Seating Management under Social Distancing

March 15, 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.

Terminologies to use

We use *seating management* to refer to the general problem which includes *seat planning with deterministic requests*, *seat planning with stochastic requests*, and *seat assignment*.

Each problem is defined for an *event* which has multiple *seating requests*, where each request has a *group* of people to be seated.

1 Introduction

Social distancing is a proven concept for containing the spread of an infectious disease. It has been widely adopted worldwide, for example, during the most recent Covid-19 pandemic. As a general principle, social distancing measures can be specified from different dimensions. The basic requirement of social distancing is the specification of a minimum physical distance between people in public areas. For example, the World Health Organization (WHO) suggests social distancing as to "keep physical distance of at least 1 meter from others" [35]. In the US, the Centers for Disease Control and Prevention (CDC) refers to social distancing as "keeping a safe space between yourself and other people who are not from your household" [8]. Note that under such a requirement, social distancing is actually applied with respect to groups of people. Similarly in Hong Kong, the government has adopted social distancing measures, in the recent 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 adopted an upper limit on the total number of people in a venue; for example, restaurants can operate at 50% or 75% of their normal seating capacity.

The implementation of social distancing measures has an extended impact beyond disease control. In particular, social distancing may disrupt the usual operations in certain sectors. For example, a restaurant needs to change or redesign the layout of its tables in order to fulfill the requirement of social distancing. Such change implies smaller capacity, fewer customers and less revenue. In such a context, an effected firm faces a new operational problem of optimizing its operations flow under given social distancing policies.

The impact of enforcing social distancing measures on economic activities is also an important factor for governmental decision making. Facing an outbreak of an infectious disease, a government shall declare a social distancing policy based on a holistic analysis, considering not only the severity of the outbreak, but also the potential impact on all stakeholders. What is particularly important is the level of business loss 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 to be used in an event. The venue is equipped with seats of multiple rows. In the event, requests for seats are in groups where each group contains a limited number of people. Any group can be accepted or rejected, and the people in an accepted group will sit 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, as the requirement of the social distancing measures. The objective is to accept the number of individuals as many as possible.

We will consider three problems for managing the seats, referred to as seat planning with deterministic requests (SPDR), seat planning with stochastic requests (SPSR), and seat assignment with dynamic demand (SADD), respectively. As we elaborate below, each of these models defines a standalone problem with suitable situations. Together, they are inherently connected to each other, jointly forming a suite of solution schemes for seating management under the social distancing constraints.

In the first problem, SPDR, we are given the complete information about seating requests in groups, and the problem is to find a seat plan which specifies a partition of the layout into small segments to match the seating requests. Such a problem is applicable for cases of which participants and their groups are known, such as people from the same family in a church gathering, and staff from the same office in a company meeting. We formulate the problem by Integer Programming and discuss some characteristics of the optimal plan.

In the second problem, SPSR, we need to find a seat plan facing the requests in terms of a probabilistic distribution. This problem may find its applications in situations where a new layout needs to be made for serving multiple events with different seating requests. For example, there are theaters [26] physically removing some seats during the Covid-19 outbreak, where the remaining seats essentially form a seat plan with stochastic requests. We formulate the problem by scenario-based optimization and develop solution approaches by Benders decomposition.

In the third problem, SADD, 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 groups. Seat assignment is applicable in commercial settings where requests arrive as a stochastic process, such as ticket sales in movie theaters. A DP-based heuristic approach is employed to determine whether to accept or reject each request, followed by a group-type control policy deciding whether to assign seats to the accepted groups.

The above three problems are closely related to each other with respect to problem solving methods and managerial insights. For example, in seat planning with deterministic requests, we identify some useful concepts such as the full patterns and largest patterns, which are important in the solution development for the other two problems. In addition, the duality analysis in the seat planning with deterministic requests facilitates the subproblem solving in the Benders decomposition algorithm for seat planning with stochastic requests. Also, the solution of seat planning with stochastic requests can be used as a reference seat plan in seat assignment.

Besides developing models and solution schemes for operational solutions satisfying social distancing requirements, we are also interested in understanding the impact of social distancing realized over particular events. Note that although the seating capacity is reduced by social distancing, this does not necessarily mean the same reduction of the number of people to be held for an event, especially when the event needs a small number of seats. For example, consider a seat plan with 70 seats available in a venue of 100 seats, i.e., a 30% reduction of the seating capacity. If an event held in the venue needs less than 70 seats, then it is possible that there will be a small number of people to be rejected, which implies that the loss caused by the social distancing is much less than 30%. It is important for a government to include such an effect in policy making.

We address the above issue from the following aspects.

1. We introduce the concept of the gap point to characterize situations where social distancing begins to cause loss to an event. Roughly speaking, given a distribution of group sizes for incoming requests, the gap point represents an upper bound on the number of requests an event can accommodate without being affected by social distancing. Specifically, if an event has fewer requests than the gap point, it will experience virtually no loss due to social distancing. Our computational experiments demonstrate that the gap point primarily depends on the mean of the group size and is relatively insensitive to the exact distribution. This provides a straightforward method for estimating the gap point and assessing the impact of social distancing.

