Seat Planning and Seat Assignment with Social Distancing

August 25, 2024

Abstract

This study addresses the seat planning and seat assignment problem with social distancing. We focus on the dynamic situation which arises when groups arrive at a venue and need to be seated together while respecting minimum physical distance requirements. To tackle this challenge, we develop a scenario-based stochastic programming for generating seat planning and propose the dynamic seat assignment policy for accepting or denying arriving groups. Our approach performs better than the traditional bid-price and booking-limit policies. The results provide insights for policymakers and venue managers on seat utilization rates and offer a practical tool for implementing social distancing measures while optimizing seat assignments.

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

1 Introduction

After the pandemic, it is crucial to sum up the lessons learned and evaluate the policies implemented during this time, including social distancing, nucleic acid testing, and quarantine measures. Our focus will be on analyzing the requirements for social distancing. By examining relevant policies and their specific implementations, we aim to gain valuable insights that can assist the government in providing tools and making informed decisions to mitigate the negative impacts of social distancing policies. Our work aims to provide the government with targeted decision-making support so that they can adopt the optimal social distancing measures based on actual circumstances. This approach will help protect public health while minimizing damage to the socioeconomic system. Furthermore, it will enhance the overall effectiveness of pandemic control and better prepare us for future public health crises.

Social distancing is a proven concept to contain the spread of an infectious disease. As a general principle, social distancing measures can be implemented in various forms. 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” [22]. In the US, the Center for Disease and Control (CDC) refers to social distancing as “keeping a safe space between yourself and other people who are not from your household" [6]. Note that under such a requirement, social distancing is actually applied with respect to groups of people. In Hong Kong, the government has adopted social distancing measures, in the recent Covid 19 pandemic, by limiting the size of groups in public gathering to two, four, and six people per group over time. Moreover, the Hong Kong government has also adopted an upper limit of the total number of people in a venue; for example, restaurants can operate at 50% or 75% of their normal seating capacity.

While the above practice of social distancing measures has been recognized for containing the spread of an infectious disease, it is not clear how the entire economy will be affected in terms of the change of operations. This is an important issue in the service sector where social distancing implies fewer clients and lower revenue. The situation is especially complicated under multiple social distancing measures, such as physical distance between groups, limit on the size of groups, and the occupancy rate of the venue. This naturally raises questions regarding the relationship of these measures, e.g., which is relatively more effective under different conditions, whether they complement or contradict each other, and more importantly, is it possible to align these measures so that they can be implemented coherently?

The distancing measures discussed can also be applied to other domains, such as the arrangement of animal cages and the placement of GPUs. During customs quarantine, different types of animals from various regions need to maintain a certain distance to meet epidemic prevention requirements. Implementing a reasonable distance-based placement can ensure safety while not occupying an excessive amount of space. Similarly, with the rapid development of big data and artificial intelligence, a large number of GPUs are required as computational tools. GPU clustering can lead to overheating, which is detrimental to heat dissipation and affects operational efficiency. Arranging the placement of GPUs with appropriate distances can help address this issue and improve the overall system performance.

We will address the above issues of social distancing in the context of seating arrangement in a venue, such as a cinema or a conference hall. The venue is equipped with seats of multiple rows. People come in groups where each group of people will sit consecutively in one row. To avoid confusion, we discuss two related terms 'seat planning' and 'seat assignment' which will be used in the following parts. In our context, the seat planning is to determine the seat partition with the known customer distribution. The planning can be altered later when the planned seats don't match the size of a coming group or when the seat planning is disrupted after assigning a coming group. In the seat assignment, for the coming group, when accepting it, we assign the seats to the group, and the seats will not be used by others in the future.

In order to adhere to social distancing guidelines, it is important to understand the process of generating seat planning based on known groups and how to assign seats to incoming groups. Additionally, it is of interest to explore how the social distancing constraints impact the sellers and the specific policies formulated by the government to address social distancing concerns.

We intend to shed light on the problem described and propose a practical dynamic seat assignment policy. In particular, we investigate the following questions.

1. How can we model the seat planning problem given the social distancing restrictions? What kind of property does this problem have? How can we give a seat planning to accommodate the maximum

people with stochastic demand?

2. How to use the property of seat planning problem to design the dynamic seat assignment policy? How good is the performance of this policy compared with other policies?

3. What kind of insights regarding the social distancing and occupancy rates can we obtain when implementing the dynamic seat assignment policy?

To answer these questions, we construct the seat planning problem with deterministic and stochastic demand under social distancing requirement. For the deterministic situation, we have complete and accurate information about the demand for seating. We aim to provide a seat planning that maximizes the number of people accommodated. This situation is applicable in venues like churches or company meetings, where fixed seat layouts are available, and the goal is to assign seats to accommodate as many people as possible within the given layout. The seat planning obtained shows the utilization of all seats as many as possible. Thus, we introduce the concept of full or largest pattern to indicate the seat partition of each row. For the seat planning that does not utilize all available seats, we propose to improve the seat planning by incorporating full or largest patterns.

