Seating Management under Social Distancing

March 23, 2025

Abstract

This study tackles the challenge of seat planning and assignment with social distancing measures. Initially, we analyze seat planning with deterministic requests. Subsequently, we introduce a scenario-based stochastic programming approach to formulate seat planning with stochastic requests. We also investigate the dynamic situation where groups enter a venue and need to sit together while adhering to physical distancing criteria. The seat plan can serve as the basis for the assignment. Combined with relaxed dynamic programming, we propose a dynamic seat assignment policy for either accommodating or rejecting incoming groups. Our method outperforms traditional bid-price and booking-limit control policies. 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: Social Distancing, Scenario-based Stochastic Programming, Seating Management, Dynamic Arrival.

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 Problem (SPDRP), the Seat Planning with Stochastic Requests Problem (SPSRP), and the Seat Assignment with Dynamic Requests Problem (SADRP). In the first problem, SPDRP, 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, SPSRP, 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, SADRP, 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 the SPDRP using Integer Programming and discuss the key characteristics of the optimal seating plan. For the SPSRP, we utilize scenario-based optimization and develop solution approaches based on Benders decomposition. Regarding the SADRP, we employ a dynamic programming-based heuristic approach to determine whether to reject or mark each request as ready-to-accept, followed by a group-type control policy to determine seat assignments for the ready-to-accept groups. This dynamic seat 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 seat planning with deterministic requests (SPDRP), we identify valuable concepts such as the full pattern and the largest pattern, which play a crucial role in developing solutions for the other two problems. Additionally, SPDRP serves as a useful offline benchmark for evaluating the performance of policies in SADRP. The duality analysis in SPDRP also aids in developing the bid-price policy for SADRP. Furthermore, the solution to SPSRP can serve as a reference seat plan for dynamic seat assignment in SADRP.

We investigate the impact of social distancing on revenue loss. To facilitate this analysis, we introduce the concept of the gap request threshold, 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 gap request threshold, it will experience virtually no revenue loss due to social distancing. Our computational experiments demonstrate that the gap request threshold 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 gap point 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 occupancy rate, defined as the occupancy rate at the gap point. The maximum allowable occupancy rate is effective for an event only if it is lower than the event's threshold 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 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 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 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 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 dynamic seat assignment policy to assign seats for incoming groups. 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) 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.

SADRP 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 SADRP, requests also arrive sequentially; however, the sizes of resource requirements and rewards depend on the type of request and are revealed upon arrival. Additionally, SADRP 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 SPSRP in our work, where the number of items is stochastic.

Generally speaking, SADRP relates to the *network revenue management* (NRM) problem, which focuses on deciding whether to accept or reject a request (Gallego and Van Ryzin, 1997). The NRM problem can be fully characterized by a dynamic programming (DP) formulation. 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 approaches have been proposed, such as deriving bid-price or booking-limit controls from static formulations or approximating the value function using simplified structures. Talluri and Van Ryzin (1998) was the first to propose bid-price control policies. Since then, a significant body of literature has focused on deriving refined bid prices and tighter bounds on value functions. In our work, we also consider a bid-price control policy, similar to the certainty equivalence control policy proposed by Bertsimas and Popescu (2003), which directly uses the optimal value from a static model to approximate the initial value function. The seminal contribution to booking-limit control is from Gallego and Van Ryzin (1997), which studied a static model and introduced make-to-stock and make-to-order policies. However, these policies lack flexibility in handling stochastic demand.

Most of the work focuses on individual requests and booking decisions. 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.

Notably, in our model, the supply planned for larger groups can also be utilized by smaller groups. This flexibility stems from our approach, which focuses on group arrivals rather than individual units, distinguishing it from traditional partitioned and nested approaches (Curry, 1990; Van Ryzin and Vulcano, 2008).

Another key characteristic of our study is the importance of seat assignment, which distinguishes it from traditional revenue management.

The assign-to-seat feature introduced by Zhu et al. (2023) further emphasizes the significance of seat assignment. This approach tackles the challenge of selling high-speed train tickets, where each request must be assigned to a specific seat for the entire journey. However, this paper focuses on individual passengers rather than groups, which sets it apart from our research.

Similar group-based characteristics are also observed in multi-day stays in hotel revenue management (Aydin and Birbil, 2018; Bitran and Mondschein, 1995),

3 Seat Planning Problem with Social Distancing

We formally describe the problem of considering social distancing measures in the seat planning process. We first introduce some 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 entail leaving one or more empty seats. Specifically, each group must ensure the empty seat(s) with the adjacent group(s). 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 $L_j = L_j^0 + \delta$. It is a clear one-to-one mapping between the original physical seat plan and the model of seat plan. By incorporating the additional seat(s) and designating certain seat(s) for social distancing, we can integrate social distancing measures into the seat plan 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 planned 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., $|\boldsymbol{h}| = \sum_{i=1}^{M} i h_i$. 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.

The above description can be illustrated by the example in Fig. 1.

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.*

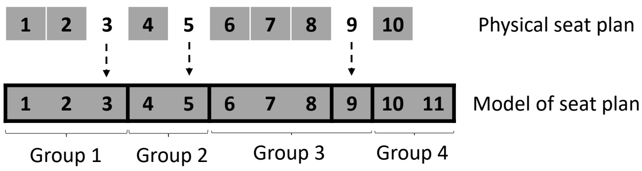


Figure 1: Illustration of Groups with Social Distancing

In the model, 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 $|\boldsymbol{h}| = 7$.

SPDRP can be formulated by an integer programming, where we define x_{ij} to be the number of groups of type i planned in row j.

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

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.

By examining the monotonic ratio between the original group sizes and the adjusted group sizes, we can establish the upper bound of supply corresponding to the optimal solution of the LP relaxation of SPDRP. This is illustrated in Proposition 1 and will be utilized in the bid-price control policy discussed in Appendix A.

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

In other words, the supply corresponding to the optimal solutions for group types i (where $i > \tilde{i}$) exactly matches the demand of group type i. For group types i (where $i < \tilde{i}$), the supply is zero. The supply for group type $\tilde{i}$ is determined by the remaining available seats.

3.2 Seat Planning with Full or Largest Patterns

While the IP (1) is not difficult to solve for any practical problem, the optimal solution reveals some interesting observations regarding the structure of the optimal seat plan. In particular, when some constraints (2) are not tight, most constraints (3) tend to be tight. In words, if some requests are rejected, the seats in each row tend to be fully used. Note that each constraint in (3) corresponds to a pattern of a row. This 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 of a pattern. In a full pattern, the corresponding constraint is mathematically tight. In a maximum pattern, the mathematical tightness is not enforced on the constraint; instead, it is reflected with respect to the objective function in that any other pattern cannot lead to a higher objective function value. Both the full pattern and the maximum pattern imply the efficient use of the 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, $\phi(M, L)$ is non-decreasing with M and L, and non-increasing with 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.

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

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 δ, but not monotone in $L_j^0, \forall j \in \mathcal{N}$.*

Further discussion on the maximum achievable occupancy rate will be provided in Section 6.2.

Example 2. *Consider the given values: $\delta = 1$, $L = 21$, and $M = 4$. The size of the largest pattern can be calculated as $qM + \max\{r-\delta, 0\} = 4\times4+0 = 16$. 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.

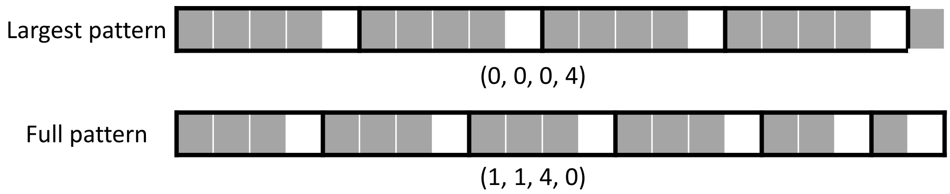


Figure 2: Largest and Full Patterns

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.