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

3. The above qualitative insights are stable with respect to the tightness of the policy as well as the specific characteristics of various venues, such as the minimum physical distance, the allowable largest group size, and the layout of the venue. When the minimum physical distance increases, both the threshold occupation rate and maximum achievable occupation rate decrease accordingly. When the allowable largest group size decreases, although the number of accepted requests increases, the threshold occupation rate and maximum achievable occupation rate decrease accordingly. 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.

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 introduces the dynamic seat assignment problem. Section 6 demonstrates the dynamic seat assignment policy to assign seats for incoming groups. Section 7 gives the numerical results and insights of implementing social distancing. Conclusions are shown in Section 8.

2 Literature Review

The present study is closely connected to the following research areas: seat management with social distancing, multiple knapsack problem and network revenue management. The subsequent sections review the literature on each perspective and highlight significant differences between the present study and previous research.

2.1 Seating Management with Social Distancing

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 [9,12]. 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 [25]. In parliamentary seating assignments, members of the same party are typically grouped in clusters to facilitate intra-party communication as much as possible [34]. In e-sports gaming centers, customers arrive to play games in groups and require seating arrangements that allow them to sit together [24].

Incorporating social distancing into seating management has introduced an additional layer of complexity, sparking a new stream of research. In some studies, addressing social distancing involves optimizing the layout design itself. The focus is on determining seating positions within a given venue to maximize physical distance between individuals, such as students in classrooms [6] or customers in restaurants and beach umbrella arrangements [15]. In other cases, where the seating layout is fixed, individuals are assigned seats while adhering to social distancing guidelines. Examples include problems in air travel [17] and long-distance train travel [18]. These studies underscore the growing relevance and importance of seating management in the context of social distancing.

Our work relates to seating management with social distancing for group-based requests, which has found its applications in various areas, including airplanes [29], trains [19], and theaters [5]. Due to the diversity of applications, there are different issues to handle. For example, in [29], the distance between different groups is taken into account, leading to the development of a seating assignment strategy that outperforms the simplistic airline policy of blocking all middle seats. In [19], when designing seat allocation for groups with social distancing, not only was the transmission risk inside the train considered, but also the transmission risk between different cities where the stops were located.

Our work in this paper is most closely related to [5], in that both addressing group-based seating problem in theaters. In [5], they primarily focus on the cases with known groups, which is referred to as seat planning with deterministic requests in this paper, we have a broader scope. We also consider group-based seat planning with stochastic requests. Additionally, we incorporate dynamic seat assignment, assuming that groups arrive with a certain probability, to provide a comprehensive solution pattern.

2.2 Multiple Knapsack Problem

When all requests are known in SPDR, this problem belongs to the *multiple knapsack problem* (MKP) [14, 28], which has been studied with efficient algorithm. However, our problem focuses on the analysis of properties. The knapsacks can have different capacities and there are many identical groups of the same size, thus, the aggregation form would be useful to obtain the seat plan. Furthermore, seat planning to utilize all seats is a crucial aspect of our analysis, which shows that our proposed approach will be different from those in the literature.

To address stochastic demand, we propose a scenario-based programming approach [7, 13, 20] to determine the seat plan. Unlike existing literature, which typically considers the total supply for different demands or treats each type of supply as a protection level for corresponding demand types, we introduce a hierarchical consumption mechanism for the aggregated supply. Specifically, we adapt the constraints to reflect the hierarchical utilization of seats, where seats planned for larger groups can be repurposed for smaller groups. Additionally, the seat plan derived under demand uncertainty can serve as a reference for dynamic seat assignment, enhancing flexibility and effectiveness in real-time decision-making.

In dynamic seat assignment, the decision to either reject or accept-and-assign groups is made at each stage upon their arrival. Our problem falls under the category of the dynamic multiple knapsack problem. When there is only one row, the problem reduces to the dynamic knapsack problem [23]. However, there is limited research on the dynamic multiple knapsack problem, with only one mildly related study [27]. This study models a multiperiod, single-resource capacity reservation problem by using multiple knapsacks to represent multiple time periods, so that this paper does not apply to ours.

2.3 Network Revenue Management

Our work is also related to the group-based *network revenue management* (NRM) problem, which focuses on deciding whether to accept or reject a reques [16]. 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.

[31] was the first to propose the bid-price control policy. 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 [3], 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 [16], which studied a static model and introduced make-to-stock and make-to-order policies. However, these policies lack flexibility in handling stochastic demand and do not perform well for the group-based problem.

One of the key characteristics of our problem is that decisions must be made on an all-or-none basis for each group, which introduces additional complexity in handling group arrivals [32]. Similar group-based characteristics are also observed in multi-day stays in hotel revenue management [1, 4], though the concept of a "group" in those contexts differs from our problem.

The introduction of group-based characteristics complicates seat management when using traditional bid-price and booking-limit control policies. 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 [10, 33].

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. [36] 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.

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 distancing, 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 [26].

