and consequently, less revenue. In this context, affected firms face the challenge of optimizing its operational flow when adhering to social distancing policies. From a government perspective, the impact of enforcing social distancing measures on economic activities is a critical consideration in decision-making. Facing an outbreak of an infectious disease, a government must implement social distancing policy based on a holistic analysis. This analysis should take into account not only the severity of the outbreak but also the potential impact on all stakeholders. What is particularly important is the evaluation of business losses suffered by the industries that are directly affected.

We will address the above issues of social distancing in the context of seating management. Consider a venue, such as a cinema or a conference hall, which is used to host an event. The venue is equipped with seats of multiple rows. During the event, requests for seats arrive in groups, each containing a limited number of individuals. Each group can be either accepted or rejected, and those that are accepted will be seated consecutively in one row. Each row can accommodate multiple groups as long as any two adjacent groups in the same row are separated by one or multiple empty seats to comply with social distancing requirements. The objective is to maximize the number of individuals accepted for seating.

Seat management is critically dependent on demand patterns. We will consider three distinct problems related to seat management: the Seat Planning with Deterministic Requests (SPDR) problem, the Seat Planning with Stochastic Requests (SPSR) problem, and the Seat Assignment with Dynamic Requests (SADR) problem. In the first problem, SPDR problem, complete information about seating requests in groups is known. This applies to scenarios where the participants and their groups are identified, such as family members attending a church gathering or staff from the same office at a company meeting. In the second problem, SPSR problem, the requests are unknown but follow a probabilistic distribution. This problem is relevant in situations where a new seating layout must accommodate multiple events with varying seating requests. For example, during the COVID-19 outbreak, some theaters physically removed some seats and used the remaining ones to create a seating plan that accommodates stochastic requests. In the third problem, SADR problem, groups of seating requests arrive dynamically. The problem is to decide, upon the arrival of each group of request, whether to accept or reject the group, and assign seats for each accepted group. Such seat assignment is applicable in commercial settings where requests arrive as a stochastic process, such as ticket sales in movie theaters.

We develop models and derive optimal solutions for each of these three problems. Specifically, we formulate SPDR problem using Integer Programming and discuss the key characteristics of the optimal seating plan. For SPSR problem, we utilize scenario-based optimization and develop solution approaches based on Benders decomposition. Regarding SADR problem, we implement a two-stage seat-plan-based assignment approach. In the initial decision phase, a relaxed dynamic programming evaluates each incoming request to determine acceptance. The accepted requests then proceed to the assignment phase, where the group-type control allocation is performed. This seat-plan-based assignment policy outperforms traditional bid-price and booking-limit policies. Although each of these models represents a standalone problem tailored to specific situations, they are closely interconnected in terms of problem-solving methods and managerial insights. In the seat planning with deterministic requests (SPDR) problem, we identify important concepts such as the full pattern and the largest pattern, which play a crucial role in developing solutions for the other two problems. Additionally, SPDR problem serves as a useful offline benchmark for evaluating the performance of policies in SADR problem. Furthermore, the solution to SPSR problem can serve as a reference seat plan for dynamic seat assignment in SADR problem.

We investigate the impact of social distancing on revenue loss. To facilitate this analysis, we introduce the concept of the threshold of request-volume, which represents the upper limit on the number of requests an event can accommodate without being affected by social distancing measures. Specifically, if an event receives fewer requests than the threshold of request-volume, it will experience virtually no revenue loss due to social distancing. Our computational experiments demonstrate that the threshold of request-volume primarily depends on the mean of group size and is relatively insensitive to the specific distribution of group sizes. This finding provides a straightforward method for estimating the threshold of request-volume and evaluating the impact of social distancing. In some instances, the government imposes a maximum allowable occupancy rate to enforce stricter social distancing requirements. To assess this effect, we introduce the concept of the threshold of occupancy rate, defined as the occupancy rate at the threshold of request-volume. The maximum allowable occupancy rate is effective for an event only if it is lower than the event's threshold of occupancy rate. Moreover, it becomes redundant if it exceeds the maximum achievable occupancy rate for all events.

These qualitative insights are stable with respect to the government policy's strictness and the specific characteristics of various venues, such as minimum physical distance, allowable largest group size, and venue layout. When the minimum physical distance increases, both the threshold of occupation rate and maximum achievable occupation rate decrease accordingly. Conversely, when the allowable largest group size decreases, the number of accepted requests may increase; however, both the threshold of occupation rate and maximum achievable occupation rate decline. Although venue layouts may vary in shapes (rectangular or otherwise) and row lengths (long or short), the threshold of occupancy rate and maximum achievable occupancy rate do not exhibit significant variation.

The rest of this paper is structured as follows. We review the relevant literature in Section 2. Then we introduce the major issues brought by social distancing and define the seating planning with deterministic requests in Section 3. In Section 4, we establish the stochastic model, analyze its properties and obtain the seat planning. Section 5 demonstrates the seat-plan-based assignment policy to assign seats for incoming requests. Section 6 gives the numerical results and insights of implementing social distancing. Conclusions are shown in Section 7.

2 Literature Review

Seating management is a practical problem that presents unique challenges in various applications, each with its own complexities, particularly when accommodating group-based seating requests. For instance, in passenger rail services, groups differ not only in size but also in their departure and arrival destinations, requiring them to be assigned consecutive seats (Clausen et al., 2010; Deplano et al., 2019). In social gatherings such as weddings or dinner galas, individuals often prefer to sit together at the same table while maintaining distance from other groups they may dislike (Lewis and Carroll, 2016). In parliamentary seating assignments, members of the same party are typically grouped in clusters to facilitate intra-party communication as much as possible (Vangerven et al., 2022). In e-sports gaming centers, customers arrive to play games in groups and require seating arrangements that allow them to sit together (Kwag et al., 2022).

Incorporating social distancing into seating management has introduced an additional layer of complexity, sparking a new stream of research. Some works focus on the layout design and determine seating positions to maximize physical distance between individuals, such as students in classrooms (Bortolete et al., 2022) or customers in restaurants and beach umbrella arrangements (Fischetti et al., 2023). Other works assume the seating layout is fixed, and assign seats to individuals while adhering to social distancing guidelines. For example, Bortolete et al. (2022) also consider the fixed seat setting for students in the classroom, Salari et al. (2020); Pavlik et al. (2021) consider the seat assignment in the airplanes. These studies consider the seating management with social distancing for the individual requests.

Our work relates to seating management with social distancing for group-based requests, which has found its applications in various areas, including single-destination public transits (Moore et al., 2021), airplanes (Ghorbani et al., 2020; Salari et al., 2022), trains (Haque and Hamid, 2022, 2023), and theaters (Blom et al., 2022). Due to the diversity of applications, there are different issues to handle. For example, Salari et al. (2022) take the distance between different groups into account, leading to the development of a seating assignment strategy that outperforms the simplistic airline policy of blocking all middle seats. In Haque and Hamid (2023), when designing seat allocation for groups with social distancing, was not only the transmission risk inside the train considered, but also the transmission risk between different cities where the stops were located. Blom et al. (2022) address group-based seating problem in theaters. While they primarily focus on scenarios with known groups - referred to as seat planning with deterministic requests in our work - we consider a broader range of demand patterns. Specifically, we also examine group-based seat planning with stochastic requests. We also consider dynamic seat assignment, assuming that groups arrive sequentially according to a stochastic process.

Technically speaking, when all requests are known, SPDRP belongs to the category of the multiple knapsack problem (MKP) (Martello and Toth, 1990). Most existing work focus on deriving bounds or competitive ratios for algorithms designed to solve the general multiple knapsack problem (Khuri et al., 1994; Ferreira et al., 1996; Pisinger, 1999; Chekuri and Khanna, 2005). However, in our problem, the sizes of item weights, profits and knapsack capacities are all integers. Additionally, there are many identical groups of the same size, making the aggregation form particularly useful for determining the seat plan. Our work emphasizes the structure and properties of the solution to this specific problem, offering valuable insights for subsequent research in the dynamic situation.

The SADR problem falls under the category of the dynamic multiple knapsack problem. A related problem is the dynamic stochastic knapsack problem, which has been extensively studied in the literature (Kleywegt and Papastavrou, 1998, 2001; Papastavrou et al., 1996). In the dynamic stochastic knapsack problem, requests arrive sequentially, and their resource requirements and rewards are unknown prior to arrival but revealed upon arrival. In SADR problem, requests also arrive sequentially; however, the sizes of resource requirements and rewards depend on the type of request and are revealed upon arrival. Additionally, the SADR problem involves multiple knapsacks, adding another layer of complexity.

There is limited research on the dynamic or stochastic multiple knapsack problem. (Perry and Hartman, 2009) employs multiple knapsacks to model multiple time periods for solving a multiperiod, single-resource capacity reservation problem. Essentially, it remains a dynamic knapsack problem but with time-varying capacity. (Tönissen et al., 2017) considers a two-stage stochastic multiple knapsack problem with a set of scenarios where the capacity of the knapsacks can be subject to disturbances. This problem is similar to SPSR problem in our work, where the number of items is stochastic.

Generally speaking, the SADR problem relates to the *revenue management* (RM) problem, which has been extensively studied in industries such as airlines, hotels, and car rentals, where perishable inventory must be allocated dynamically to maximize revenue (Van Ryzin and Talluri, 2005). Network Revenue Management (NRM) extends traditional RM by considering multiple resources (e.g., flight legs, hotel nights) and interdependent demand (Williamson, 1992). The standard NRM problem is typically formulated as a dynamic programming (DP) model, where decisions involve accepting or rejecting requests based on their revenue contribution and remaining capacity (Talluri and Van Ryzin, 1998). However, a significant challenge arises because the number of states grows exponentially with the problem size, rendering direct solutions computationally infeasible. To address this, various control policies have been proposed, such as bid-price (Adelman, 2007; Bertsimas and Popescu, 2003), booking limits (Gallego and Van Ryzin, 1997), and dynamic programming decomposition (Talluri and Van Ryzin, 2006; Liu and Van Ryzin, 2008). These methods basically assume that demand arrives individually (e.g., one seat per booking). However, in our problem, customers often request multiple units simultaneously. The decisions in our problem must be made on an all-or-none basis for each request, which introduces additional complexity in handling group arrivals (Talluri and Van Ryzin, 2006). The introduction of group-based characteristics complicates seat management.