We now discuss a related problem: given a feasible seat plan, we seek to obtain a seat plan that satisfies the original requirements of the feasible seat plan while utilizing as many seats as possible. Interestingly, the optimal seat plan consists of either full or largest patterns.

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 aim to obtain a seat plan that is suitable to the stochastic requests. Specifically, we develop the scenario-based stochastic programming (SBSP) to obtain the seat plan with available seats. To take advantage of the well-structured nature of SBSP, we implement Benders decomposition to solve it efficiently. In some cases, solving the integer programming with Benders decomposition remains still computationally prohibitive. Thus, we can consider the LP relaxation first, then construct a seat plan with full or largest patterns to fully utilize all seats.

4.1 Scenario-Based Stochastic Programming Formulation

Now suppose the demand of groups is stochastic, and the stochastic information can be derived from scenarios based on historical data. Let ω index the different scenarios, where each scenario $\omega \in \Omega$. We assume there are $|\Omega|$ possible scenarios, each associated with a specific realization of group requests, represented as $\mathbf{d}_\omega = (d_{1\omega}, d_{2\omega}, \ldots, d_{M,\omega})^\intercal$. Let p_ω denote the probability of any scenario ω, which we assume to be positive. 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} 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 account for the possibility of groups occupying seats originally planned for larger group types when the corresponding supply is insufficient, 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:

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

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

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

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

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

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

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.

By reformulating the objective function, we have

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

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 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, the 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 Solve SBSP by Benders Decomposition

We reformulate SBSP problem into a master problem and a subproblem. The iterative process of solving the master problem and the subproblem is known as Benders decomposition Benders (1962). The solution obtained from the master problem serves as input for the subproblem, while the subproblem's solutions help refine the master problem by adding constraints. This iterative process improves the overall solution until convergence is achieved. To accelerate the solving process, we derive a closed-form solution for the subproblem. Subsequently, we obtain the solution to the LP relaxation of SBSP problem through a constraint generation approach.

4.2.1 Reformulation

We write SBSP in a matrix form to apply the Benders decomposition technique. Let $\mathbf{n}=(n_1,\dots,n_M)^\intercal$ represent the vector of seat sizes for each group type, where n_i denotes the size of seats taken by group type i. Let $\mathbf{L}=(L_1,\dots,L_N)^\intercal$ represent the vector of row sizes, where L_j denotes the size of row j as defined previously. The constraint (9) can be expressed as $\mathbf{x}^\intercal\mathbf{n}\le\mathbf{L}$. This constraint ensures that the total size of seats occupied by each group type, represented by $\mathbf{x}^\intercal \mathbf{n}$, does not exceed the available row sizes $\mathbf{L}$. We can use the product $\mathbf{x1}$ to indicate the supply of group types, where $\mathbf{1}$ is a column vector of size N with all elements equal to 1.

The linear constraints associated with scenarios, denoted by constraints (7) and (8), can be expressed in matrix form as:

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

where $\mathbf{V} = [\mathbf{W},\ \mathbf{I}]$.

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

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

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

The size of the deterministic equivalent formulation increases with the size of the scenario set, rendering directly solving it computationally infeasible. To overcome the difficulty, we reformulate the problem to apply Benders decomposition approach. 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 SBSP problem can be expressed as below,

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal \mathbf{x} + z(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x}^\intercal \mathbf{n} \leq \mathbf{L} \\ & \mathbf{x} \in \mathbb{N}^{M \times N}, \end{aligned} \tag{10}$$

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

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

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

$$\begin{aligned} z_\omega(\mathbf{x}) := \max \quad & \mathbf{f}^\intercal \mathbf{y}_\omega \\ \text{s.t.} \quad & \mathbf{V}\mathbf{y}_\omega = \mathbf{d}_\omega - \mathbf{x1} \\ & \mathbf{y}_\omega \geq 0. \end{aligned} \tag{11}$$

The efficiency of solving problem (10) hinges on the method used to solve problem (11). Next, we will demonstrate how to solve problem (11) efficiently.

4.2.2 Solve The Subproblem

Notice that the feasible region of the dual of problem (11) remains unaffected by $\mathbf{x}$. This observation provides insight into the properties of this problem. Let $\boldsymbol{\alpha}_\omega = (\alpha_{1\omega}, \alpha_{2\omega}, \ldots, \alpha_{M,\omega})^\intercal$ denote the vector of dual variables. For each ω, we can form its dual problem, which is

$$\begin{aligned} \min \quad & \boldsymbol{\alpha}_\omega^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \\ \text{s.t.} \quad & \boldsymbol{\alpha}_\omega^\intercal \mathbf{V} \geq \mathbf{f}^\intercal \end{aligned} \tag{12}$$

Lemma 1. *The feasible region of problem (12) is nonempty and bounded. Furthermore, all the extreme points of the feasible region are integral.*

Let $\mathbb{P}$ indicate the feasible region of problem (12). According to Lemma 1, the optimal value of the problem (11), $z_\omega(\mathbf{x})$, is finite and can be achieved at extreme points of $\mathbb{P}$. Let $\mathcal{O}$ be the set of all extreme points of $\mathbb{P}$. Then, we have $z_\omega(\mathbf{x}) = \min_{\boldsymbol{\alpha}_\omega \in \mathcal{O}} \boldsymbol{\alpha}_\omega^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1})$.

Alternatively, $z_\omega(\mathbf{x})$ is the largest number z_ω such that $\boldsymbol{\alpha}_\omega^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_w, \forall \boldsymbol{\alpha}_\omega \in \mathcal{O}$. We use this characterization of $z_w(\mathbf{x})$ in problem (10) and conclude that problem (10) can thus be put in the form by setting z_w as the variable:

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} p_\omega z_\omega \\ \text{s.t.} \quad & \mathbf{x}^\intercal \mathbf{n} \leq \mathbf{L} \\ & \boldsymbol{\alpha}_\omega^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega, \forall \boldsymbol{\alpha}_\omega \in \mathcal{O}, \forall \omega \\ & \mathbf{x} \in \mathbb{N}^{M \times N} \end{aligned} \tag{13}$$

Before applying Benders decomposition to solve problem (13), it is important to address the efficient computation of the optimal solution to problem (12). When $\mathbf{x}^*$ is given, $\mathbf{y}_\omega$ can be obtained from equation (5). Let $\alpha_{0,\omega} = 0$ for each ω, then we have Proposition 5.

Proposition 5. *The optimal solutions to problem (12) are given by*

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

We can obtain $\alpha_{i\omega}$ through a forward calculation, iterating from $\alpha_{1\omega}$ to $\alpha_{M\omega}$. In practice, we choose $\alpha_{i\omega} = \alpha_{i-1,\omega} + 1$ when $y_{i\omega}^-, y_{i\omega}^+$ satisfy the third condition in Proposition 5.

Instead of solving problem (12) directly, we can obtain the values of $\alpha_{i\omega}$ by performing a forward calculation from $\alpha_{1\omega}$ to $\alpha_{M\omega}$ and choose $\alpha_{i\omega} = \alpha_{i-1,\omega} + 1$ when calculate $\alpha_{i\omega}$.

4.2.3 Constraint Generation

Due to the computational infeasibility of solving problem (13) with an exponentially large number of constraints, it is a common practice to use a subset, denoted as $\mathcal{O}^t$, to replace $\mathcal{O}$ in problem (13).

This results in a modified problem known as the Restricted Benders Master Problem (RBMP). To find the optimal solution to problem (13), we employ the technique of constraint generation. It involves iteratively solving the RBMP and incrementally adding more constraints until the optimal solution to problem (13) is obtained.