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.*

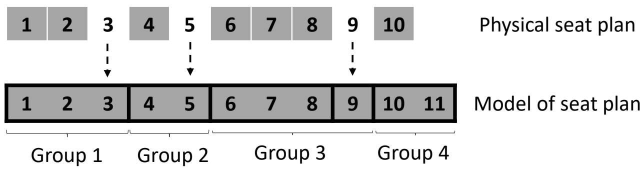


Figure 1: Illustration of Groups with Social Distancing

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 $|\boldsymbol{h}| = 7$.

The seat planning with deterministic requests problem (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.

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

The objective is to maximize the number of individuals accommodated. Constraint (1) ensures the number of accommodated groups does not exceed the number of requests. Constraint (2) 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 Section 9.

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

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

The seat plan obtained from SPDRP may not utilize all available seats, as it depends on the given requests. To improve a given seat plan and utilize all seats, we aim to generate a new seat plan with full or largest patterns while ensuring that the original group type requirements are met.

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

In other words, a full pattern ensures that the available row seats are fully occupied. A largest pattern ensures that it can accommodate the maximum number of individuals within the given row size.

Proposition 2. *If the size of a feasible pattern $\boldsymbol{h}$ is $|\boldsymbol{h}| = qM + \max\{r - \delta, 0\}$, where $q = \lfloor \frac{L}{M+\delta} \rfloor$, and $r = L - q(M + \delta)$, then this pattern is a largest pattern.*

The size, $qM + \max\{r - \delta, 0\}$, corresponds directly to a largest pattern that includes q group type M and r seats for one group type $(r - \delta)$ when $r > \delta$. However, the form of the largest pattern is not unique; there are other largest patterns that share the same size. Specifically, when $r = 0$, the largest pattern $\boldsymbol{h}$ is unique and full, indicating that only one pattern can accommodate the maximum number of individuals; when $r > \delta$, the largest pattern $\boldsymbol{h}$ is full, as it utilizes the available space up to the social distancing requirement.

A concept closely related to the largest pattern is the *maximum achievable occupancy rate*, which will be discussed in Section 7.2 regarding the impact of social distancing. When all rows of a given layout consist of the largest pattern, the layout achieves its maximum achievable occupancy rate. This rate is defined as:

$$\frac{\sum_j \phi(M, L_j; \delta)}{\sum_j L_j - N\delta},$$

where $\phi(M, L; \delta)$ represents the size of the largest pattern under M and L. According to Proposition 2, $\phi(M, L; \delta)$ does not decrease in M. This is because any largest pattern $\boldsymbol{h}$ under M remains a feasible pattern under $(M + 1)$, implying that $\phi(M, L; \delta) \leq \phi(M + 1, L; \delta)$. Consequently, when M increases while L remains constant, the maximum achievable occupancy rate also increases.

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.

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.

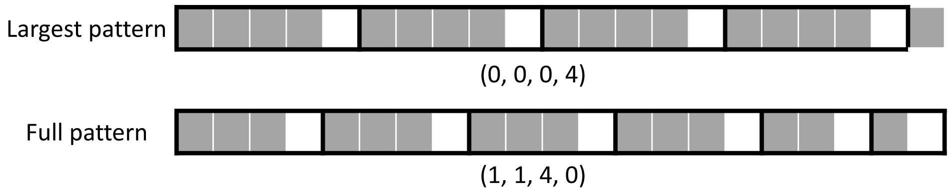


Figure 2: Largest and Full Patterns

To obtain a seat plan with the full or largest patterns, we make the following statements. Let the original seat plan be $\boldsymbol{H}$ and the desired seat plan be $\boldsymbol{H'}$. To satisfy requirements for the original group types, 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:

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

Proposition 3. *Given a feasible seat plan $\boldsymbol{H}$, the solution to problem (3) corresponds to a seat plan $\boldsymbol{H'}$ such that $\boldsymbol{H} \subseteq \boldsymbol{H'}$ and $\boldsymbol{H'}$ is composed of 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:

$$
\begin{aligned}
y^+_{i\omega} &= (\sum_{j=1}^{N} x_{ij} - d_{i\omega} + y^+_{i+1,\omega})^+, i = 1, \ldots, M-1 \\
y^-_{i\omega} &= (d_{i\omega} - \sum_{j=1}^{N} x_{ij} - y^+_{i+1,\omega})^+, i = 1, \ldots, M-1 \\
y^+_{M\omega} &= (\sum_{j=1}^{N} x_{Mj} - d_{M\omega})^+ \\
y^-_{M\omega} &= (d_{M\omega} - \sum_{j=1}^{N} x_{Mj})^+,
\end{aligned}
\tag{4}
$$

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:

$$\max \quad E_\omega \left[(n_M - \delta)(\sum_{j=1}^{N} x_{Mj} - y^+_{M\omega}) + \sum_{i=1}^{M-1} (n_i - \delta)(\sum_{j=1}^{N} x_{ij} + y^+_{i+1,\omega} - y^+_{i\omega}) \right] \tag{5}$$

$$\text{s.t.} \quad \sum_{j=1}^{N} x_{ij} - y^+_{i\omega} + y^+_{i+1,\omega} + y^-_{i\omega} = d_{i\omega}, \quad i = 1, \ldots, M-1, \omega \in \Omega \tag{6}$$