Prior work has examined group arrivals in the contexts such as: in hotel revenue management (Bitran and Mondschein, 1995; Goldman et al., 2002), where customers request multi-day stays but room assignments can be deferred until check-in; in high-speed train ticket sales Zhu et al. (2023), where each booking must be assigned to a specific seat for the entire multi-leg journey in real time. Both scenarios involve assignment challenges, but with key differences: hotels allow flexible room allocation, while train seating requires immediate assignment. Our SADR problem aligns more closely with the latter - requiring real-time group assignment - but differs in that requests specify only group size, without fixed start dates (for hotels) or boarding stops (for trains).

3 Seat Planning Problem with Social Distancing

We formally describe the problem of incorporating social distancing measures into the seat planning process. We first introduce several key concepts, then present an optimization model for the problem with deterministic requests.

3.1 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.

To adhere to social distancing requirements, individuals from the same group must sit together in one specific row while maintaining a distance, measured by the number of empty seats, from adjacent groups in the same row. Let δ denote the social distance, which could require leaving one or more empty seats between groups. Specifically, each group must ensure that there are empty seats adjacent to other groups. To model the social distancing requirements into the seat planning process, we define the size of group type i as $n_i = i + \delta$, where $i \in \mathcal{M}$. Correspondingly, the size of each row is defined as $L_j = L_j^0 + \delta$. This is a clear one-to-one mapping between the original physical seat plan and the model of our seat plan. By incorporating additional seats and designating certain seats for social distancing, we can integrate social distancing measures into the seat planning problem. Since each group occupies only one row, we assume that the physical distance between different rows is sufficient. If the social distancing requirement is more stringent, an empty row can be implemented, as practiced by some theaters (Lonely Planet, 2020).

We introduce the term *pattern* to describe the seat planning arrangement for a single row. A specific pattern can be represented by a vector $\boldsymbol{h} = (h_1, \ldots, h_M)$, where h_i denotes the number of groups of type i in the row for $i = 1, \ldots, M$. A feasible pattern, $\boldsymbol{h}$, must satisfy the condition $\sum_{i=1}^{M} h_i n_i \leq L$ and belong to the set of non-negative integer values, denoted as $\boldsymbol{h} \in \mathbb{N}^M$. A seat plan with N rows can be represented as $\boldsymbol{H} = [\boldsymbol{h}_1^\intercal, \ldots, \boldsymbol{h}_N^\intercal]$, where each element, H_{ij}, denotes the number of groups of type i contained in row j. The supply of the seat plan is represented by $\boldsymbol{X} = (X_1, \ldots, X_M)^\intercal$, where $X_i = \sum_{j=1}^{N} H_{ij}$ indicates the supply for group type i. In other words, $\boldsymbol{X}$ captures the number of groups of each type that can be accommodated in the seat layout by aggregating the supplies across all rows.

Let $|\boldsymbol{h}|$ denote the maximum number of individuals that can be assigned according to pattern $\boldsymbol{h}$, i.e.,

$|\boldsymbol{h}| = \sum_{i=1}^{M} ih_i$. The size of $\boldsymbol{h}$, $|\boldsymbol{h}|$, serves as a measure of the maximum seat occupancy achievable under social distancing constraints. By analyzing $|\boldsymbol{h}|$ across different patterns, we can evaluate the effectiveness of various seat plan configurations in accommodating the desired number of individuals while complying with social distancing requirements.

We use an example (Figure 1) to illustrate the above description of the seat planning problem. In this example, the size of the row is $L = L^0 + \delta = 11$. The seat plan for the row can be represented by $\boldsymbol{h} = (2, 1, 1, 0)$ with $|\boldsymbol{h}| = 7$.

Example 1. *Consider a single row of $L^0 = 10$ seats and the social distancing requirement of $\delta = 1$ empty seat between groups. There are four groups, groups 2 and 4 in group type 1, group 1 in type 2, and group 3 in type 3.*

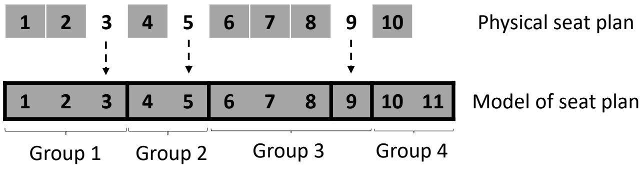


Figure 1: Illustration of Groups with Social Distancing

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, $i = 1, \ldots, \tilde{i} - 1$; $\sum_j x_{ij}^* = d_i$ for $i = \tilde{i} + 1, \ldots, M$; $\sum_j x_{ij}^* = \frac{L - \sum_{i=\tilde{i}+1}^{M} d_i n_i}{n_{\tilde{i}}}$ for $i = \tilde{i}$.*

Proposition 1 implies that, in the optimal fractional seat plan, the requests from larger groups ($i > \tilde{i}$)

are fully accommodated by the available supply. The requests from smaller groups ($i < \tilde{i}$) are rejected. Any remaining supply, if available, will be used to partially satisfy the request from group type $\tilde{i}$.

3.2 Seat Planning with Full or Largest Patterns

While the SPDR problem is straightforward to solve for practical problems, the optimal solution reveals some interesting observations about the structure of the optimal seat plan. Specifically, when some constraints (2) are not binding, most constraints (3) tend to be tight. In other words, in the case of supply less than demand, the seats in each row are likely to be fully used. It is worth noting that each constraint in (3) corresponds to a seating pattern of a row. This insight leads to the following definition.

Definition 1. *Consider a pattern $\boldsymbol{h} = (h_1, \ldots, h_M)$ for a row of size L. We say $\boldsymbol{h}$ to be a full pattern if $\sum_{i=1}^{M} n_i h_i = L$, and $\boldsymbol{h}$ to be a largest pattern if its size $|\boldsymbol{h}| \geq |\boldsymbol{h}'|$, for any other feasible pattern $\boldsymbol{h}'$.*

The above concepts characterize two types of tightness in a seating pattern. In a full pattern, the corresponding constraint is mathematically tight. In a maximum pattern, mathematical tightness is not enforced on the constraint; instead, it is reflected with respect to the objective function in that any other pattern cannot yield a higher objective function value. Both full pattern and maximum patterns indicate efficient use of seats and shall be used in an optimal seat plan.

Proposition 2. *The size of a largest pattern, denoted by $\phi(M, L^0, \delta)$ as a function of M, L^0 and δ, is given by*

$$\phi(M, L^0, \delta) = qM + \max\{r - \delta, 0\},$$

where q is the quotient of $(L^0 + \delta)$ divided by $(M + \delta)$ and r is the remainder. In addition, $\phi(M, L^0, \delta)$ is non-decreasing in M and L^0, and non-increasing in δ, respectively.

The size, $qM + \max\{r - \delta, 0\}$, corresponds directly to a largest pattern which includes q group type M and r seats allocated to a group type $(r - \delta)$ when $r > \delta$. However, the form of the largest pattern is not unique; there are other largest patterns with the same size. Specifically, when $r = 0$, the largest pattern is unique and full, indicating that only one pattern can accommodate the maximum number of individuals; when $r > \delta$, the largest pattern is full, as it uses all available seats. A concrete example illustrating largest and full patterns is provided in Example 2.

We can also measure the effectiveness of a seat plan by measuring the relative utilization of the seats. To this end, we define the occupancy rate of a pattern h as $|h|/L^0$. By this definition, a largest pattern achieves the maximum achievable occupancy rate of a row. Similarly, we can define the maximum achievable occupancy rate of the venue as follows.

$$\rho_{ac} = \frac{\sum_{j \in \mathcal{N}} \phi(M, L_j^0, \delta)}{\sum_{j \in \mathcal{N}} L_j^0},$$

where $\phi(M, L^0, \delta)$ represents the size of the largest pattern under M, L^0 and δ.

The monotonicity of $\phi(M, L^0, \delta)$ stated in Proposition 2 also applies to the maximum achievable occupancy rate with respect to M and δ, but not with respect to L^0. This result is established in the following corollary. Further discussion on the maximum achievable occupancy rate will be provided in Section 6.2.

Corollary 1. *The maximum achievable occupancy rate is non-decreasing in M and non-increasing in δ, but not monotone in $L_j^0, \forall j \in \mathcal{N}$.*

Example 2. *Consider the given values: $\delta = 1$, $L = 21$, and $M = 4$. The size of the largest pattern can be calculated as $qM + \max\{r - \delta, 0\} = 4 \times 4 + 0 = 16$. The largest patterns are as follows: $(1, 0, 1, 3)$, $(0, 1, 2, 2)$, $(0, 0, 0, 4)$, $(0, 0, 4, 1)$, and $(0, 2, 0, 3)$. Among these, $(0, 0, 0, 4)$ is the form referenced in Proposition 2.*

The following figure shows that the largest pattern may not be full and the full pattern may not be largest.