We can conclude that the RBMP will have the form:

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

Given the initial $\mathcal{O}^t$, we can have the solution $\mathbf{x}^*$ and $\mathbf{z}^* = (z_1^*, \ldots, z_{|\Omega|}^*)$. Then $c^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega z_\omega^*$ is an upper bound of problem (15). When $\mathbf{x}^*$ is given, the optimal solution, $\tilde{\boldsymbol{\alpha}}_\omega$, to problem (12) can be obtained according to Proposition 5. Let $\tilde{z}_\omega = \tilde{\boldsymbol{\alpha}}_\omega(d_\omega - \mathbf{x}^*\mathbf{1})$, then $(\mathbf{x}^*, \tilde{\mathbf{z}})$ is a feasible solution to problem (15) because it satisfies all the constraints. Thus, $\mathbf{c}^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega \tilde{z}_\omega$ is a lower bound of problem (13).

If for every scenario ω, the optimal value of the corresponding problem (12) is larger than or equal to z_ω^*, which means all contraints are satisfied, then we have an optimal solution, $(\mathbf{x}^*, \mathbf{z}^*)$, to problem (13). However, if there exists at least one scenario ω for which the optimal value of problem (12) is less than z_ω^*, indicating that the constraints are not fully satisfied, we need to add a new constraint $(\tilde{\boldsymbol{\alpha}}_\omega)^\intercal (\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega$ to RBMP.

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

Proposition 6. *RBMP is bounded when there is at least one constraint for each scenario.*

From Proposition 6, we can set $\boldsymbol{\alpha}_\omega = \mathbf{0}$ initially. Notice that only contraints are added in each iteration, thus UB is decreasing monotone over iterations. Then we can use $UB - LB < \epsilon$ to terminate the algorithm.

Algorithm 1: Benders Decomposition

```
Initialize α_ω = 0, ∀ω, and let LB ← 0, UB ← ∞;
while UB − LB > ε do
    Solve problem (15) and obtain an optimal solution (x*, z*);
    UB ← cᵀx* + Σ_{ω∈Ω} p_ω z*_ω;
    for ω = 1, ..., |Ω| do
        Obtain α̃_ω according to Proposition 5;
        z̃_ω = (α̃_ω)ᵀ(d_ω − x*1);
        if z̃_ω < z*_ω then
            Add one new constraint, (α̃_ω)ᵀ(d_ω − x1) ≥ z_ω, to problem (15);
        end
    end
    LB ← cᵀx* + Σ_{ω∈Ω} p_ω z̃_ω;
end
```

However, solving problem (15) directly can be computationally challenging in some cases, so we practically first obtain the optimal solution to the LP relaxation of problem (10). Then, we generate a seat plan from this solution.

4.3 Obtain Seat Plan with Full or Largest Patterns

We may obtain a fractional optimal solution when we solve the LP relaxation of problem (10). This solution represents the optimal allocations of groups to seats but may involve fractional values, indicating partial assignments. 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 last step.

Let the optimal solution to the LP relaxation of problem (15) be $\mathbf{x}^*$. Aggregate $\mathbf{x}^*$ to the number of each group type, $\tilde{X}_i = \sum_j x^*_{ij}, \forall i \in \mathbf{M}$. Solve the SPDRP with $\boldsymbol{d} = \tilde{\boldsymbol{X}}$ to obtain the optimal solution, $\tilde{\mathbf{x}}$, and the corresponding pattern, $\boldsymbol{H}$, then generate the seat plan by problem (4) with $\boldsymbol{H}$.

Algorithm 2: Seat Plan Construction

1. Solve the LP relaxation of SBSP, and obtain an optimal solution $\mathbf{x}^*$;
2. Solve SPDRP with $d_i = \sum_j x^*_{ij}, i \in \mathbf{M}$, and obtain an optimal solution $\tilde{\mathbf{x}}$ and the corresponding pattern, $\tilde{\boldsymbol{H}}$;
3. 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, and once seats are assigned to a group, they cannot be altered or reassigned to other requests.

To model this problem, we adopt 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 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 $u^t_{i,j}$, where $u^t_{i,j} = 1$ 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

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

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 maximal expected revenue earned by the best decisions regarding group seat assignments at the beginning of period t, given remaining capacity $\mathbf{L}$. Then, the dynamic programming formula 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.

Initially, we have the current remaining capacity vector denoted as $\mathbf{L}^0 = (L_1, L_2, \ldots, L_N)$. Our objective is to make group assignments that maximize the total expected revenue during the horizon from period 1 to T which is represented by $V^1(\mathbf{L}^0)$.

Solving the dynamic programming problem described in equation (16) can be challenging due to the curse of dimensionality, which arises when the problem involves a large number of variables or states. To mitigate this complexity, we aim to develop a heuristic method for assigning arriving groups. In our approach, we begin by generating a seat plan, as outlined in Section 4. This seat plan acts as a foundation for the seat assignment.

5.1 DP-Based Heuristic

To simplify the complexity of DP (16), we consider a simplified version by relaxing all rows to a single row with the same total capacity, denoted as $\tilde{L} = \sum_{j=1}^{N} L_j$. Using the relaxed dynamic programming approach, we can determine the seat assignment decisions for each group arrival. Let u denote the decision, where $u_i^t = 1$ if we accept a request of type i in period t, $u_i^t = 0$ otherwise. Similarly, the DP with one row can be expressed as:

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

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

Note that we must first verify whether the group can be accommodated with the available seats.

Specifically, if the size of the arriving group exceeds the maximum remaining capacity across all rows, the group must be rejected. Once a group is accepted, the next step is to determine where to assign the seats. However, in the absence of a specific seat plan, there are no predefined rules to guide this assignment process. To address this, we adopt a rule similar to the Best Fit rule (Johnson, 1974). Specifically, the group is assigned to the row with the smallest remaining seats that can still accommodate the group.

This policy is stated in the following algorithm.

Algorithm 3: DP-based Heuristic Algorithm

Calculate $V^t(l)$ by (17), $\forall t = 2, \dots, T; \forall l = 1, 2, \dots, l^1 = \tilde{L}$;
for $t = 1, \dots, T$ **do**
  Observe a request of group type i;
  if $\max_{j \in \mathcal{N}} L_j^t \geq n_i$ *and* $V^{t+1}(l^t) \leq V^{t+1}(l^t - n_i) + i$ **then**
    Set $k = \arg\min_{j \in \mathcal{N}} \{L_j^t | L_j^t \geq n_i\}$ and break ties arbitrarily;
    Assign the group to row k, let $L_k^{t+1} \leftarrow L_k^t - n_i$, $l^{t+1} \leftarrow l^t - n_i$;
  else
    Reject the group and let $L_k^{t+1} \leftarrow L_k^t$, $l^{t+1} \leftarrow l^t$;
  end
end

Since this policy does not guide a more effective assignment approach, we proceed with the assignment based on the seat plan strategy.

5.2 Assignment Based on Seat Plan

In this section, we assign groups based on a seat plan that incorporates full or largest patterns. When a request of group type i is ready to be assigned using the DP-based heuristic, we allocate seats if the supply for type i satisfies $X_i > 0$. If $X_i = 0$, we apply the group-type control to determine whether the group can be assigned to a specific row. Addtionally, we discuss the tie-breaking rules in section 5.2.2 for selecting a specific row when multiple options are available. Finally, we outline the conditions under which the seat plan should be regenerated.

In the following part, we will refer to the whole policy as Dynamic Seat Assignment (DSA).

5.2.1 Group-Type Control

The group-type control policy is designed to determine the appropriate group type when the supply for the arriving group is insufficient, thereby aiding in the decision of whether to assign the group and selecting the suitable row during seat assignment. This policy evaluates whether the supply allocated for larger groups can be utilized to accommodate the arriving group, based on the current seat plan.

We balance the trade-off between preserving the current seat plan for potential future requests and accepting the current request. To make this decision, we calculate the expected number of acceptable individuals for both options and compare these values to determine the optimal strategy.