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

$$\sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} \tag{8}$$

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

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

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 (6) and (7) 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 [2]. 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 $\mathbf{n} = (n_1, \ldots, n_M)^\intercal$ represent the vector of seat sizes for each group type, where n_i denotes the size of seats taken by group type i. Let $\mathbf{L} = (L_1, \ldots, L_N)^\intercal$ represent the vector of row sizes, where L_j denotes the size of row j as defined previously. The constraint (8) can be expressed as $\mathbf{x}^\intercal\mathbf{n} \leq \mathbf{L}$. 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 (6) and (7), can be expressed in matrix form as:

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

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

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

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,

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal\mathbf{x} + z(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x}^\intercal\mathbf{n} \le \mathbf{L} \\ & \mathbf{x} \in \mathbb{N}^{M\times N}, \end{aligned} \tag{9}$$

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 $\omega \in \Omega$,

$$\begin{aligned} z_\omega(\mathbf{x}) := \max \quad & \mathbf{f}^\intercal\mathbf{y}_\omega \\ \text{s.t.} \quad & \mathbf{V}\mathbf{y}_\omega = \mathbf{d}_\omega - \mathbf{x}\mathbf{1} \\ & \mathbf{y}_\omega \ge 0. \end{aligned} \tag{10}$$

The efficiency of solving problem (9) hinges on the method used to solve problem (10). Next, we will demonstrate how to solve problem (10) efficiently.

4.2.2 Solve The Subproblem

Notice that the feasible region of the dual of problem (10) 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

$$\begin{aligned} \min \quad & \boldsymbol{\alpha}_\omega^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \\ \text{s.t.} \quad & \boldsymbol{\alpha}_\omega^\intercal\mathbf{V} \ge \mathbf{f}^\intercal \end{aligned} \tag{11}$$

Lemma 1. *The feasible region of problem* (11) *is nonempty and bounded. Furthermore, all the extreme points of the feasible region are integral.*

Let $\mathbb{P}$ indicate the feasible region of problem (11). According to Lemma 1, the optimal value of the problem (10), $z_\omega(\mathbf{x})$, 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 $\boldsymbol{\alpha}_\omega^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \ge z_w, \forall \boldsymbol{\alpha}_\omega \in \mathcal{O}$. We use this characterization of $z_w(\mathbf{x})$ in problem (9) and conclude that problem (9) can thus be put in the form by setting z_w as the variable:

$$
\begin{aligned}
\max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} p_\omega z_\omega \\
\text{s.t.} \quad & \mathbf{x}^\intercal \mathbf{n} \leq \mathbf{L} \\
& \boldsymbol{\alpha}_\omega^\intercal (\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega, \forall \boldsymbol{\alpha}_\omega \in \mathcal{O}, \forall \omega \\
& \mathbf{x} \in \mathbb{N}^{M \times N}
\end{aligned} \tag{12}
$$

Before applying Benders decomposition to solve problem (12), it is important to address the efficient computation of the optimal solution to problem (11). When $\mathbf{x}^*$ is given, $\mathbf{y}_\omega$ can be obtained from equation (4). Let $\alpha_{0,\omega} = 0$ for each ω, then we have Proposition 5.

Proposition 5. *The optimal solutions to problem* (11) *are given by*

$$
\begin{aligned}
\alpha_{i\omega} &= 0 && \text{if } y_{i\omega}^- > 0, i = 1, \dots, M \text{ or } y_{i\omega}^- = y_{i\omega}^+ = 0, y_{i+1,\omega}^+ > 0, i = 1, \dots, M-1 \\
\alpha_{i\omega} &= \alpha_{i-1,\omega} + 1 && \text{if } y_{i\omega}^+ > 0, i = 1, \dots, M \\
0 \leq \alpha_{i\omega} &\leq \alpha_{i-1,\omega} + 1 && \text{if } y_{i\omega}^- = y_{i\omega}^+ = 0, i = M \text{ or } y_{i\omega}^- = y_{i\omega}^+ = 0, y_{i+1,\omega}^+ = 0, i = 1, \dots, M-1
\end{aligned} \tag{13}
$$

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 (11) 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 (12) 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 (12). This results in a modified problem known as the Restricted Benders Master Problem (RBMP). To find the optimal solution to problem (12), we employ the technique of constraint generation. It involves iteratively solving the RBMP and incrementally adding more constraints until the optimal solution to problem (12) is obtained.

We can conclude that the RBMP will have the form:

$$
\begin{aligned}
\max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} p_\omega z_\omega \\
\text{s.t.} \quad & \mathbf{x}^\intercal \mathbf{n} \leq \mathbf{L} \\
& \boldsymbol{\alpha}_\omega^\intercal (\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega, \boldsymbol{\alpha}_\omega \in \mathcal{O}^t, \forall \omega \\
& \mathbf{x} \in \mathbb{N}^{M \times N}
\end{aligned} \tag{14}
$$

Given the initial $\mathcal{O}^t$, we can have the solution $\mathbf{x}^*$ and $\mathbf{z}^* = (z_1^*, \dots, z_{|\Omega|}^*)$. Then $c^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega z_\omega^*$ is an upper bound of problem (14). When $\mathbf{x}^*$ is given, the optimal solution, $\tilde{\boldsymbol{\alpha}}_\omega$, to problem (11) 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 (14) 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 (12).