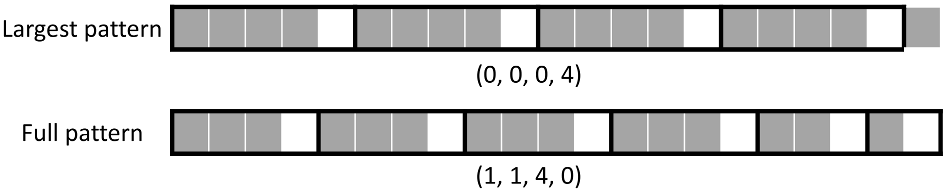


Figure 2: Largest and Full Patterns

Pattern $(0, 0, 0, 4)$ is a largest pattern as its size is 16. However, it does not satisfy the requirement of fully utilizing all available seats since $4 \times 5 \neq 21$. Pattern $(1, 1, 4, 0)$ is a full pattern as it utilizes all available seats. However, its size is 15, indicating that it is not a largest pattern.

Given a feasible seat plan, our goal is to derive a seat plan that satisfies the original requirements of the feasible seat plan while utilizing as many seats as possible. Will the optimal seat plan consist of either full or largest patterns? To address this question, let the feasible seat plan be $\boldsymbol{H}$ and the desired seat plan be $\boldsymbol{H}'$. To satisfy the requirements of planned group types in $\boldsymbol{H}$, the total quantity of groups from type i to type M in $\boldsymbol{H}'$ must be at least equal to the total quantity from group type i to group type M in $\boldsymbol{H}$. Mathematically, we aim to find a feasible seat plan $\boldsymbol{H}'$ such that $\sum_{k=i}^{M} \sum_{j=1}^{N} H_{kj} \leq \sum_{k=i}^{M} \sum_{j=1}^{N} H'_{kj}, \forall i \in \mathcal{M}$. We say $\boldsymbol{H} \subseteq \boldsymbol{H}'$ if this condition is satisfied.

To utilize all seats in the seat plan, the objective is to maximize the number of individuals that can be accommodated. Thus, we have the following formulation:

$$\begin{aligned} \max \quad & \sum_{i=1}^{M} \sum_{j=1}^{N} (n_i - \delta) x_{ij} \\ s.t. \quad & \sum_{k=i}^{M} \sum_{j=1}^{N} x_{kj} \geq \sum_{k=i}^{M} \sum_{j=1}^{N} H_{kj}, i \in \mathcal{M} \\ & \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} \\ & x_{ij} \in \mathbb{N}, i \in \mathcal{M}, j \in \mathcal{N} \end{aligned} \tag{4}$$

Proposition 3. *Given a feasible seat plan $\boldsymbol{H}$, the optimal solution to problem (4) corresponds to a seat plan $\boldsymbol{H}'$ such that $\boldsymbol{H} \subseteq \boldsymbol{H}'$ and $\boldsymbol{H}'$ is composed of either full or largest patterns.*

This approach guarantees the seat allocation with full or largest patterns while still accommodating the original groups' requirements. Furthermore, the improved seat plan can be used for the seat assignment when the group arrives sequentially.

4 Seat Planning with Stochastic Requests

We now investigate the problem of seat planning with stochastic requests. Specifically, we need to determine a seating plan that accommodates uncertain requests. To model this problem, we develop a scenario-based stochastic programming (SBSP) and follow the framework of Benders decomposition to solve it.

4.1 Scenario-Based Stochastic Programming Formulation

Assume the requests can be defined as a set of scenarios, denoted by $\omega \in \Omega$. Each scenario ω is associated with a specific realization of group requests, represented as $\mathbf{d}_\omega = (d_{1\omega}, d_{2\omega}, \ldots, d_{M,\omega})^\intercal$, and p_ω, the realization probability of the scenario ω. To maximize the expected number of individuals accommodated across all scenarios, we propose a scenario-based stochastic programming approach to determine a seat plan.

Recall that x_{ij} given in (1) represents the number of groups of type i planned in row j. To account for the variability across different scenarios, it is essential to model potential excess or shortage of supply. To capture this, we introduce a scenario-dependent decision variable, denoted as $\mathbf{y}$, which consists of two vectors: $\mathbf{y}^+ \in \mathbb{N}^{M\times|\Omega|}$ and $\mathbf{y}^- \in \mathbb{N}^{M\times|\Omega|}$. Here, each component of $\mathbf{y}^+$, denoted as $y^+_{i\omega}$, represents the excess of supply for group type i under scenario ω, while $y^-_{i\omega}$ represents the shortage of supply for group type i under scenario ω.

To address the possibility of larger group types being unable to fully occupy their designated seats due to insufficient supply, we assume that surplus seats for group type i can be allocated to smaller group types $j < i$ in descending order of group size. This implies that if there is excess supply after assigning groups of type i to rows, the remaining seats can be hierarchically allocated to groups of type $j < i$ based on their sizes. Recall that the supply for group type i is denoted as $\sum_{j=1}^N x_{ij}$. Thus, for any scenario ω, the excess and shortage of supply can be recursively defined as follows:

$$
\begin{aligned}
&y^+_{i\omega} = (\sum_{j=1}^N x_{ij} - d_{i\omega} + y^+_{i+1,\omega})^+, i = 1, \ldots, M-1 \\
&y^-_{i\omega} = (d_{i\omega} - \sum_{j=1}^N x_{ij} - y^+_{i+1,\omega})^+, i = 1, \ldots, M-1 \\
&y^+_{M\omega} = (\sum_{j=1}^N x_{Mj} - d_{M\omega})^+ \\
&y^-_{M\omega} = (d_{M\omega} - \sum_{j=1}^N x_{Mj})^+,
\end{aligned}
\tag{5}
$$

where $(\cdot)^+$ denotes the non-negative part of the expression.

Based on the considerations outlined above, the total supply of group type i under scenario ω can be expressed as: $\sum_{j=1}^{N} x_{ij} + y^{+}_{i+1,\omega} - y^{+}_{i\omega}, i = 1, \ldots, M-1$. For the special case of group type M, the total supply under scenario ω is $\sum_{j=1}^{N} x_{Mj} - y^{+}_{M\omega}$. Then we have the following formulation:

$$\text{(SBSP)} \quad \max \quad E_\omega \left[(n_M - \delta)(\sum_{j=1}^{N} x_{Mj} - y^{+}_{M\omega}) + \sum_{i=1}^{M-1} (n_i - \delta)(\sum_{j=1}^{N} x_{ij} + y^{+}_{i+1,\omega} - y^{+}_{i\omega}) \right] \tag{6}$$

$$\text{s.t.} \quad \sum_{j=1}^{N} x_{ij} - y^{+}_{i\omega} + y^{+}_{i+1,\omega} + y^{-}_{i\omega} = d_{i\omega}, \quad i = 1, \ldots, M-1, \omega \in \Omega \tag{7}$$

$$\sum_{j=1}^{N} x_{ij} - y^{+}_{i\omega} + y^{-}_{i\omega} = d_{i\omega}, \quad i = M, \omega \in \Omega \tag{8}$$

$$\sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} \tag{9}$$

$$y^{+}_{i\omega}, y^{-}_{i\omega} \in \mathbb{N}, \quad i \in \mathcal{M}, \omega \in \Omega$$

$$x_{ij} \in \mathbb{N}, \quad i \in \mathcal{M}, j \in \mathcal{N}.$$

The objective function consists of two parts. The first part represents the number of individuals in group type M that can be accommodated, given by $(n_M - \delta)(\sum_{j=1}^{N} x_{Mj} - y^{+}_{M\omega})$. The second part represents the number of individuals in group type i, excluding M, that can be accommodated, given by $(n_i - \delta)(\sum_{j=1}^{N} x_{ij} + y^{+}_{i+1,\omega} - y^{+}_{i\omega}), i = 1, \ldots, M-1$. The overall objective function is subject to an expectation operator denoted by E_ω, which represents the expectation with respect to the scenario set. This implies that the objective function is evaluated by considering the average values of the decision variables and constraints over the different scenarios. The objective function can be then reformulated as follows.

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

Here, $\sum_{j=1}^{N} \sum_{i=1}^{M} i \cdot x_{ij}$ indicates the maximum number of individuals that can be accommodated in the seat plan $\{x_{ij}\}$. The second part, $\sum_{\omega \in \Omega} p_\omega \sum_{i=1}^{M} y^{+}_{i\omega}$ indicates the expected excess of supply for group type i over scenarios.

In the optimal solution, at most one of $y^{+}_{i\omega}$ and $y^{-}_{i\omega}$ can be positive for any i, ω. Suppose there exist i_0 and ω_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 (7) and (8) but increase the objective value when p_{ω_0} is positive. Thus, in the optimal solution, at most one of $y^{+}_{i\omega}$ and $y^{-}_{i\omega}$ can be positive.

Proposition 4. *There exists an optimal solution to SBSP given in* (6) *such that the patterns* *associated with this optimal solution are composed of the full or largest patterns under any given scenarios.*

When there is only one scenario, SBSP reduces to the deterministic model. This aligns with Section 3.2, which outlines the generation of seat plan consisting of full or largest patterns.

Solving SBSP directly is computationally prohibitive when there are numerous scenarios, instead, we apply Benders decomposition to simplify the solving process in Section 4.2, then obtain the seat plan composed of full or largest patterns, as stated in Section 4.3.

4.2 Solving SBSP via Benders Decomposition

We apply Benders decomposition (Benders, 1962) to solve SBSP by partitioning it into a master problem with decisions $\mathbf{x}$ and a set of independent scenario-specific subproblems. To link the decomposed problems, we introduce scenario-specific auxiliary variables z_ω in the master problem, each representing a lower bound on the optimal value of the subproblem through the optimality cuts. At each iteration, the values of the master problem variables, $\mathbf{x}$ and z_ω are first determined. Then the dual solutions of the subproblems can be computed with $\mathbf{x}$. For each subproblem, an optimality cut incorporating its dual solution and the auxiliary variable z_ω is added to the master problem. The lower bound of SBSP (LB) is derived from the feasible subproblems and the upper bound (UB) is obtained from the master problem's optimal value. The iterative process continues until the gap of UB $-$ LB is smaller than a given tolerance or an optimal solution is found.