Specifically, suppose the supply is $[X_1, \dots, X_M]$ at period t, the number of remaining periods is

$(T-t)$. When $X_i = 0$ for the request of group type i, we can use one supply of group type $\hat{i}$ to accept a group type i for any $\hat{i} = i+1, \ldots, M$. In this case, when $\hat{i} = i+1, \ldots, i+\delta$, the expected number of accepted individuals is i and the remaining seats beyond the accepted group, which is $\hat{i} - i$, will be wasted. When $\hat{i} = i+\delta+1, \ldots, M$, the rest $(\hat{i} - i - \delta)$ seats can be provided for one group type $(\hat{i} - i - \delta)$ with δ seats of social distancing. Let $D^t_{\hat{i}}$ be the random variable that indicates the number of group type $\hat{i}$ in the future t periods. The expected number of accepted people is $i + (\hat{i} - i - \delta)P(D^{T-t}_{\hat{i}-i-\delta} \geq X_{\hat{i}-i-\delta} + 1)$, where $P(D^{T-t}_{i} \geq X_i)$ is the probability that the demand of group type i in $(T-t)$ periods is no less than X_i, the remaining supply of group type i. Thus, the term, $P(D^{T-t}_{\hat{i}-i-\delta} \geq X_{\hat{i}-i-\delta} + 1)$, indicates the probability that the demand of group type $(\hat{i} - i - \delta)$ in $(T-t)$ periods is no less than its current remaining supply plus 1.

Similarly, when we retain the supply of group type $\hat{i}$ by rejecting the group type i, the expected number of accepted individuals is $\hat{i}P(D^{T-t}_{\hat{i}} \geq X_{\hat{i}})$. The term, $P(D^{T-t}_{\hat{i}} \geq X_{\hat{i}})$, indicates the probability that the demand of group type $\hat{i}$ in $(T-t)$ periods is no less than its current remaining supply.

Let $d^t(i, \hat{i})$ be the difference of the expected number of accepted individuals between acceptance and rejection in the group type i that occupies seats of $(\hat{i} + \delta)$ size in period t. Then we have

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

The decision is to choose $\hat{i}$ with the largest difference. For all $\hat{i} = i+1, \ldots, M$, we obtain the largest $d^t(i, \hat{i})$, denoted as $d^t(i, \hat{i}^*)$. If $d^t(i, \hat{i}^*) \geq 0$, we will assign the group type i in $(\hat{i}^* + \delta)$-size seats. Otherwise, reject the group.

Although the group-type control policy can help us determine whether to assign and narrow down the row selection options in the assignment, we still need to discuss the tie-breaking rules to determine a specific row.

5.2.2 Tie-Breaking Rules for Determining A Specific Row

A tie occurs when there are serveral rows to accommodate the group. To determine the appropriate row for seat assignment, we can apply the following tie-breaking rules among the possible options. Suppose one group type i arrives, the current seat plan is $\boldsymbol{H} = \{\boldsymbol{h}_1; \ldots; \boldsymbol{h}_N\}$, the corresponding supply is $\boldsymbol{X}$. Let $\beta_j = L_j - \sum_i n_i H_{ji}$ represent the remaining number of seats in row j after considering the seat allocation for the groups. When $X_i > 0$, we assign the group to row $k \in \arg\min_j \{\beta_j\}$ such that the row can be filled as much as possible. When $X_i = 0$ and the group is ready to take the seats designated for group type $\hat{i}, \hat{i} > i$, we assign the group to a row $k \in \arg\max_j \{\beta_j | H_{j\hat{i}} > 0\}$. That can help reduce the number of rows that are not full. When there are multiple ks available, we can choose one arbitrarily. This rule in both scenarios prioritizes filling rows and leads to better seat management.

Combining the group-type control strategy with the evaluation of relaxed DP values, we obtain a comprehensive decision-making process within a single period. This integrated approach enables us to make informed decisions regarding the acceptance or rejection of incoming requests, as well as determine the appropriate row for the assignment when acceptance is made.

5.2.3 Regenerate 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 decreases from one to zero. This ensures that the largest groups are 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 achieve real-time seat assignment while reducing the frequency of planning updates, thereby balancing efficiency and effectiveness.

The algorithm is shown below.

Algorithm 4: Dynamic Seat Assignment

Obtain $\boldsymbol{X}$ and $\boldsymbol{H}$, calculate $V^t(l)$ by (17), $\forall t = 2, \ldots, T; \forall l = 1, 2, \ldots, l^1 = \tilde{L}$;
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 $k = \arg\min_j \{L_j^t - \sum_i n_i H_{ji}^t | H_{ji}^t > 0\}$, break ties arbitrarily;
            Assign group type i in row k, let $L_k^{t+1} \leftarrow L_k^t - n_i$, $H_{ki} \leftarrow H_{ki} - 1$, $X_i \leftarrow X_i - 1$;
            if $i = M$ *and* $X_M = 0$ **then**
                Generate seat plan $\boldsymbol{H}$ from Algorithm 2, update the corresponding $\boldsymbol{X}$;
            end
        else
            Calculate $d^t(i, \hat{i}^*)$;
            if $d^t(i, \hat{i}^*) \geq 0$ **then**
                Set $k = \arg\max_j \{L_j^t - \sum_i n_i H_{ji}^t | H_{j\hat{i}^*}^t > 0\}$, 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 $\boldsymbol{H}$ from Algorithm 2, update the corresponding $\boldsymbol{X}$;
            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 carried out several experiments, including analyzing the performances of different policies, evaluating the impact of implementing social distancing, and comparing different layouts, Ms and social distances. In the experiments, we set the following parameters.

The default setting in the experiments is as follows, $\delta = 1$ and $M = 4$. The default seat layout consists of 10 rows, each with the same size of 21. Different realistic layouts, group sizes and social distances are also explored. We simulate the arrival of exactly one group in each period, i.e., $p_0 = 0$. The average number of individuals per period, denoted as γ, can be expressed as $\gamma = \sum_{i=1}^{M} i p_i$. 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: $D1$:

$[0.18, 0.7, 0.06, 0.06]$ and $D2 : [0.2, 0.8, 0, 0]$, $D3 = [0.34, 0.51, 0.07, 0.08]$ and $D4 : [0.12, 0.5, 0.13, 0.25]$. The first two distributions, $D1$ and $D2$, are experimented in Blom et al. (2022). Here, $D1$ represents the statistical distribution of group sizes, while $D2$ reflects a restricted situation where groups of more than 2 people are not allowed. The other two distributions, $D3$ and $D4$, are derived from real-world movie data. The specific procedure is detailed in Appendix C. We use $D4$ as the default probability distribution in the other experiments.

6.1 Performances of Different Policies

We compare the performance of four assignment policies against the optimal one, which is derived by solving the deterministic model after observing all arrivals. The policies under evaluation are DSA, DP-based heuristic (DPBH), bid-price control (BPC), and booking-limit control (BLC) policies.

Parameters Description

The following table presents the performance results of four different policies: DSA, DPBH, BPC, BLC, which stand for dynamic seat assignment, dynamic programming-based heuristic, bid-price control and booking-limit control policies, respectively. The procedures of the BPC and BLC policies are detailed in Appendix A. Performance is evaluated by comparing the ratio of the number of accepted individuals under each policy to that under the optimal policy, which assumes complete knowledge of all incoming groups before making seat assignments.

Table 1: Performances of Different Policies

Distribution	T	DSA (%)	DPBH (%)	BPC (%)	BLC (%)
D1	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
D2	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
D3	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
D4	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

We can find that DSA is better than DPBH, BPC and BLC policies consistently. DPBH and BPC policies can only decide to accept or deny, cannot decide which row to assign the group to. BLC policy does not consider using more seats to meet the demand of one group.