If for every scenario ω, the optimal value of the corresponding problem (11) 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 (12). However, if there exists at least one scenario ω for which the optimal value of problem (11) 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 $\boldsymbol{\alpha}_\omega = \mathbf{0}, \forall\omega$, and let $LB \leftarrow 0$, $UB \leftarrow \infty$;
while $UB - LB > \epsilon$ **do**
    Solve problem (14) and obtain an optimal solution $(\mathbf{x}^*, \mathbf{z}^*)$;
    $UB \leftarrow c^\intercal \mathbf{x}^* + \sum_{\omega\in\Omega} p_\omega z_\omega^*$;
    for $\omega = 1, \dots, |\Omega|$ **do**
        Obtain $\tilde{\boldsymbol{\alpha}}_\omega$ according to Proposition 5;
        $\tilde{z}_\omega = (\tilde{\boldsymbol{\alpha}}_\omega)^\intercal(\mathbf{d}_\omega - \mathbf{x}^*\mathbf{1})$;
        if $\tilde{z}_\omega < z_\omega^*$ **then**
            Add one new constraint, $(\tilde{\boldsymbol{\alpha}}_\omega)^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega$, to problem (14);
        end
    end
    $LB \leftarrow c^\intercal \mathbf{x}^* + \sum_{\omega\in\Omega} p_\omega \tilde{z}_\omega$;
end

However, solving problem (14) directly can be computationally challenging in some cases, so we practically first obtain the optimal solution to the LP relaxation of problem (9). Then, we generate a seat plan from this solution.

4.3 Obtain The Feasible Seat Plan

We may obtain a fractional optimal solution when we solve the LP relaxation of problem (9). 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 (14) 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 (3) 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 (3) with $\boldsymbol{H} = \tilde{\boldsymbol{H}}$, and obtain the seat plan $\boldsymbol{H}'$;

5 Dynamic Seat Assignment with Social Distancing

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 $u_{i,j}^t$, where $u_{i,j}^t = 1$ indicates acceptance of group type i in row j during period t, while $u_{i,j}^t = 0$ signifies rejection of that group type in row j. The feasible decision set is defined as

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

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

Let $V^t(\mathbf{L})$ denote the 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:

$$V^t(\mathbf{L}) = \max_{u_{i,j}^t \in U^t(\mathbf{L})} \left\{ \sum_{i=1}^{M} p_i \big(\sum_{j=1}^{N} i u_{i,j}^t + V^{t+1}(\mathbf{L} - \sum_{j=1}^{N} n_i u_{i,j}^t \mathbf{e}_j) \big) + p_0 V^{t+1}(\mathbf{L}) \right\} \tag{15}$$

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 (15) 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. In Section 6, building upon the generated seat plan, we further develop a dynamic seat assignment policy which guides the allocation of seats to the incoming requests sequentially.

6 Seat Assignment with Dynamic Demand

We propose our policy for assigning arriving requests in a dynamic context. First, we employ relaxed dynamic programming to determine whether to prepare a request for assignment or to reject it. Then, we develop the seat assignment approach based on the seat plan genrated from Section 4.

6.1 DP-Based Heuristic

To simplify the complexity of DP (15), 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:

$$V^t(l) = \max_{u_i^t \in \{0,1\}} \left\{ \sum_i p_i [V^{t+1}(l - n_i u_i^t) + i u_i^t] + p_0 V^{t+1}(l) \right\} \tag{16}$$

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 [22]. 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 (16), $\forall t = 2, \dots, T; \forall l = 1, 2, \dots, l^1 = \tilde{L}$;
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.

6.2 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 6.2.2 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).

6.2.1 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 $\hat{\imath}$ to accept a group type i for any $\hat{\imath} = i + 1, \dots, M$. In this case, when $\hat{\imath} = i + 1, \dots, i + \delta$, the expected number of accepted individuals is i and the remaining seats beyond the accepted group, which is $\hat{\imath} - i$, will be wasted. When $\hat{\imath} = i + \delta + 1, \dots, M$, the rest $(\hat{\imath} - i - \delta)$ seats can be provided for one group type $(\hat{\imath} - i - \delta)$ with δ seats of social distancing. Let $D_{\hat{\imath}}^t$ be the random variable that indicates the number of group type $\hat{\imath}$ in the future t periods. The expected number of accepted people is $i + (\hat{\imath} - i - \delta) P(D_{\hat{\imath}-i-\delta}^{T-t} \geq X_{\hat{\imath}-i-\delta} + 1)$, where $P(D_i^{T-t} \geq X_i)$ 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, $P(D_{\hat{i}-i-\delta}^{T-t} \geq X_{\hat{i}-i-\delta}+1)$, 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 $\hat{i}P(D_{\hat{i}}^{T-t} \geq X_{\hat{i}})$. The term, $P(D_{\hat{i}}^{T-t} \geq X_{\hat{i}})$, 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

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

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.