4.2.1 Definition and Solution Property of Subproblems

To facilitate the presentation, we express the deterministic equivalent of SBSP in compact matrix form with the following components. The objective components are $\mathbf{c}^\intercal \mathbf{x} = \sum_{j=1}^{N} \sum_{i=1}^{M} i \cdot x_{ij}, \mathbf{f}^\intercal \mathbf{y}_\omega = -\sum_{i=1}^{M} y_{i\omega}^{+}$. Let $\mathbf{n} = (n_1, \ldots, n_M)^\intercal$ denote group sizes and $\mathbf{L} = (L_1, \ldots, L_N)^\intercal$ represent row sizes, $\mathbf{x1}$ indicate the supply of group types, where $\mathbf{1}$ is a column vector of size N with all elements equal to 1, $\mathbf{V} = [\mathbf{W},\ \mathbf{I}]$, where $\mathbf{W}$ is an $M \times M$ lower-bidiagonal matrix:

$$\mathbf{W} = \begin{bmatrix} -1 & 1 & 0 & \ldots & \ldots & 0 \\ 0 & -1 & 1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & -1 & 1 & 0 \\ 0 & \ldots & \ldots & 0 & -1 & 1 \\ 0 & \ldots & \ldots & \ldots & 0 & -1 \end{bmatrix}$$

and $\mathbf{I}$ is the M-dimensional identity matrix.

The complete formulation can be expressed as:

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} \mathbf{f}^\intercal \mathbf{y}_\omega \\ \text{s.t.} \quad & \mathbf{x}\mathbf{1} + \mathbf{V}\mathbf{y}_\omega = \mathbf{d}_\omega, \forall \omega \in \Omega \\ & \mathbf{x}^\intercal \mathbf{n} \leq \mathbf{L} \\ & \mathbf{x} \in \mathbb{N}^{M \times N}, \mathbf{y}_\omega \in \mathbb{N}^{2M}, \forall \omega \in \Omega, \end{aligned} \tag{10}$$

which can also be formulated as:

$$\max_{\mathbf{x} \in \mathbb{N}^{M \times N}} \left\{ \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} \max_{\mathbf{y}_\omega \in \mathbb{N}^{2M}} \left\{ \mathbf{f}^\intercal \mathbf{y}_\omega \middle| \mathbf{V}\mathbf{y}_\omega = \mathbf{d}_\omega - \mathbf{x}\mathbf{1} \right\} \middle| \mathbf{x}^\intercal \mathbf{n} \leq \mathbf{L} \right\}. \tag{11}$$

Given $\mathbf{x} = \bar{\mathbf{x}}$, for each scenario $\omega \in \Omega$, the subproblem can be defined by:

$$z_\omega(\bar{\mathbf{x}}) = \max_{\mathbf{y}_\omega \geq 0} \left\{ \mathbf{f}^\intercal \mathbf{y}_\omega \middle| \mathbf{V}\mathbf{y}_\omega = \mathbf{d}_\omega - \bar{\mathbf{x}}\mathbf{1} \right\}. \tag{12}$$

The dual of the subproblem (12) can be expressed as:

$$z_\omega(\bar{\mathbf{x}}) = \min_{\boldsymbol{\alpha}_\omega} \left\{ \boldsymbol{\alpha}_\omega^\intercal (\mathbf{d}_\omega - \bar{\mathbf{x}}\mathbf{1}) \middle| \boldsymbol{\alpha}_\omega^\intercal \mathbf{V} \geq \mathbf{f}^\intercal \right\}, \tag{13}$$

where $\boldsymbol{\alpha}_\omega = (\alpha_{1\omega}, \alpha_{2\omega}, \ldots, \alpha_{M,\omega})^\intercal$ denote the vector of dual variables.

Lemma 1. *The feasible region of problem* (13) *is nonempty and bounded. Furthermore, all the extreme points of the feasible region are integral.*

Lemma 1 establishes that the optimal value of (13), $z_\omega(\mathbf{x})$, is always finite, implying that only optimality cuts are necessary to be added in the master problem.

When $\mathbf{x}$ is fixed, the scenario-specific variables $\mathbf{y}_\omega$ can be computed via the recursive relations in (5). Recall that for the optimal $y_{i\omega}^+$ and $y_{i\omega}^-$, at most one can be positive. The dual variables $\alpha_{i\omega}$ can be obtained via a piecewise recursive scheme established in Proposition 5, where each $\alpha_{i\omega}$ is determined by the active case of $y_{i\omega}^\pm$ (i.e., whether $y_{i\omega}^+ > 0$, $y_{i\omega}^- > 0$, or both are zero). The scheme is initialized with $\alpha_{0,\omega} = 0$ for all $\omega \in \Omega$.

Proposition 5. *The optimal solutions to problem* (13) *are given by*

$$\begin{aligned} & \alpha_{i\omega} = \alpha_{i-1,\omega} + 1, \quad \textit{if } y_{i\omega}^+ > 0, i = 1, \ldots, M \\ & \alpha_{i\omega} = 0, \quad \textit{if } y_{i\omega}^- > 0, i = 1, \ldots, M \textit{ or } y_{i\omega}^- = y_{i\omega}^+ = 0, y_{i+1,\omega}^+ > 0, i = 1, \ldots, M-1 \\ & 0 \leq \alpha_{i\omega} \leq \alpha_{i-1,\omega} + 1, \quad \textit{if } y_{i\omega}^- = y_{i\omega}^+ = 0, i = M \textit{ or } y_{i\omega}^- = y_{i\omega}^+ = 0, y_{i+1,\omega}^+ = 0, i = 1, \ldots, M-1 \end{aligned} \tag{14}$$

When $y_{i\omega}^-$ and $y_{i\omega}^+$ jointly satisfy the third condition in Proposition 5, the dual variable $\alpha_{i\omega}$ admits multiple feasible values. To maintain strong optimality cuts, we select the extreme point of the feasible region by setting: $\alpha_{i\omega} = \alpha_{i-1,\omega} + 1$.

4.2.2 Master Problem and Iterative Procedure

We now construct the master problem through dual analysis of the subproblems. Let $\mathbb{P}$ denote the feasible region of the dual subproblem (13), with $\mathcal{O}$ representing its set of extreme points. The subproblem's optimal value admits the dual representation: $z_\omega(\mathbf{x}) = \min_{\boldsymbol{\alpha}_\omega \in \mathcal{O}} \boldsymbol{\alpha}_\omega^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1})$. This yields optimality cuts for the master problem of the form: $\boldsymbol{\alpha}_\omega^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_w, \forall \boldsymbol{\alpha}_\omega \in \mathcal{O}$, where z_w is an auxiliary variable in problem (15) providing a lower bound approximation of $z_\omega(\mathbf{x})$. Let $\mathcal{O}^s$ denote the subset of $\mathcal{O}$. The master problem is then formulated as:

$$
\begin{aligned}
\max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} p_\omega z_\omega \\
\text{s.t.} \quad & \mathbf{x}^\intercal \mathbf{n} \leq \mathbf{L} \\
& \boldsymbol{\alpha}_\omega^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega, \forall \boldsymbol{\alpha}_\omega \in \mathcal{O}^s, \forall \omega \\
& \mathbf{x} \in \mathbb{N}^{M \times N}, z_\omega \in \mathbb{R}, \forall \omega
\end{aligned}
\tag{15}
$$

To find the optimal solution to SBSP, we iteratively solve the master problem (15) and incrementally adding more cuts. The algorithm begins by initializing the cut set $\mathcal{O}^s$ using extreme points of the dual feasible region $\mathbb{P}$, where in practice we set $\boldsymbol{\alpha}_\omega = \mathbf{0}, \forall \omega$. Solving the initialized master problem yields a candidate solution $(\mathbf{x}^*, \mathbf{z}^*)$ with $\mathbf{z}^* = (z_1^*, \ldots, z_{|\Omega|}^*)$. Then $c^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega z_\omega^*$ is an upper bound of SBSP (UB). Subsequently, fixing $\mathbf{x}^*$ allows us to solve the dual of subproblem (13) to obtain the optimal dual variables, $\tilde{\boldsymbol{\alpha}}_\omega$, via Proposition 5. The resulting values $\tilde{z}_\omega = \tilde{\boldsymbol{\alpha}}_\omega(d_\omega - \mathbf{x}^*\mathbf{1})$ yield a feasible solution $(\mathbf{x}^*, \tilde{\mathbf{z}})$ that produces a valid lower bound of SBSP (LB) through $\mathbf{c}^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega \tilde{z}_\omega$.

The algorithm proceeds iteratively by verifying optimality conditions for each scenario. Whenever any scenario ω for which the optimal value of problem (13) is less than z_ω^*, indicating that the constraints are not fully satisfied, we generate and add the corresponding optimality cut $(\tilde{\boldsymbol{\alpha}}_\omega)^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega$ to the master problem (15). Conversely, for every scenario ω, the optimal value of the problem (13) is larger than or equal to z_ω^*, which means that all contraints in SBSP are satisfied, the current solution $(\mathbf{x}^*, \mathbf{z}^*)$ is confirmed as optimal for SBSP.

The algorithm's convergence is guaranteed by the following properties: First, we only add optimality cuts to the master problem in each iteration, which ensures the upper bound (UB) decreases monotonically as the feasible region becomes progressively tighter. Second, while the initial cuts $\boldsymbol{\alpha}_\omega = \mathbf{0}, \forall \omega$ don't guarantee monotonic lower bound (LB) improvement, the finite convergence of the gap UB − LB to zero is assured because the dual feasible region P has finitely many extreme points and each iteration introduces new cuts corresponding to previously unexplored extreme points. The algorithm terminates when UB − LB $< \epsilon$ for a prescribed tolerance $\epsilon > 0$.