The performance of DSA, DP-based heuristic, and bid-price policies follows a pattern where it initially decreases and then gradually improves as T increases. When T is small, the demand for capacity 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 plan, 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 booking-limit 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 distributions listed, DSA performs more stably and consistently for the same demand. In contrast, the performance of DPBH and BPC fluctuates more significantly.

6.2 Impact of Social Distancing

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

The *gap request threshold* $\tilde{Q}$ is defined as

$$\tilde{Q} = \sum_{i \in \mathcal{M}} p_i \cdot \max \left\{ T \,\middle|\, \bar{E}(T;\delta) + 1 > \bar{E}(T;\delta = 0) \right\}, \quad \delta \in \{1, 2\},$$

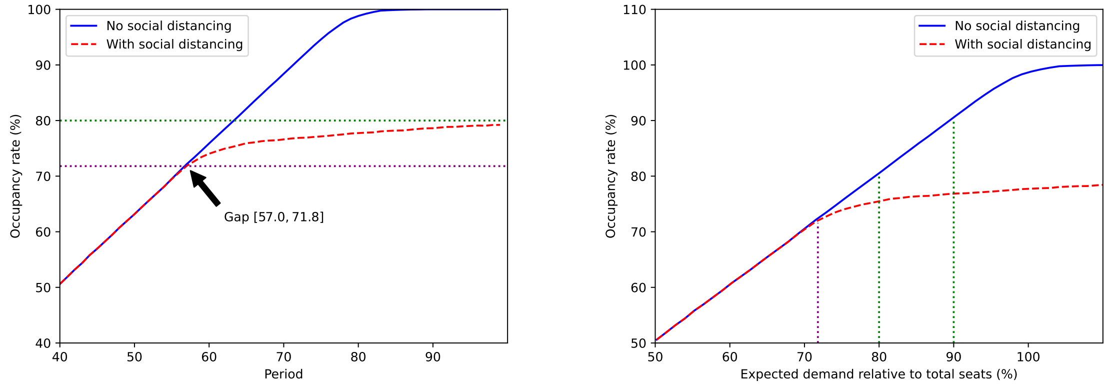
where $\bar{E}(T;\delta)$ and $\bar{E}(T;\delta = 0)$ denotes the average number of accepted individuals by DSA with social distancing level δ and without social distancing, respectively. Here, the maximization is performed over T while keeping all other parameters constant. Intuitively, the gap request threshold is the expected maximum number of requests where, on average, the loss in accepted individuals due to social distancing does not exceed one. In the specific case where $p_0 = 0$ (i.e., only one request arrives in every period), it represents the maximum number of requests that can be accommodated while keeping the average loss below one.

The occupancy rate corresponding to the gap request threshold is referred to as the *threshold occupancy rate*. This rate represents the maximum occupancy rate when the difference in the number of accepted 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 $\frac{\sum_{j \in \mathcal{N}} \phi(M, L_j^0; \delta)}{\sum_{j \in \mathcal{N}} L_j^0}$, as introduced in Section 3.2.

We examine the impact of social distancing when implementing DSA under varying levels of demand. Specifically, we test a one-seat social distancing requirement ($\delta = 1$) against a benchmark scenario with no social distancing ($\delta = 0$). 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 shows the occupancy rate under different demand levels.

Figure 3(a) displays occupancy rate over period. The gap point is 57, the threshold occupancy rate is 71.8%. For the social distancing situation, when the largest pattern is realized in each row, the maximum achievable occupancy rate is given by 80%.



(a)

(b)

Figure 3: The occupancy rate over demand

Figure 3(b) displays occupancy rate over expected demand. When the expected demand is less than 71.8%, the social distancing requirement will not cause the loss; when the expected demand is larger than 71.8%, the difference between the number of accepted individuals with and without social distancing requirements becomes more pronounced.

Requirement of Maximum Allowable Occupancy Rate

Sometimes, policies impose a maximum allowable occupancy rate to enforce stricter measures. This rate becomes redundant if it exceeds the maximum achievable rate. When the maximum allowable rate is below the threshold occupancy rate, only the occupancy rate requirement is effective, rendering the social distancing requirement irrelevant. However, when the maximum allowable rate falls between the threshold occupancy rate and the maximum achievable rate, both the occupancy rate and social distancing requirements jointly influence seat assignments.

In the above example, the maximum achievable rate is 80%, implying that when the maximum allowable rate exceeds 80%, it has no effect. The threshold occupancy rate is 71.8%, so when the maximum allowable rate is below 71.8%, only the occupancy rate requirement is effective. When the maximum allowable rate is between 71.8% and 80%, both the occupancy rate and social distancing requirements jointly determine seat assignments.

6.3 Estimation of Gap Points and Occupancy Rates

To estimate the gap point, 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. Recall that $\tilde{L} = \sum_j L_j$ denotes the total number of seats, and γ represents the average number of individuals in each period. Then, we can derive the inequality $T \leq \frac{\tilde{L}}{\gamma+\delta}$. Therefore, the upper bound for the expected maximal period is given by $T' = \frac{\tilde{L}}{\gamma+\delta}$.

Assuming that all arrivals within T' periods are accepted and fill all the available seats, the threshold occupancy rate can be calculated as $\frac{\gamma T'}{(\gamma+\delta)T'-N\delta} = \frac{\gamma}{\gamma+\delta}\frac{\tilde{L}}{\tilde{L}-N\delta}$. However, it is important to note that the actual maximal period will be smaller than T' because it is nearly impossible to accept groups to fill all seats exactly. To estimate the gap point when applying DSA, we can use $y_1 = c_1\frac{\tilde{L}}{\gamma+\delta}$, where c_1 is a discount factor compared to the ideal assumption. Similarly, we can estimate the threshold occupancy rate as $y_2 = c_2\frac{\gamma}{\gamma+\delta}\frac{\tilde{L}}{\tilde{L}-N\delta}$, where c_2 is a discount factor for the occupancy rate compared to the ideal scenario.

To analyze the relation between the gap point, the threshold occupancy rate and γ, we conducted a study using a sample of 200 probability distributions. The figure below shows the gap point and the threshold occupancy rate as functions of γ, along with their corresponding estimations.

We applied an Ordinary Least Squares (OLS) model to fit the data and estimate the parameter values. The resulting fitted equations, $y_1 = \frac{c_1\tilde{L}}{\gamma+\delta}$ (represented by the solid line in the figure) and $y_2 = c_2\frac{\gamma}{\gamma+\delta}\frac{\tilde{L}}{\tilde{L}-N\delta}$ (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 $c_2 = 0.9576$.

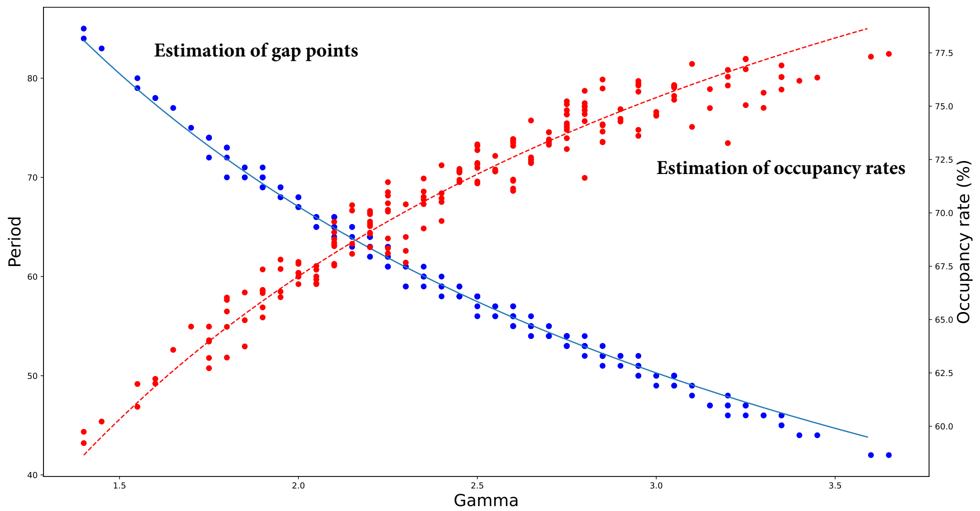


Figure 4: The estimations of gap points and occupancy rates

Based on the above analysis, we also explore the results of different layouts, different group sizes and different social distances. Since the figure about the occupancy rate over demand is similar to Figure 3, we only use three metrics to show the results: the gap point and the threshold occupancy rate, the maximum achievable occupancy rate.