6.2.2 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.

6.2.3 Regenerate the Seat Plan

A useful technique often applied in network revenue management to enhance performance is resolving [21, 30], 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 (16), $\forall t = 2, \ldots, T; \forall l = 1, 2, \ldots, l^1 = \tilde{L}$;
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

7 Computational Experiments

We carried out several experiments, including analyzing the performances of different policies, evaluating the impact of implementing social distancing, comparing different layouts, Ms and social distancings. In the experiments, we set the following parameters.

The default setting in the experiments is as follows, $\delta = 1$ and $M = 4$. The number of scenarios in SBSP is $|\Omega| = 1000$. 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 $\gamma = \sum_{i=1}^{M} ip_i$. Each experiment result is the average of 100 instances.

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 with in [5]. 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. Specifically, 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. The seat plans for the tickets were obtained from a Hong Kong cinema website. We focused on scattered seat plans and excluded cases where the number of consecutive seats exceeded four. By counting the occurrences of different group types, we obtained the distributions.

We use $D4$ as the default probability distribution in other experiments.

7.1 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), 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 9. 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.

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

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

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.

7.2 Impact of Social Distancing

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

The *gap point* $\tilde{T}$ is defined as the largest value of T for which the inequality $E(T; \delta = 1) + 1 \geq E(T; \delta = 0)$ holds, where $E(T; \delta = 1)$ denotes the average number of accepted individuals by DSA with one seat as social distancing, $E(T; \delta = 0)$ denotes the average number of accepted individuals by DSA when there is no social distancing.

The occupancy rate corresponding to the gap point is referred to as the *threshold occupancy rate.* This rate represents the maximum demand that can be satisfied when the difference in the number of accepted individuals remains unaffected by social distancing constraints.

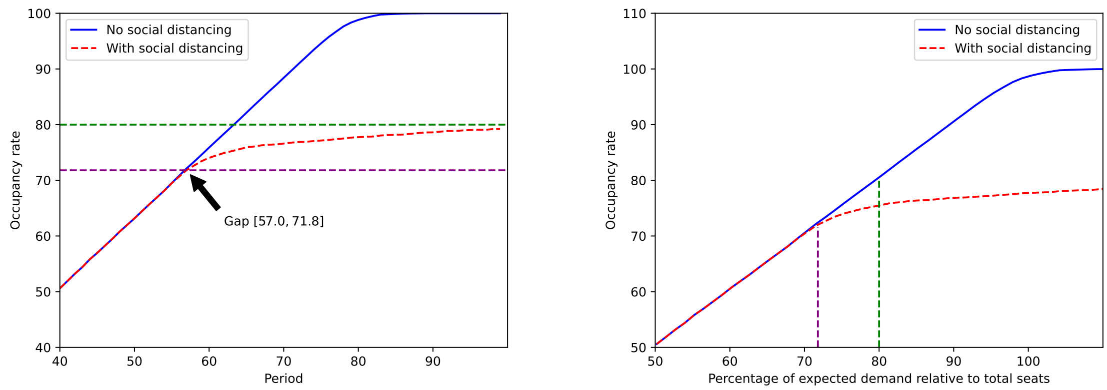
The *maximum achievable occupancy rate* is attained when all rows of a given layout consist of the largest pattern. This rate is defined as:

$$\frac{\sum_j \phi(M, L_j; \delta)}{\sum_j L_j - N\delta},$$

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

Then, we examine the impact of social distancing when implementing DSA under varying levels of demand. Two situations: $\delta = 0$ (no social distancing) and $\delta = 1$ (with social distancing) are tested. 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. Figure 3(b) displays occupancy rate over expected demand.



(a) (b)

Figure 3: The occupancy rate over demand

In figure 3(a), the gap point is 57, the threshold occupancy rate is 71.8%. As the expected demand continues to increase, both situations reach their maximum capacity acceptance. For the social distancing situation, when the largest pattern is realized in each row, the maximum achievable occupancy rate is given by $\frac{16}{20} = 80\%$. The figure 3(b) is plotted to show that when the expected demand is less than 71.8%, the social distancing measures will not have an impact; when the expected demand is larger than 71.8%, the difference between the number of accepted individuals with and without social distancing measures becomes more pronounced.

Impact of Maximum Allowable Occupancy Rate

Sometimes, policies impose a maximum allowable occupancy rate to enforce stricter social distancing. This maximum allowable rate becomes redundant if it exceeds the maximum achievable rate for all events. As shown by the green line in figure 3(a), when the maximum allowable rate is above 80%, it has no effect. Only the occupancy rate requirement is effective, while the social distancing requirement becomes irrelevant for events with an occupancy rate below the threshold. This is illustrated by the purple line in figure 3(a), where a maximum allowable rate below 71.8% renders only the occupancy rate requirement effective. Additionally, 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.

7.3 Estimation of Gap Points

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 γ and the gap point, we conducted an analysis using a sample of 200 probability distributions. The figure below illustrates the gap point as a function of γ, along with the corresponding estimations. Additionally, the threshold occupancy rate is represented by red points.