Algorithm 1 presents the complete Benders decomposition procedure.

4.3 Generating Seat Plans with Full or Largest Patterns

Solving problem (15) iteratively with integrality constraints can be computationally intensive in certain cases. To address this, we adopt a two-phase approach: first, we solve the LP relaxation of SBSP

Algorithm 1: Benders Decomposition

```
Initialize $\boldsymbol{\alpha}_\omega = \mathbf{0}, \forall \omega$, $\epsilon > 0$, and let $LB \leftarrow 0$, $UB \leftarrow \infty$;
while $UB - LB > \epsilon$ do
    Solve problem (15) and obtain an optimal solution $(\mathbf{x}^*, \mathbf{z}^*)$;
    $UB \leftarrow c^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega z_\omega^*$;
    for $\omega = 1, \ldots, |\Omega|$ do
        Obtain $\tilde{\boldsymbol{\alpha}}_\omega$ according to Proposition 5;
        $\tilde{z}_\omega = (\tilde{\boldsymbol{\alpha}}_\omega)^\intercal (\mathbf{d}_\omega - \mathbf{x}^* \mathbf{1})$;
        if $\tilde{z}_\omega < z_\omega^*$ then
            Add one new constraint, $(\tilde{\boldsymbol{\alpha}}_\omega)^\intercal (\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega$, to problem (15);
        end
    end
    $LB \leftarrow c^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega \tilde{z}_\omega$;
end
```

using Benders decomposition to obtain a fractional solution; from this relaxed solution, we then generate a seat plan consisting of full or largest patterns.

When solving the LP relaxation of SBSP using the Benders decomposition method described above, we can obtain a fractional optimal solution. This solution represents the optimal partial allocation of groups to seats in the relaxed problem. Based on the fractional solution obtained, we use the deterministic model to generate a feasible seat plan. The objective of this model is to allocate groups to seats in a way that satisfies the supply requirements for each group without exceeding the corresponding supply values obtained from the fractional solution. To accommodate more groups and optimize seat utilization, we aim to construct a seat plan composed of full or largest patterns based on the feasible seat plan obtained in the previous step.

Let $\mathbf{x}^*$ denote the optimal solution to the LP relaxation of SBSP. Aggregate $\mathbf{x}^*$ to the number of each group type, defined as $\tilde{X}_i = \sum_j x_{ij}^*, \forall i \in \mathbf{M}$. Next, solve the SPDR problem with $\boldsymbol{d} = \tilde{\boldsymbol{X}}$ to obtain the optimal solution $\tilde{\mathbf{x}}$, and the corresponding pattern $\tilde{\boldsymbol{H}}$. Then generate the seat plan by problem (4) with $\boldsymbol{H} = \tilde{\boldsymbol{H}}$.

Algorithm 2: Seat Plan Construction

```
Solve the LP relaxation of SBSP in (6), and obtain an optimal solution $\mathbf{x}^*$;
Solve the SPDR problem in (1) with $d_i = \sum_j x_{ij}^*, i \in \mathbf{M}$, and obtain an optimal solution $\tilde{\mathbf{x}}$ and
  the corresponding pattern, $\tilde{\boldsymbol{H}}$;
Solve problem (4) with $\boldsymbol{H} = \tilde{\boldsymbol{H}}$, and obtain the seat plan $\boldsymbol{H}'$.
```

5 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:

$$V^t(\mathbf{L}) = \max_{u_{i,j}^t \in U^t(\mathbf{L})} \left\{ \sum_{i=1}^{M} p_i \Big(\sum_{j=1}^{N} i u_{i,j}^t + V^{t+1}(\mathbf{L} - \sum_{j=1}^{N} n_i u_{i,j}^t \mathbf{e}_j) \Big) + p_0 V^{t+1}(\mathbf{L}) \right\} \tag{16}$$

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 (16) 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.

5.1 Seat-Plan-Based Assignment

The Seat-Plan-Based Assignment (SPBA) policy dynamically allocates groups through a two-stage process. In the first stage, requests are evaluated using relaxed dynamic programming (RDP). The second stage, known as group-type control, initially accepted requests are verified and assigned based on expected future revenue, considering the current seat plan and remaining time periods. As part of this stage, accepted requests are further assigned to specific rows according to tie-breaking rules. To enhance computational efficiency and avoid regenerating the seat plan in every period, we establish specific criteria for determining when to update the seat plan.

5.1.1 Relaxed Dynamic Programming (RDP)

To simplify the complexity of the DP formulation in (16), we employ a 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:

$$V^t(l) = \max_{u_i^t \in \{0,1\}} \left\{ \sum_i p_i \left[V^{t+1}(l - n_i u_i^t) + i u_i^t \right] + p_0 V^{t+1}(l) \right\} \tag{17}$$

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.

5.1.2 Group-Type Control

The group-type control allocation verifies and assigns requests initially accepted in the first stage. It assesses whether the current seat plan can accommodate the arriving group while balancing the trade-off between preserving seat availability for potential future requests and accepting the current request. To make this decision, we compare the expected number of acceptable individuals for both options. Accepted requests are then assigned to specific rows based on tie-breaking rules.

Specifically, suppose the supply at period t is $[X_1^t, \ldots, X_M^t]$, with $(T - t)$ remaining periods. A request of type i arrives and is initially accepted in the first stage. If $X_i^t > 0$, the request is accepted directly. If $X_i^t = 0$, the request can still be accepted by utilizing one unit of supply from group type $\hat{\imath}$ for any $\hat{\imath} = i + 1, \ldots, M$.

- When $\hat{\imath} = i + \delta + 1, \ldots, M$, the remaining $(\hat{\imath} - i - \delta)$ seats can be allocated to one additional group type $(\hat{\imath} - i - \delta)$, ensuring the social distancing of δ seats.

- When $\hat{i} = i+1, \ldots, i+\delta$, the expected number of accepted individuals is i, while the remaining $\hat{i} - i$ seats beyond the accepted group are wasted.

Let $D^t_{\hat{i}}$ be the random variable representing the number of future arrivals of group type $\hat{i}$ in the remaining t periods. The expected number of accepted individuals is given by:

$$i + (\hat{i} - i - \delta) P(D^{T-t}_{\hat{i}-i-\delta} \geq X^t_{\hat{i}-i-\delta} + 1),$$

where $P(D^{T-t}_i \geq X^t_i)$ represents the probability that demand for group type i in the remaining $(T-t)$ periods meets or exceeds the current remaining supply X^t_i. Thus, the term, $P(D^{T-t}_{\hat{i}-i-\delta} \geq X^t_{\hat{i}-i-\delta} + 1)$, specifically captures the probability that demand for group type $(\hat{i} - i - \delta)$ in future periods exceeds its current remaining supply by at least one unit.

Similarly, if we reject the current group type i to preserve capacity for potential future groups of type $\hat{i}$, the expected number of accepted individuals becomes:

$$\hat{i} P(D^{T-t}_{\hat{i}} \geq X^t_{\hat{i}}),$$

where $P(D^{T-t}_{\hat{i}} \geq X^t_{\hat{i}})$ represents the probability that the demand for group type $\hat{i}$ during the remaining $(T-t)$ periods meets or exceeds its current remaining supply $X^t_{\hat{i}}$.

Let $d^t(i, \hat{i})$ denote the difference of the expected number of accepted individuals between accepting a group type i (occupying $(\hat{i} + \delta)$-size seats) and rejecting it in period t. This difference is given by:

$$d^t(i, \hat{i}) = \begin{cases} i + (\hat{i} - i - \delta) P(D^{T-t}_{\hat{i}-i-\delta} \geq X^t_{\hat{i}^t-i-\delta} + 1) - \hat{i} P(D^{T-t}_{\hat{i}} \geq X^t_{\hat{i}}), & \text{if } \hat{i} = i + \delta + 1, \ldots, M \\ i - \hat{i} P(D^{T-t}_{\hat{i}} \geq X^t_{\hat{i}}), & \text{if } \hat{i} = i + 1, \ldots, i + \delta. \end{cases}$$

The optimal decision selects $\hat{i}^* = \arg\max_{\hat{i}=i+1,\ldots,M} d^t(i, \hat{i})$. The group is accepted and assigned to $(\hat{i}^* + \delta)$-size seats if $d^t(i, \hat{i}^*) \geq 0$, otherwise rejected. After determining the optimal group type $\hat{i}^*$, we apply the tie-breaking rule to assign the request to a specific row that includes group type $\hat{i}^*$.

Tie-Breaking for Row Selection

A tie occurs when there are several rows to assign the request. To determine the appropriate row for seat assignment, we can apply the following tie-breaking rules among the possible options. Suppose one request of type i arrives, the current seat plan is $\boldsymbol{H} = [\boldsymbol{h}_1^\intercal, \ldots, \boldsymbol{h}_N^\intercal]$, the corresponding supply is $\boldsymbol{X}$. Let $\beta_j = L_j - \sum_i n_i H_{ij}$ represent the remaining number of seats in row j after considering the seat allocation for the assigned requests. When $X_i > 0$, we assign the request to row $k \in \arg\min_{j \in \mathcal{N}} \{\beta_j | H_{ij} > 0\}$ such that the row can be filled as much as possible. When $X_i = 0$ and the request is accepted to take the seats planned for type $\hat{i}, \hat{i} > i$, we assign the request to a row $k \in \arg\max_{j \in \mathcal{N}} \{\beta_j | H_{\hat{i}j} > 0\}$. That can help reduce the number of rows that are not full. When there are multiple ks available, we can choose one arbitrarily.