Different Layouts

We experiment with several realistic seat layouts selected from the theater seat plan website, https://www.lcsd.gov.hk/en/ticket/seat.html. We select five places, Hong Kong Film Archive Cinema, Kwai Tsing Theatre Transverse Stage, Sai Wan Ho Civic Centre, Sheung Wan Civic Centre, Ngau Chi

Wan Civic Centre, represented as HKFAC, KTTTS, SWHCC, SWCC, NCWCC respectively. HKFAC, SWCC, NCWCC, are approximately rectangular layouts, SWHCC is a standard rectangular layout. While KTTTS is an irregular layout. In these layouts, wheelchair seats and management seats are excluded, while seats with sufficient space for an aisle are treated as new rows.

The occupancy rate over demand follows the typical pattern of Figure 3. The gap point, the threshold 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 2: Gap points and occupancy rates of the layouts

Seat Layout	Gap point	Threshold occupancy rate	Maximum achievable occupancy rate
HKFAC	36	72.3 %	82.4 %
KTTTS	38	75.79 %	84.1 %
SWHCC	32	72.83 %	80 %
SWCC	43	74.07 %	83.6 %
NCWCC	102	72.37 %	81.7 %

Although the layouts may vary in shapes (rectangular or otherwise) and row lengths (long or short), the threshold occupancy rate and maximum achievable occupancy rate do not exhibit significant differences. This can be explained as follows.

Similarly, layouts with varying total seats and rows do not exhibit a clear trend in the threshold occupancy rate, as estimated based on the analysis.

Different Allowable Largest Group Sizes

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 present the gap point, the threshold occupancy rate and the maximum achievable occupancy rate in the table below.

Table 3: Gap points and occupancy rates of Ms

M	Gap point	Threshold occupancy rate	Maximum achievable occupancy rate
2	74	66.88 %	70 %
3	69	69.03 %	75 %
4	57	71.82 %	80 %

Meanwhile, based on the estimation of the gap point and threshold occupancy rate, as M increases, the threshold occupancy rate increases, while the gap point decreases.

Different Social Distances

The following figure illustrates the occupancy rate over period with different social distances.

The gap point, the threshold occupancy rate and the maximum achievable occupancy rate are shown in the table below.

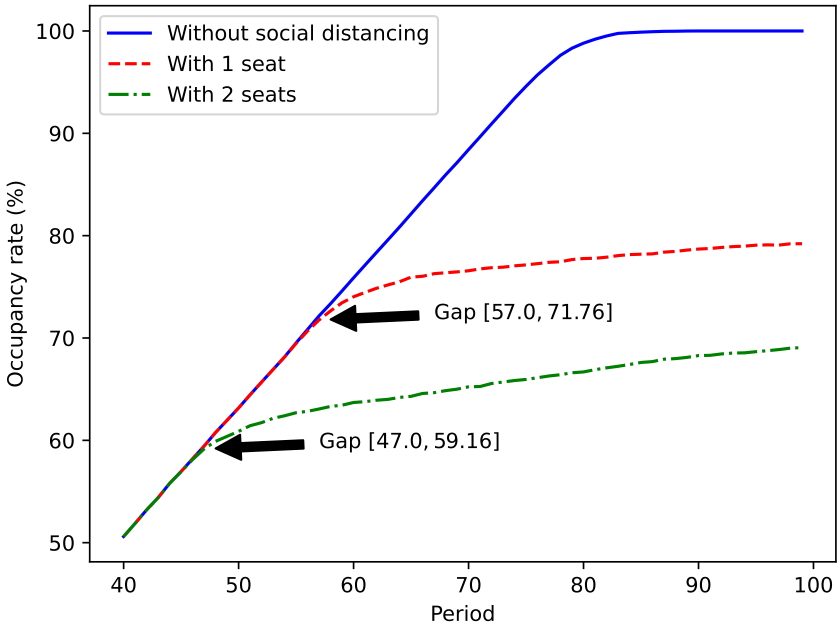


Figure 5: The occupancy rate over demand for different social distances

Table 4: Gap points and occupancy rates of δs

δ	Gap point	Threshold occupancy rate	Maximum achievable occupancy rate
1	57	71.76 %	80 %
2	47	59.16 %	70 %

When the social distance increases from 1 seat to 2 seats, the gap point and threshold occupancy rate decrease, as estimated in the analysis above, and the maximum achievable occupancy rate also decreases according to its definition.

7 Conclusion

We study the seating management problem under social distancing requirement. 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, we introduce an approach to address the problem of dynamic seat assignment by integrating relaxed dynamic programming and a group-type control policy.

We conduct several numerical experiments to investigate various aspects of our approach. First, we analyze different policies for dynamic seat assignment. In terms of dynamic seat assignment policies, we consider the bid-price control, booking-limit control, dynamic programming-based heuristic policies. Comparatively, our proposed policy exhibited superior and consistent performance.

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

Our models and analysis are developed for the social distancing requirement on the physical distance and group size, where we can determine an threshold 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 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

Aydin, N., Birbil, S.I., 2018. Decomposition methods for dynamic room allocation in hotel revenue management. European Journal of Operational Research 271, 179–192.

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. Keep a safe space. 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.

Curry, R.E., 1990. Optimal airline seat allocation with fare classes nested by origins and destinations. Transportation Science 24, 193–204.

Dantzig, G.B., 1957. Discrete-variable extremum problems. Operations Research 5, 266–288.

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 .

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.

Johnson, D.S., 1974. Fast algorithms for bin packing. Journal of Computer and System Sciences 8, 272–314.

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.

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.

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., Vulcano, G., 2008. Simulation-based optimization of virtual nesting controls for network revenue management. Operations Research 56, 865–880.

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. https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public.

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.

A Policies for Dynamic Situations

Bid-Price Control Policy

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 of LP relaxation of SPDRP is:

$$\begin{aligned} \min \quad & \sum_{i=1}^{M} d_i z_i + \sum_{j=1}^{N} L_j \beta_j \\ \text{s.t.} \quad & z_i + \beta_j n_i \geq (n_i - \delta), \quad i \in \mathcal{M}, j \in \mathcal{N} \\ & z_i \geq 0, i \in \mathcal{M}, \beta_j \geq 0, j \in \mathcal{N}. \end{aligned} \tag{18}$$

In (18), β_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 no less than the sum of the bid prices of the seats it uses. Thus, if $i - \beta_j n_i \geq 0$, 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 2. *The optimal solution to problem (18) is given by* $z_1, \ldots, z_{\tilde{i}} = 0$, $z_i = \frac{\delta(n_i - n_{\tilde{i}})}{n_{\tilde{i}}}$ *for* $i = \tilde{i}+1, \ldots, M$ *and* $\beta_j = \frac{n_{\tilde{i}} - \delta}{n_{\tilde{i}}}$ *for all* j.

The bid-price decision can be expressed as $i - \beta_j n_i = i - \frac{n_{\tilde{i}} - \delta}{n_{\tilde{i}}} n_i = \frac{\delta(i - \tilde{i})}{n_{\tilde{i}}}$. When $i < \tilde{i}$, $i - \beta_j n_i < 0$. When $i \geq \tilde{i}$, $i - \beta_j n_i \geq 0$. This implies that group type i greater than or equal to $\tilde{i}$ will be accepted if the capacity allows. However, it should be noted that β_j does not vary with j, which means the bid-price control cannot determine the specific row to assign the group to. In practice, groups are often assigned arbitrarily based on availability when the capacity allows, which may result in the empty seats.