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 blue line in the figure) and $y_2 = c_2\frac{\gamma}{\gamma+\delta}\frac{\tilde{L}}{\tilde{L}-N\delta}$ (represented by the orange 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 $c_2 = 0.9576$.

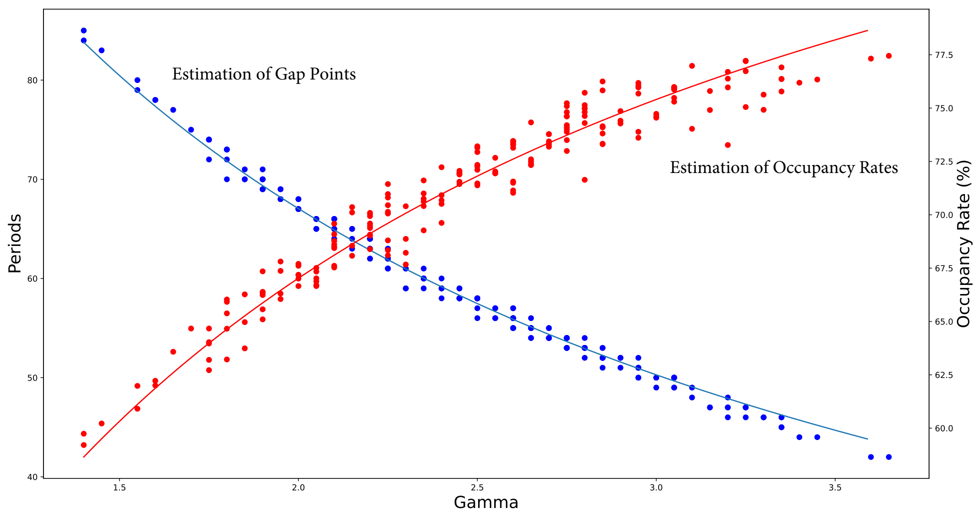


Figure 4: Gap points and their estimation under 200 probabilities

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, maximal 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 threshold 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.

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.

Corollary 1. *The maximum achievable occupancy rate does not decrease in M for a fixed layout.*

Different Social Distances

The following figure illustrates the occupancy rate over periods with social distancing set at 0, 1, and 2 seats, respectively.

Table 3: Gap points and threshold 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 %

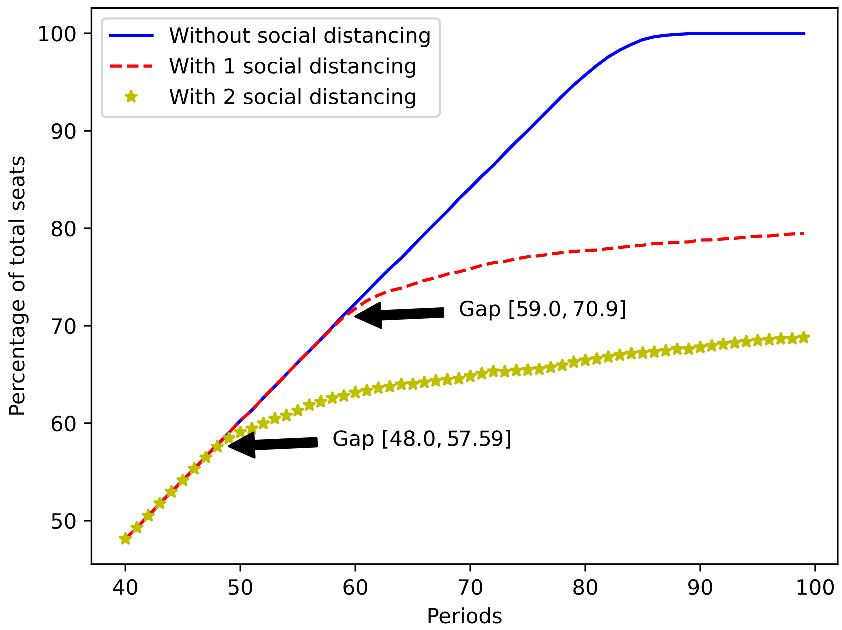


Figure 5: The occupancy rate over demand for different social distancings

8 Conclusion

We study the seat planning and seat assignment problem under social distancing requirement. Specifically, we first consider seat planning with deterministic requests. To utilize all seats, we introduce the full and largest patterns. Subsequently, we investigate seat planning with stochastic requests. To tackle this problem, we propose a scenario-based stochastic programming model. Then, we utilize the Benders decomposition method to efficiently obtain 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 classical bid-price control, booking limit control in revenue management, dynamic programming-based heuristics, and the first-come-first-served policy. Comparatively, our proposed policy exhibited superior and consistent performance.

Building upon our policies, we further evaluate the impact of implementing social distancing. By defining the gap point to characterize the situations under which social distancing begins to cause loss to an event, the 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 expected 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 allowed occupancy rate to tighten the social distancing requirement. This maximum allowed rate is effective for an event if it is lower than the expected occupancy rate of the event. Furthermore, the maximum allowed rate will be redundant if it is higher than the maximum achievable rate for all events. The above qualitative insights are stable with respect to different parameters in the model, such as the minimum physical distances, the maximum group sizes, and the layout of the venue.