5.1.3 Regenerating the Seat Plan

A useful technique often applied in network revenue management to enhance performance is resolving (Secomandi, 2008; Jasin and Kumar, 2012), which, in our context, corresponds to regenerating the seat plan. However, to optimize computational efficiency, it is unnecessary to regenerate the seat plan for every request. Instead, we adopt a more streamlined approach. Since seats allocated for the largest group type can accommodate all smaller group types, the seat plan must be regenerated when the supply for the largest group type reaches zero. This ensures that the largest group is not rejected due to infrequent updates. Additionally, regeneration is required after determining whether to assign the arriving group to seats originally planned for larger groups. By regenerating the seat plan in these specific situations, we integrate real-time information into seat assignment while reducing the frequency of planning updates, thereby balancing efficiency and effectiveness.

The algorithm for regenerating the seat plan is outlined below.

Algorithm 3: Seat-Plan-Based Assignment

```
Obtain $\boldsymbol{X}^1$ and $\boldsymbol{H}^1$ from Algorithm 2, calculate $V^t(l)$ by (17), $\forall t = 2, \ldots, T; \forall l = 1, 2, \ldots, \tilde{L}$;
for $t = 1, \ldots, T$ do
    Observe a request of group type $i$;
    if $V^{t+1}(l^t) \leq V^{t+1}(l^t - n_i) + i$ then
        if $X_i > 0$ then
            Set $k = \arg\min_j \{L_j^t - \sum_i n_i H_{ij}^t | H_{ij}^t > 0\}$, break ties arbitrarily;
            Assign group type $i$ in row $k$, let $L_k^{t+1} \leftarrow L_k^t - n_i$, $H_{ik}^{t+1} \leftarrow H_{ik}^t - 1$, $X_i^{t+1} \leftarrow X_i^t - 1$;
            if $i = M$ and $X_M^t = 0$ then
                Generate seat plan $\boldsymbol{H}^{t+1}$ from Algorithm 2, update the corresponding $\boldsymbol{X}^{t+1}$;
            end
        else
            Calculate $d^t(i, \hat{i}^*)$;
            if $d^t(i, \hat{i}^*) \geq 0$ then
                Set $k = \arg\max_j \{L_j^t - \sum_i n_i H_{ij}^t | H_{\hat{i}^* j}^t > 0\}$, break ties arbitrarily;
                Assign group type $i$ in row $k$, let $L_k^{t+1} \leftarrow L_k^t - n_i$, $l^{t+1} \leftarrow l^t$;
                Generate seat plan $\boldsymbol{H}^{t+1}$ from Algorithm 2, update the corresponding $\boldsymbol{X}^{t+1}$;
            else
                Reject group type $i$ and let $L_k^{t+1} \leftarrow L_k^t$, $l^{t+1} \leftarrow l^t$;
            end
        end
    else
        Reject group type $i$ and let $L_k^{t+1} \leftarrow L_k^t$, $l^{t+1} \leftarrow l^t$;
    end
end
```

6 Computational Experiments

We conduct several experiments, including analyzing the performances of different policies, evaluating the impact of implementing social distancing, and comparing the performance under varied constraints. In the experiments, we set the following parameters.

The default parameters in the experiments are as follows, the size of social distancing $\delta = 1$, the largest group size $M = 4$, the number of rows $N = 10$, and the size of each row $L_j = 21$, for all $j \in \mathcal{N}$. We simulate the arrival of exactly one group in each period, i.e., $p_0 = 0$. Each experiment result is the average of 100 instances. In each instance, the number of scenarios in SBSP is $|\Omega| = 1000$.

To assess the performances of different policies across varying demand levels, we conduct experiments spanning a range of 60 to 100 periods and we consider four probability distributions for our analysis: $D_1 : [0.18, 0.7, 0.06, 0.06]$ and $D_2 : [0.2, 0.8, 0, 0]$, $D_3 : [0.34, 0.51, 0.07, 0.08]$ and $D_4 : [0.12, 0.5, 0.13, 0.25]$. The first two distributions, D_1 and D_2, are experimented in Blom et al. (2022). Here, D_1 represents the statistical distribution of group sizes, while D_2 reflects a restricted situation where groups of more than 2 people are not allowed. The other two distributions, D_3 and D_4, are derived from real-world movie data. The specific procedure is detailed in EC.3. We use D_4 as the default probability distribution in the other experiments.

6.1 Performances of Different Policies

We evaluate four assignment policies: seat-plan-based assignment (SPBA), relaxed dynamic programming heuristic (RDPH), bid-price control (BPC), and booking-limit control (BLC). Detailed procedures of RDPH, BPC, and BLC are provided in EC.1. We compare their performances against the optimal policy derived from solving the deterministic model with perfect information of all requests. Table 1 quantifies their effectiveness using the ratio of assigned individuals under each policy to those assigned under the optimal policy.

Our results demonstrate that the SPBA policy consistently outperforms the RDPH, BPC, and BLC policies. The RDPH and BPC policies can make the initial decision to accept or reject a request, but lack the capability to optimize seat assignments. The BLC policy strictly adheres to predetermined booking limits; even if the supply of one type is exhausted, it does not utilize seats planned for other types to accept the request, leading to its limited effectiveness.

The performance of SPBA, RDPH, and BPC policies basically follows a pattern where it initially declines and then gradually improves as T increases. When T is small, the demand of requests is generally low, allowing these policies to achieve relatively optimal performance. However, as T increases, it becomes more challenging for these policies to consistently achieve a perfect allocation, resulting in a decrease in performance. Nevertheless, as T continues to grow, these policies tend to accept larger groups, thereby narrowing the gap between their performance and the optimal value. Consequently, their performances improve. In contrast, the BLC policy shows improved performance as T increases because it reduces the number of unoccupied seats reserved for the largest groups.

The performance of the policies can vary with different probabilities. For the different probability

Table 1: Performances of Different Policies

Distribution	T	SPBA (%)	RDPH (%)	BPC (%)	BLC (%)
D_1	60	100.00	100.00	100.00	88.56
	70	99.53	99.01	98.98	92.69
	80	99.38	98.91	98.84	97.06
	90	99.52	99.23	99.10	98.24
	100	99.58	99.27	98.95	98.46
D_2	60	100.00	100.00	100.00	93.68
	70	100.00	100.00	100.00	92.88
	80	99.54	97.89	97.21	98.98
	90	99.90	99.73	99.44	99.61
	100	100.00	100.00	100.00	99.89
D_3	60	100.00	100.00	100.00	91.07
	70	99.85	99.76	99.73	90.15
	80	99.22	98.92	98.40	96.98
	90	99.39	99.12	98.36	96.93
	100	99.32	99.18	98.88	97.63
D_4	60	99.25	99.18	99.13	93.45
	70	99.20	98.65	98.54	97.79
	80	99.25	98.69	98.40	98.22
	90	99.29	98.65	98.02	98.42
	100	99.60	99.14	98.32	98.68

distributions listed, the SPBA policy performs more stably and consistently for the same demand. In contrast, the performances of the other policies fluctuate more significantly.

6.2 Impact of Social Distancing

We introduce three key terms, the threshold of request-volume, the threshold of occupancy rate, and the maximum achievable occupancy rate, to describe the impact of implementing social distancing.

The *threshold of request-volume*, q^{th}, is defined as

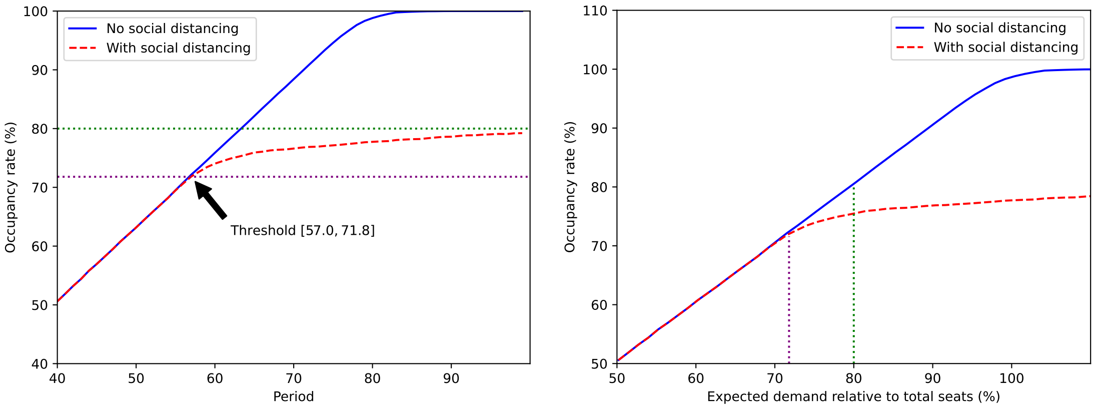
$$q^{\text{th}} = (1 - p_0) \cdot \max \left\{ T \,\middle|\, E^0(T) - E(T) < 1 \right\},$$

where $E(T)$ and $E^0(T)$ denotes the average number of assigned individuals by SPBA with social distancing level δ and without social distancing, respectively. Here, the maximization is performed over T while keeping all other parameters constant. Intuitively, the threshold of request-volume represents the maximum number of requests that can be accommodated while keeping the average loss below one.

The occupancy rate corresponding to the threshold of request-volume is referred to as the *threshold of occupancy rate*, ρ^{tr}. This rate represents the maximum occupancy rate when the difference in the number of assigned individuals remains unaffected by the social distancing requirement.

The *maximum achievable occupancy rate* is attained when each row of a given layout is the largest pattern, denoted by $\rho^{ac} = \frac{\sum_{j \in \mathcal{N}} \phi(M, L_j^0, \delta)}{\sum_{j \in \mathcal{N}} L_j^0}$, as introduced in Section 3.2.

We examine the impact of social distancing when implementing SPBA under varying levels of demand. Specifically, we calculate $E(T)$ with $\delta = 1$ and $E^0(T)$ across different values of T. The demand levels are varied by adjusting the parameter T from 40 to 100 in increments of 1. The results are visualized in Figure 3, which illustrates the occupancy rate under different demand levels.