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

Algorithm 5: Bid-Price Control

for $t = 1, \ldots, T$ **do**
    Observe a request of group type i;
    Solve the LP relaxation of SPDRP with $\boldsymbol{d}^t = (T - t) \cdot \boldsymbol{p}$ and $\mathbf{L}^t$;
    Obtain $\tilde{i}$ such that the aggregate optimal solution is $x e_{\tilde{i}} + \sum_{i=\tilde{i}+1}^{M} d_i e_i$;
    if $i \geq \tilde{i}$ *and* $\max_{j \in \mathcal{N}} L_j^t \geq n_i$ **then**
        Set $k = \arg\min_{j \in \mathcal{N}} \{L_j^t | L_j^t \geq n_i\}$;
        Break ties arbitrarily;
        Assign the group to row k, let $L_k^{t+1} \leftarrow L_k^t - n_i$;
    else
        Reject the group;
    end
end

Booking-Limit Control Policy

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

In this policy, we solve SPDRP with 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.

Algorithm 6: Booking-Limit Control

for $t = 1, \ldots, T$ **do**
  Observe a request of group type i;
  Solve SPDRP with $\boldsymbol{d}^t = (T - t) \cdot \boldsymbol{p}$ and $\mathbf{L}^t$;
  Obtain the seat plan, $\boldsymbol{H}$, and the aggregate optimal solution, $\mathbf{X}$;
  if $X_i > 0$ **then**
    Set $k = \arg\min_j \{L_j^t - \sum_i n_i H_{ji} | H_{ji} > 0\}$;
    Break ties arbitrarily;
    Assign the group to row k, let $L_k^{t+1} \leftarrow L_k^t - n_i$;
  else
    Reject the group;
  end
end

B Proofs

Proof of Proposition 1

First, we regard this problem as a special case of the Multiple Knapsack Problem (MKP), then we consider the LP relaxation of this problem. 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. 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 $b = \min\{j : \sum_{k=1}^{j} w_k \geq \tilde{L}\}$, where $\tilde{L} = \sum_{j=1}^{N} L_j$ is the total size of all knapsacks. For the LP relaxation of (2), the Dantzig upper bound Dantzig (1957) is given by $u_{\text{MKP}} = \sum_{j=1}^{b-1} p_j + \left(\tilde{L} - \sum_{j=1}^{b-1} w_j\right) \frac{p_b}{w_b}$. The corresponding optimal solution is to accept the whole items from 1 to $b-1$ and fractional $(\tilde{L} - \sum_{j=1}^{b-1} w_j)$ item b. Suppose the item b belong to type $\tilde{i}$, then for $i < \tilde{i}$, $x^*_{ij} = 0$; for $i > \tilde{i}$, $x^*_{ij} = d_i$; for $i = \tilde{i}$, $\sum_j x^*_{ij} = (\tilde{L} - \sum_{i=\tilde{i}+1}^{M} d_i n_i)/n_{\tilde{i}}$. ■

Proof of Proposition 2

First, we construct a feasible pattern with the size of $qM + \max\{r-\delta, 0\}$, then we prove this pattern is largest. Let $L = n_M \cdot q + r$, where q represents the number of times n_M is selected (the quotient), and r represents the remainder, indicating the number of remaining seats. It holds that $0 \leq r < n_M$. The number of people accommodated in the pattern $\boldsymbol{h}_g$ is given by $|\boldsymbol{h}_g| = qM + \max\{r - \delta, 0\}$. To establish the optimality of $|\boldsymbol{h}_g|$ as the largest number of people accommodated given the constraints of L, δ, and M, we can employ a proof by contradiction.

Assuming the existence of a pattern $\boldsymbol{h}$ such that $|\boldsymbol{h}| > |\boldsymbol{h}_g|$, we can derive the following inequalities:

$$
\begin{aligned}
&\sum_i (n_i - \delta) h_i > qM + \max\{r - \delta, 0\} \\
\Rightarrow\ & L \geq \sum_i n_i h_i > \sum_i \delta h_i + qM + \max\{r - \delta, 0\} \\
\Rightarrow\ & q(M + \delta) + r > \sum_i \delta h_i + qM + \max\{r - \delta, 0\} \\
\Rightarrow\ & q\delta + r > \sum_i \delta h_i + \max\{r - \delta, 0\}
\end{aligned}
$$

(i) When $r > \delta$, the inequality becomes $q + 1 > \sum_i h_i$. It should be noted that h_i represents the number of group type i in the pattern. Since $\sum_i h_i \leq q$, the maximum number of people that can be accommodated is $qM < qM + r - \delta$.

(ii) When $r \leq \delta$, we have the inequality $q\delta + \delta \geq q\delta + r > \sum_i \delta h_i$. Similarly, we obtain $q + 1 > \sum_i h_i$. Thus, the maximum number of people that can be accommodated is qM, which is not greater than $|\boldsymbol{h}_g|$.

Therefore, $\boldsymbol{h}$ cannot exist. The maximum number of people that can be accommodated in the largest pattern is $qM + \max\{r - \delta, 0\}$. ■

Proof of Proposition 3

First of all, we demonstrate the feasibility of problem (4). Given the feasible seat plan $\boldsymbol{H}$ and $\tilde{d}_i = \sum_{j=1}^{N} H_{ji}$, let $\hat{x}_{ij} = H_{ji}, i \in \mathcal{M}, j \in \mathcal{N}$, then $\{\hat{x}_{ij}\}$ satisfies the first set of constraints. Because $\boldsymbol{H}$ is feasible, $\{\hat{x}_{ij}\}$ satisfies the second set of constraints and integer constraints. Thus, problem (4) always has a feasible solution.

Suppose there exists at least one pattern $\boldsymbol{h}$ is neither full nor largest in the optimal seat plan obtained from problem (4). Let $\beta = L - \sum_i n_i h_i$, and denote the smallest group type in pattern $\boldsymbol{h}$ by k. If $\beta \geq n_1$, we can assign at least n_1 seats to a new group to increase the objective value. Thus, we consider the situation when $\beta < n_1$. If $k = M$, then this pattern is largest. When $k < M$, let $h_k^1 = h_k - 1$ and $h_j^1 = h_j + 1$, where $j = \min\{M, \beta + k\}$. In this way, the constraints will still be satisfied but the objective value will increase when the pattern $\boldsymbol{h}$ changes. Therefore, by contradiction, problem (4) always generate a seat plan composed of full or largest patterns. ■

Proof of Proposition 4

Suppose that H is the seat plan associated with the optimal solution to SBSP, but there exists a pattern that is neither full nor the largest. The corresponding excess of supply is $\mathbf{y}^+$. According to Proposition 3, H' can be obtained from H. The seat plan, H', is composed of full or largest patterns and satisfies all constraints of SBSP. The corresponding excess of supply is $\mathbf{y}^{'+}$.

Then we will demonstrate that for each scenario ω, the objective function of SBSP, given by $\sum_{j=1}^{N} \sum_{i=1}^{M} i \cdot x_{ij} - \sum_{i=1}^{M} y_{i\omega}^+$, does not decrease when transitioning from H to H'.