Future research can be pursued in several ways. First, when the seating requests are established, a scattered seat assignment can be examined to maximize the distance between adjacent groups when sufficient seating is available. Second, more flexible scenarios where individuals can select the seats by their preference may be considered. Third, research could also examine the problem where individuals can arrive and leave at different times in the shared areas.

References

[1] Nursen Aydin and S Ilker Birbil. Decomposition methods for dynamic room allocation in hotel revenue management. *European Journal of Operational Research*, 271(1):179–192, 2018.

[2] J Benders. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math*, 4(1):238–252, 1962.

[3] Dimitris Bertsimas and Ioana Popescu. Revenue management in a dynamic network environment. *Transportation Science*, 37(3):257–277, 2003.

[4] Gabriel R Bitran and Susana V Mondschein. An application of yield management to the hotel industry considering multiple day stays. *Operations Research*, 43(3):427–443, 1995.

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

[6] Juliano Cavalcante Bortolete, Luis Felipe Bueno, Renan Butkeraites, et al. A support tool for planning classrooms considering social distancing between students. *Computational and Applied Mathematics*, 41:1–23, 2022.

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

[8] CDC. Keep a safe space. https://stacks.cdc.gov/view/cdc/90522, 2020.

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

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

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

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

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

[14] Carlos E Ferreira, Alexander Martin, and Robert Weismantel. Solving multiple knapsack problems by cutting planes. *SIAM Journal on Optimization*, 6(3):858–877, 1996.

[15] Martina Fischetti, Matteo Fischetti, and Jakob Stoustrup. Safe distancing in the time of COVID-19. *European Journal of Operational Research*, 304(1):139–149, 2023.

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

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

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

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

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

[21] Stefanus Jasin and Sunil Kumar. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research*, 37(2):313–345, 2012.

[22] David S Johnson. Fast algorithms for bin packing. *Journal of Computer and System Sciences*, 8(3):272–314, 1974.

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

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

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

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

[27] Thomas C Perry and Joseph C Hartman. An approximate dynamic programming approach to solving a dynamic, stochastic multiple knapsack problem. *International Transactions in Operational Research*, 16(3):347–359, 2009.

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

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

[30] Nicola Secomandi. An analysis of the control-algorithm re-solving issue in inventory and revenue management. *Manufacturing & Service Operations Management*, 10(3):468–483, 2008.

[31] Kalyan Talluri and Garrett Van Ryzin. An analysis of bid-price controls for network revenue management. *Management Science*, 44(11-part-1):1577–1593, 1998.

[32] Kalyan T Talluri and Garrett J Van Ryzin. *The Theory and Practice of Revenue Management*. Springer Science & Business Media, 2006.

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

[34] Bart Vangerven, Dirk Briskorn, Dries R Goossens, and Frits CR Spieksma. Parliament seating assignment problems. *European Journal of Operational Research*, 296(3):914–926, 2022.

[35] WHO. Advice for the public. https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public, 2020.

[36] Feng Zhu, Shaoxuan Liu, Rowan Wang, and Zizhuo Wang. Assign-to-seat: Dynamic capacity control for selling high-speed train tickets. *Manufacturing & Service Operations Management*, 25(3):921–938, 2023.

9 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:

$$\begin{aligned} \min \quad & \sum_{i=1}^{M} d_i z_i + \sum_{j=1}^{N} L_j \beta_j \\ \text{s.t.} \quad & z_i + \beta_j n_i \geq (n_i - \delta), \quad i \in \mathcal{M}, j \in \mathcal{N} \\ & z_i \geq 0, i \in \mathcal{M}, \beta_j \geq 0, j \in \mathcal{N}. \end{aligned} \tag{17}$$

In (17), 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* (17) *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

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 (1), the Dantzig upper bound [11] 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 (3). 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 (3) 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 (3). 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 (3) 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 (4), 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

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

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

$$
\begin{aligned}
\max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega\in\Omega} p_\omega z_\omega\\
\text{s.t.} \quad & \mathbf{n}\mathbf{x} \leq \mathbf{L}\\
& (\boldsymbol{\alpha}^0_\omega)^\intercal \mathbf{d}_\omega \geq (\boldsymbol{\alpha}^0_\omega)^\intercal \mathbf{x}\mathbf{1} + z_\omega, \forall\omega\\
& \mathbf{x} \in \mathbb{N}^{M\times N}
\end{aligned}
\tag{18}
$$

Problem (18) 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

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

Notice that $\mathbf{x}$ is bounded by $\mathbf{L}$, then the problem (18) 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 (1) 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 **d** and **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, $\phi(M, L; \delta)$ does not decrease in M. This is because any largest pattern $\boldsymbol{h}$ under M remains a feasible pattern under $(M + 1)$, implying that $\phi(M, L; \delta) \leq \phi(M + 1, L; \delta)$. Consequently, when M increases while $\boldsymbol{L}$ remains unchanged, the maximum achievable occupancy rate does not decrease. ■