(a) (b)

Figure 3: Impact of Social Distancing

Figure 3(a) illustrates the occupancy rate over period, revealing three key metrics for policy evaluation: (1) the threshold of request-volume, $q^{\text{th}} = 57$, (2) the threshold of occupancy rate, $\rho^{th} = 71.8\%$, and (3) the maximum achievable occupancy rate $\rho^{ac} = 80\%$. These metrics collectively serve as crucial indicators for assessing the effectiveness of the government policy.

Figure 3(b) presents the occupancy rate as a function of expected demand. The key distinction between Figure 3(a) and Figure 3(b) lies in their respective x-axes. From a demand perspective, we observe two distinct points: for expected demand below 71.8%, social distancing requirements impose no significant reduction in the number of assigned individuals; above this 71.8% threshold, the performance gap between scenarios with and without social distancing becomes increasingly pronounced.

Given the conceptual similarity between period-based and demand-based occupancy rate analyses, we focus on these three metrics to concisely represent the core findings. Complete results and detailed interpretations are presented in Section 6.4, which examines performance under various policy constraints.

6.3 Estimation of q^{th} and ρ^{th}

To estimate the threshold of request-volume, we aim to find the maximal period such that all requests can be assigned into the seats during these periods, i.e., for each group type i, we have $\boldsymbol{X}_i = \sum_j x_{ij} \geq d_i$. Meanwhile, we have the capacity constraint $\sum_i n_i x_{ij} \leq L_j$, thus, $\sum_i n_i d_i \leq \sum_i n_i \sum_j x_{ij} \leq \sum_j L_j$. Notice that $E(d_i) = p_i T$, we have $\sum_i n_i p_i T \leq \sum_j L_j$ by taking the expectation. The average number of individuals per period, denoted as γ, can be expressed as $\gamma = \sum_{i=1}^{M} i p_i$. Recall that $\tilde{L} = \sum_j L_j$ denotes the total size of all rows. From this, we derive the inequality $T \leq \frac{\tilde{L}}{\gamma + \delta}$. Under the ideal assumption that all requests fully occupy the available capacity, the ideal threshold of request-volume can be estimated as $\hat{q}^{\text{th}} = (1 - p_0)\frac{\tilde{L}}{\gamma+\delta}$. Correspondingly, the threshold of occupancy rate under this ideal scenario is: $\hat{\rho}^{\text{th}} = \frac{\gamma \hat{q}^{\text{th}}}{(\gamma+\delta)\hat{q}^{\text{th}} - N\delta} = \frac{\gamma}{\gamma+\delta} \frac{(1-p_0)\tilde{L}}{(1-p_0)\tilde{L} - N\delta}$.

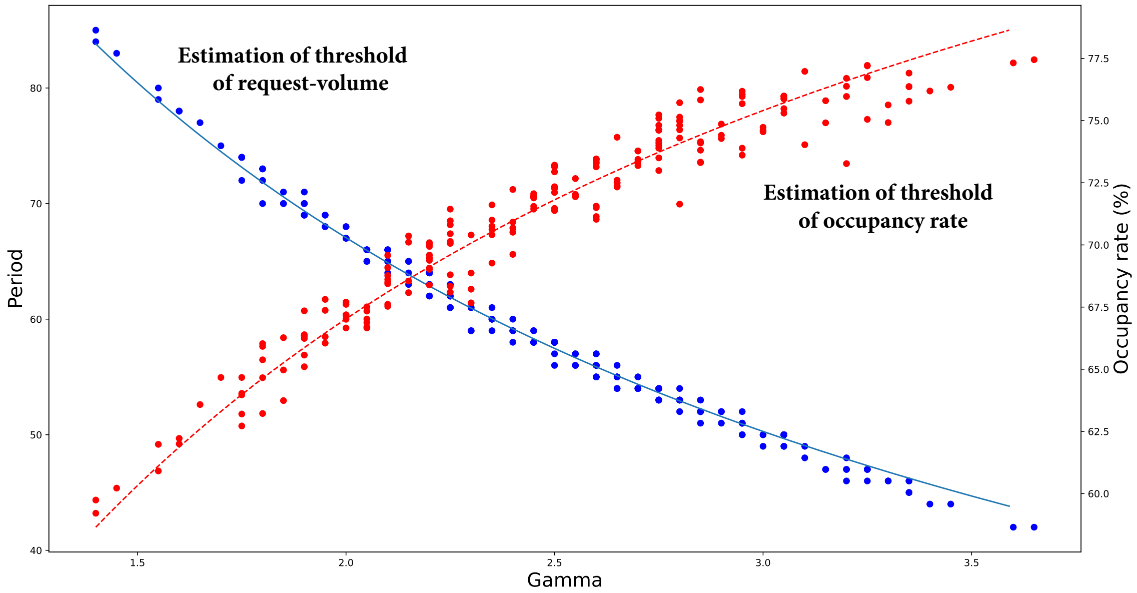
However, in practice, the actual expected maximum request-volume will be smaller than $\hat{q}^{\text{th}}$ since perfectly filling all available capacity is nearly impossible. To account for this, we introduce discount factors c_1 and c_2 for both thresholds, respectively. For the threshold of request-volume: $\tilde{q}^{th} = \frac{c_1(1-p_0)\tilde{L}}{\gamma+\delta}$, where c_1 adjusts for deviations from the ideal assumption. For the threshold of occupancy rate: $\tilde{\rho}^{th} =$

$\frac{c_2\gamma}{\gamma+\delta}\frac{(1-p_0)\tilde{L}}{(1-p_0)\tilde{L}-N\delta}$, where c_2 similarly reflects practical constraints.

To analyze the relation between the threshold of request-volume, the threshold of occupancy rate and γ, we conducted a study using a sample of 200 probability distributions with $M = 4, p_0 = 0$. The figure below shows the threshold of request-volume and the threshold of occupancy rate as functions of γ, along with their corresponding estimations.

We applied an Ordinary Least Squares model to fit the data and estimate the parameters. The resulting fitted equations, $\tilde{q}^{th} = \frac{c_1\tilde{L}}{\gamma+\delta}$ (represented by the solid line in the figure) and $\tilde{\rho}^{th} = \frac{c_2\gamma}{\gamma+\delta}\frac{\tilde{L}}{\tilde{L}-N\delta}$ (represented by the dashed line in the figure), are displayed in the figure. The goodness of fit is evaluated using R-squared values, which are 1.000 for both models, indicating a perfect fit between the data and the fitted equations. The estimated discount factor values are $c_1 = 0.9578$ and $c_2 = 0.9576$.

Figure 4: The estimations of threshold of request-volume and occupancy rate



6.4 Performance Evaluation Under Varied Constraints

To comprehensively evaluate the effectiveness of government policy strictness and venue-specific characteristics, we assess performance using three key metrics: (1) the threshold of request-volume, (2) the threshold of occupancy rate, and (3) the maximum achievable occupancy rate. Our analysis examines how these metrics respond to four critical factors: government-mandated maximum allowable occupancy rates, variations in allowable maximum group sizes, different physical distances ($\delta \in \{1, 2\}$) and alternative seat layout configurations.

6.4.1 Government-mandated maximum allowable occupancy rates

Government policies may impose a maximum allowable occupancy rate (ρ^{al}) to enforce stricter public health measures. The effectiveness of this policy depends on its relationship to two key metrics, threshold of occupancy rate (ρ^{th}) and maximum achievable occupancy rate (ρ^{ac}).

When $\rho^{al} < \rho^{th}$, only the occupancy rate requirement is binding. Once the occupancy rate reaches

ρ^{al}, all subsequent requests are rejected. When $\rho^{th} \leq \rho^{al} < \rho^{ac}$, both maximum allowable occupancy rate and social distancing requirements jointly govern seat assignments. Again, once ρ^{al} is reached, further requests are rejected. When $\rho^{al} \geq \rho^{ac}$, the maximum allowable occupancy rate constraint becomes redundant because the occupancy rate will never exceed ρ^{al} under social distancing.

The conclusion can be summarized in Table 2.

Table 2: Effectiveness of requirements

	Social distancing requirement	requirement of ρ^{al}
$\rho^{al} < \rho^{th}$	Ineffective	Effective
$\rho^{th} \leq \rho^{al} < \rho_{ac}$	Effective	Effective
$\rho^{al} \geq \rho^{ac}$	Effective	Ineffective

6.4.2 Different Allowable Maximum Group Sizes and Physical Distances

When M is restricted at 3, given the probability distribution [0.12, 0.5, 0.13, 0.25], we discard the fourth component and normalize the remaining three components to generate a new probability distribution: [0.16, 0.67, 0.17]. Similarly, when $M = 2$, the probability distribution is [0.19, 0.81]. We also consider the impact of different distances. We present the corresponding threshold of request-volume, the threshold of occupancy rate and the maximum achievable occupancy rate in the table below.

Table 3: Impact of Ms and δs

M	δ	q^{th}	ρ^{th}	ρ^{ac}
2	1	74	66.8 %	70.0 %
2	2	54	48.8 %	50.0 %
3	1	68	68.3 %	75.0 %
3	2	53	53.1 %	60.0 %
4	1	57	71.8 %	80.0 %
4	2	47	59.2 %	70.0 %

For fixed δ, as M increases, ρ^{th} increases, while q^{th} decreases according to the estimation. The maximum achievable occupancy rate increases because $\Phi(M, L_j^0, \delta)$ is monotonically increasing in M.

For fixed M, as δ increases from 1 to 2 seats, both q^{th} and ρ^{th} decrease (consistent with our earlier analysis). The maximum achievable occupancy rate decreases since $\Phi(M, L_j^0, \delta)$ is monotonically decreasing in δ.