Let $\Delta y_{M\omega}^+ = y_{M\omega}^{'+} - y_{M\omega}^+$, $\Delta \sum_{j=1}^{N} x_{Mj} = \sum_{j=1}^{N} x_{Mj}^{'} - \sum_{j=1}^{N} x_{Mj}$. According to (5), when i changes from M to 1, we obtain the following inequalities.

$$
\begin{aligned}
\Delta y_{M\omega}^+ \geq & \Delta \sum_{j=1}^{N} x_{Mj} \\
\Delta y_{M-1,\omega}^+ \geq & \Delta y_{M\omega}^+ + \Delta \sum_{j=1}^{N} x_{M-1,j} \geq \Delta \sum_{j=1}^{N} (x_{Mj} + x_{M-1,j}) \\
\vdots \cdots \geq & \cdots \vdots \\
\Delta y_{1,\omega}^+ \geq & \Delta \sum_{j=1}^{N} \sum_{i=1}^{M} x_{i,j}
\end{aligned}
$$

Since the objective function does not decrease, $H^{'}$ represents the optimal solution to SBSP and is composed of full or largest patterns. ■

Proof of Lemma 1

Note that $\mathbf{f}^\intercal = [-\mathbf{1},\ \mathbf{0}]$ and $\mathbf{V} = [\mathbf{W},\ \mathbf{I}]$. Based on this, we can derive the following inequalities: $\boldsymbol{\alpha}^\intercal \mathbf{W} \geq -\mathbf{1}$ and $\boldsymbol{\alpha}^\intercal \mathbf{I} \geq \mathbf{0}$. According to the expression of $\mathbf{W}$ and $\mathbf{I}$, we can deduce that $0 \leq \alpha_i \leq \alpha_{i-1} + 1$ for $i \in \mathcal{M}$ by letting $\alpha_0 = 0$. These inequalities indicate that the feasible region is nonempty and bounded. For $i \in \mathcal{M}$, α_i is only bounded by $\alpha_{i-1} + 1$ and 0, thus, all extreme points within the feasible region are integral. ■

Proof of Proposition 5

According to the complementary slackness property, we can obtain the following equations

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

When $y^-_{i\omega} > 0$, we have $\alpha_i = 0$. When $y^+_{i\omega} > 0$, we have $\alpha_i = \alpha_{i-1} + 1$. When $y^+_{i\omega} = y^-_{i\omega} = 0$, let $\Delta d = d_\omega - d_0$,

- if $i = M$, $\Delta d_M = 0$, the value of objective function associated with α_M is always 0, thus we have $0 \leq \alpha_M \leq \alpha_{M-1} + 1$;
- if $i < M$, we have $y^+_{i+1,\omega} = \Delta d_i \geq 0$.
  - If $y^+_{i+1,\omega} > 0$, the objective function associated with α_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 $0 \leq \alpha_i \leq \alpha_{i-1} + 1$.

■

Proof of Proposition 6

Suppose we have one extreme point $\boldsymbol{\alpha}^0_\omega$ for each scenario. Then we have the following problem.

$$
\begin{aligned}
\max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} p_\omega z_\omega \\
\text{s.t.} \quad & \mathbf{n}\mathbf{x} \leq \mathbf{L} \\
& (\boldsymbol{\alpha}^0_\omega)^\intercal \mathbf{d}_\omega \geq (\boldsymbol{\alpha}^0_\omega)^\intercal \mathbf{x}\mathbf{1} + z_\omega, \forall \omega \\
& \mathbf{x} \in \mathbb{N}^{M \times N}
\end{aligned} \tag{19}
$$

Problem (19) reaches its maximum when $(\boldsymbol{\alpha}^0_\omega)^\intercal \mathbf{d}_\omega = (\boldsymbol{\alpha}^0_\omega)^\intercal \mathbf{x}\mathbf{1} + z_\omega, \forall \omega$. Substitute z_ω with these equations, we have

$$
\begin{aligned}
\max \quad & \mathbf{c}^\intercal \mathbf{x} - \sum_{\omega} p_\omega (\boldsymbol{\alpha}^0_\omega)^\intercal \mathbf{x}\mathbf{1} + \sum_{\omega} p_\omega (\boldsymbol{\alpha}^0_\omega)^\intercal \mathbf{d}_\omega \\
\text{s.t.} \quad & \mathbf{n}\mathbf{x} \leq \mathbf{L} \\
& \mathbf{x} \in \mathbb{N}^{M \times N}
\end{aligned} \tag{20}
$$

Notice that $\mathbf{x}$ is bounded by $\mathbf{L}$, then the problem (19) is bounded. Adding more constraints will not make the optimal value larger. Thus, RBMP is bounded. ■

Proof of Lemma 2

According to the Proposition 1, the aggregate optimal solution to LP relaxation of problem (2) takes the form $xe_{\tilde{i}} + \sum_{i=\tilde{i}+1}^{M} d_i e_i$, then according to the complementary slackness property, we know that $z_1, \ldots, z_{\tilde{i}} = 0$. This implies that $\beta_j \geq \frac{n_i - \delta}{n_i}$ for $i = 1, \ldots, \tilde{i}$. Since $\frac{n_i - \delta}{n_i}$ increases with i, we have $\beta_j \geq \frac{n_{\tilde{i}} - \delta}{n_{\tilde{i}}}$. Consequently, we obtain $z_i \geq n_i - \delta - n_i \frac{n_{\tilde{i}} - \delta}{n_{\tilde{i}}} = \frac{\delta(n_i - n_{\tilde{i}})}{n_{\tilde{i}}}$ for $i = h + 1, \ldots, M$.

Given that $\mathbf{d}$ and $\mathbf{L}$ are both no less than zero, the minimum value will be attained when $\beta_j = \frac{n_{\tilde{i}} - \delta}{n_{\tilde{i}}}$ for all j, and $z_i = \frac{\delta(n_i - n_{\tilde{i}})}{n_{\tilde{i}}}$ for $i = \tilde{i} + 1, \ldots, M$. ■

Proof of Corollary 1

According to Proposition 2, any largest pattern $\boldsymbol{h}$ under M remains a feasible pattern under $(M+1)$, implying that $\phi(M, L; \delta) \leq \phi(M + 1, L; \delta)$. Similarly, since any largest pattern $\boldsymbol{h}$ under L is a feasible pattern under $L + 1$, thus, $\phi(M, L) \leq \phi(M, L + 1)$. ■

C Probabilities Estimation

We select Movie A (representing the suspense genre) and Movie B (representing the family fun genre) as target movies to analyze group information and their corresponding probability distributions, denoted as $D3$ and $D4$, respectively.

We make the screenshots about the ticket seat plans from a Hong Kong cinema website at different time intervals. When tickets were sold in advance of the movie screening, the seats were typically scattered. Therefore, we treated consecutive seats as belonging to the same group, while excluding cases where the number of consecutive seats exceeds four.

We counted the frequency of different group types in the seat plans to derive their probability distributions. For Movie A, the frequencies for the four group types are 112, 460, 121, and 226, with a total of 919 observations. For Movie B, the frequencies are 116, 178, 23, and 28, with a total of 345 observations. We keep two decimal places, then obtain the probability:

$p_1^A = 0.12$, $p_2^A = 0.50$, $p_3^A = 0.13$, $p_4^A = 0.25$

$p_1^B = 0.34$, $p_2^B = 0.52$, $p_3^B = 0.07$, $p_4^B = 0.08$

Using the normal distribution approximation method (with a 95% confidence interval):

$p_i \pm z_{\alpha/2}\sqrt{\frac{p_i(1-p_i)}{N}}$, $z_{\alpha/2} = 1.96$

The confidence intervals for the probabilities of each group type for Movie A as follows: $CI_1^A = 0.122 \pm 0.011$, $CI_2^A = 0.501 \pm 0.016$, $CI_3^A = 0.132 \pm 0.011$, $CI_4^A = 0.246 \pm 0.014$

Similarly, the confidence intervals for the probabilities of each group type for Movie B are:

$CI_1^B = 0.336 \pm 0.025$, $CI_2^B = 0.516 \pm 0.027$, $CI_3^B = 0.067 \pm 0.013$, $CI_4^B = 0.081 \pm 0.015$