6.4.3 Alternative Seat Layout configurations

We experiment with several realistic seat layouts selected from a theater seat plan website, choosing five layouts labeled A, B, C, D, and E. Layouts A, D, and E are approximately rectangular, Layout C is a standard rectangular layout, and Layout B is irregular. In these layouts, wheelchair seats and management seats are excluded, while seats with sufficient space for an aisle are treated as new rows. The specific layouts are detailed in EC.3.

The occupancy rate over demand follows the typical pattern of Figure 3. The threshold of request-volume, the threshold of occupancy rate and the maximum achievable occupancy rate are also given in the following table. The maximum achievable occupancy rate can be calculated from Proposition 2.

Table 4: Impact of the layouts

Layout	q^{th}	ρ^{th}	ρ^{ac}
A	36	72.3 %	82.4 %
B	38	75.8 %	84.1 %
C	32	72.8 %	80.0 %
D	43	74.1 %	83.6 %
E	102	72.4 %	81.7 %

Although the layouts may vary in shapes (rectangular or otherwise) and row lengths (long or short), the threshold of occupancy rate and maximum achievable occupancy rate do not exhibit significant differences. Similarly, layouts with varying total seats and rows do not exhibit a clear trend in the threshold of occupancy rate, as estimated based on the analysis.

7 Conclusion

We study the seating management problem under social distancing requirements. Specifically, we first consider the seat planning with deterministic requests problem. To utilize all seats, we introduce the full and largest patterns. Subsequently, we investigate the seat planning with stochastic requests problem. To tackle this problem, we propose a scenario-based stochastic programming model. Then, we utilize the Benders decomposition method to efficiently obtain a seat plan, which serves as a reference for dynamic seat assignment. Last but not least, to address the seat assignment with dynamic requests, we introduce the SPBA policy by integrating the relaxed dynamic programming and the group-type control allocation.

We conduct several numerical experiments to investigate various aspects of our approach. First, we compare SPBA with three benchmark policies: BPC, BLC, and RDPH. Our proposed policy demonstrates superior and more consistent performance relative to these benchmarks. All policies are assessed against the optimal policy derived from a deterministic model with perfect foresight of request arrivals.

Building upon our policies, we further evaluate the impact of implementing social distancing. By introducing the concept of the threshold of request-volume to characterize situations under which social distancing begins to cause loss to an event, our experiments show that the threshold of request-volume depends mainly on the mean of the group size. This lead us to estimate the threshold of request-volume by the mean of the group size.

Our models and analyses are developed for the social distancing requirement on the physical distance and group size, where we can determine an threshold of occupancy rate for any given event in a venue, and a maximum achievable occupancy rate for all events. Sometimes the government also imposes a maximum allowable occupancy rate to tighten the social distancing requirement. This maximum allowable rate is effective for an event if it is lower than the threshold of occupancy rate of the event. Furthermore, the maximum allowable rate will be redundant if it is higher than the maximum achievable rate for all events. The above qualitative insights are stable concerning the tightness of the policy as well as the specific characteristics of various venues.

Future research can be pursued in several directions. First, when seating requests are predetermined, a scattered seat assignment approach can be explored to maximize the distance between adjacent groups when sufficient seating is available. Second, more flexible scenarios could be considered, such as allowing individuals to select seats based on their preferences. Third, research could also investigate scenarios where individuals arrive and leave at different times, adding another layer of complexity to the problem.

References

Adelman, D., 2007. Dynamic bid prices in revenue management. Operations Research 55, 647–661.

Benders, J., 1962. Partitioning procedures for solving mixed-variables programming problems. Numer. Math 4, 238–252.

Bertsimas, D., Popescu, I., 2003. Revenue management in a dynamic network environment. Transportation Science 37, 257–277.

Bitran, G.R., Mondschein, S.V., 1995. An application of yield management to the hotel industry considering multiple day stays. Operations Research 43, 427–443.

Blom, D., Pendavingh, R., Spieksma, F., 2022. Filling a theater during the COVID-19 pandemic. INFORMS Journal on Applied Analytics 52, 473–484.

Bortolete, J.C., Bueno, L.F., Butkeraites, R., et al., 2022. A support tool for planning classrooms considering social distancing between students. Computational and Applied Mathematics 41, 1–23.

CDC, 2020. Social distancing : keep a safe distance to slow the spread. https://stacks.cdc.gov/view/cdc/90522.

Chekuri, C., Khanna, S., 2005. A polynomial time approximation scheme for the multiple knapsack problem. SIAM Journal on Computing 35, 713–728.

Clausen, T., Hjorth, A.N., Nielsen, M., Pisinger, D., 2010. The off-line group seat reservation problem. European Journal of Operational Research 207, 1244–1253.

Deplano, I., Yazdani, D., Nguyen, T.T., 2019. The offline group seat reservation knapsack problem with profit on seats. IEEE Access 7, 152358–152367.

Ferreira, C.E., Martin, A., Weismantel, R., 1996. Solving multiple knapsack problems by cutting planes. SIAM Journal on Optimization 6, 858–877.

Fischetti, M., Fischetti, M., Stoustrup, J., 2023. Safe distancing in the time of COVID-19. European Journal of Operational Research 304, 139–149.

Gallego, G., Van Ryzin, G., 1997. A multiproduct dynamic pricing problem and its applications to network yield management. Operations Research 45, 24–41.

Ghorbani, E., Molavian, H., Barez, F., 2020. 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 .

Goldman, P., Freling, R., Pak, K., Piersma, N., 2002. Models and techniques for hotel revenue management using a rolling horizon. Journal of Revenue and Pricing Management 1, 207–219.

Haque, M.T., Hamid, F., 2022. 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.

Haque, M.T., Hamid, F., 2023. Social distancing and revenue management-A post-pandemic adaptation for railways. Omega 114, 102737.

Jasin, S., Kumar, S., 2012. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. Mathematics of Operations Research 37, 313–345.

Khuri, S., Bäck, T., Heitkötter, J., 1994. The zero/one multiple knapsack problem and genetic algorithms, in: Proceedings of the 1994 ACM symposium on Applied computing, pp. 188–193.

Kleywegt, A.J., Papastavrou, J.D., 1998. The dynamic and stochastic knapsack problem. Operations Research 46, 17–35.

Kleywegt, A.J., Papastavrou, J.D., 2001. The dynamic and stochastic knapsack problem with random sized items. Operations Research 49, 26–41.

Kwag, S., Lee, W.J., Ko, Y.D., 2022. Optimal seat allocation strategy for e-sports gaming center. International Transactions in Operational Research 29, 783–804.

Lewis, R., Carroll, F., 2016. Creating seating plans: a practical application. Journal of the Operational Research Society 67, 1353–1362.

Liu, Q., Van Ryzin, G., 2008. On the choice-based linear programming model for network revenue management. Manufacturing & Service Operations Management 10, 288–310.

Lonely Planet, 2020. Berlin theatre shows what socially-distanced entertainment might look like. https://www.lonelyplanet.com/news/berlin-theatre-socially-distanced-seating.

Martello, S., Toth, P., 1990. Knapsack problems: algorithms and computer implementations. John Wiley & Sons, Inc.

Moore, J.F., Carvalho, A., Davis, G.A., Abulhassan, Y., Megahed, F.M., 2021. Seat assignments with physical distancing in single-destination public transit settings. IEEE Access 9, 42985–42993.

Papastavrou, J.D., Rajagopalan, S., Kleywegt, A.J., 1996. The dynamic and stochastic knapsack problem with deadlines. Management Science 42, 1706–1718.

Pavlik, J.A., Ludden, I.G., Jacobson, S.H., Sewell, E.C., 2021. Airplane seating assignment problem. Service Science 13, 1–18.

Perry, T.C., Hartman, J.C., 2009. An approximate dynamic programming approach to solving a dynamic, stochastic multiple knapsack problem. International Transactions in Operational Research 16, 347–359.

Pisinger, D., 1999. An exact algorithm for large multiple knapsack problems. European Journal of Operational Research 114, 528–541.

Salari, M., Milne, R.J., Delcea, C., Cotfas, L.A., 2022. Social distancing in airplane seat assignments for passenger groups. Transportmetrica B: Transport Dynamics 10, 1070–1098.

Salari, M., Milne, R.J., Delcea, C., Kattan, L., Cotfas, L.A., 2020. Social distancing in airplane seat assignments. Journal of Air Transport Management 89, 101915.

Secomandi, N., 2008. An analysis of the control-algorithm re-solving issue in inventory and revenue management. Manufacturing & Service Operations Management 10, 468–483.

Talluri, K., Van Ryzin, G., 1998. An analysis of bid-price controls for network revenue management. Management Science 44, 1577–1593.

Talluri, K.T., Van Ryzin, G.J., 2006. The Theory and Practice of Revenue Management. Springer Science & Business Media.

Tönissen, D.D., Van den Akker, J., Hoogeveen, J., 2017. Column generation strategies and decomposition approaches for the two-stage stochastic multiple knapsack problem. Computers & Operations Research 83, 125–139.

Van Ryzin, G.J., Talluri, K.T., 2005. An introduction to revenue management, in: Emerging theory, methods, and applications. INFORMS, pp. 142–194.

Vangerven, B., Briskorn, D., Goossens, D.R., Spieksma, F.C., 2022. Parliament seating assignment problems. European Journal of Operational Research 296, 914–926.

WHO, 2020. Advice for the public: Coronavirus disease (COVID-19). https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public.

Williamson, E.L., 1992. Airline network seat inventory control: Methodologies and revenue impacts. Ph.D. thesis. Massachusetts Institute of Technology.

Zhu, F., Liu, S., Wang, R., Wang, Z., 2023. Assign-to-seat: Dynamic capacity control for selling high-speed train tickets. Manufacturing & Service Operations Management 25, 921–938.