For the stochastic situation, we know that the demand distribution before the actual demand is realized. We aim to generate a seat planning that maximizes the expected number of people accommodated. This approach is suitable for venues where seats have been pre-allocated to ensure compliance with social distancing rules. With the given demand scenarios, we develop the scenario-based stochastic programming to obtain the seat planning. To solve this problem efficiently, we apply the Benders decomposition technique. However, in some cases, solving the integer programming with Benders decomposition remains still computationally prohibitive. Thus, we can consider the LP relaxation then obtain a feasible seat planning by deterministic model. Based on that, we construct a seat planning composed of full or largest patterns to utilize all seats fully.

We mainly focus on addressing the dynamic seat assignment problem with a given set of seats in the context of social distancing. Solving the problem by dynamic programming can be prohibitive due to the curse of dimensionality, which arises when the problem involves a large number of variables or states. To mitigate this complexity, we begin by generating a seat planning in the stochastic situation. This seat planning acts as a foundation for the seat assignment. Then, we develop the dynamic seat assignment policy which guides the allocation of seats to the incoming groups sequentially. In the numerical result, our policy performs well compared with other policies. We use $\tilde{T}$ to denote the gap point, which refers to the first period at which, on average, the number of people accepted without social distancing is not less than that accepted with social distancing plus one. By sampling many probability combinations, the results show that $\tilde{T}$ and the corresponding occupancy rate, $\beta(\tilde{T})$, can be estimated with γ, the expected number of people per period. Different γ corresponds to different $\tilde{T}$. When the total number of periods, T, is less than $\tilde{T}$, we tend to accept all incoming groups. In this case, whether to implement the social distancing restriction is no difference. When T is larger, there will be more groups rejected when implementing the social distancing requirement. The government can consider the potential losses when making policies regarding group size and occupancy rate. Similarly, the seller can implement corresponding measures to adhere to these requirements.

Our main contributions in this paper are summarized as follows. First, this study presents the first attempt to consider the arrangement of seat assignments with social distancing under dynamic arrivals. Our study provides a new perspective to help the government adopt a mechanism for setting seat assignments to implement social distancing during pandemic.

Second, we establish a deterministic model to analyze the effects of social distancing when the demand is known. Due to the medium size of the problem, we can solve the IP model directly. We then develop the scenario-based stochastic programming by considering the stochastic demands of different group types. By using Benders decomposition methods, we can obtain the seat planning quickly.

Third, to address the problem in the dynamic situation, we first obtain a feasible seat planning from scenario-based stochastic programming. We then decide for each incoming group based on our dynamic seat assignment policy, either accepting or rejecting the group. Our results demonstrate a significant improvement over the traditional control policies and provide insights on the implementation of social distancing.

The rest of this paper is structured as follows. The following section reviews relevant literature. We describe the motivating problem in Section 3. In Section 4, we establish the stochastic model,analyze its properties and obtain the seat planning. Section 5 demonstrates the dynamic seat assignment policy to assign the seats for incoming groups. Section 6 gives the numerical results and the insights of implementing social distancing. The conclusions are shown in Section 7.

2 Literature Review

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

2.1 Seat Planning with Social Distancing

Social distancing in seat planning has attracted significant attention from the research community. This topic involves various applications and optimization techniques to address the challenges posed by the need for physical distancing in different settings.

In the context of layout design, i.e., determine the seats location in the given venue. For example, maximizing the social distancing between students in the classroom [5]. Fischetti et al. [10] consider how to plant positions with social distancing and apply it in restaurants and beach umbrellas. In some other applications, where the layout is predetermined, individuals are assigned seats while adhering to social distancing guidelines. For instance, in the field of air travel, seat planning with a given layout is addressed in the literature [12, 20]. Additionally, the optimization of seating arrangements has also been investigated in the context of long-distance train travel [14].

Group seat reservation had two main applications in terms of revenue optimization and customer preference satisfaction. In the transportation domain, such as for passenger rail services, the primary focus of these studies was on maximizing capacity utilization or reducing the total capacity required [7,9]. Similar optimization techniques have also been applied in the context of seat planning for large events, such as weddings or dinner events [18]. In these cases, the focus was on satisfying customer preferences and enhancing the overall experience.

The pandemic has amplified the focus on and importance of group-based seat planning, especially within the framework of social distancing mandates. It has underscored the potential advantages of group reservations in curbing the spread of infections, as they can be structured to boost revenue without heightening transmission risks. The issue of group seat reservations has evolved into a more intricate problem with the constraint of social distancing. This complexity presents diverse applications across various industries, including: airplanes [20], trains [15], sports arenas [17], theaters [4].

The major difference in our study is that we consider group-based seat planning with stochastic demand not only the known specific demand. Meanwhile, we consider the dynamic seat assignment assuming the groups arrive in a certain probability.

2.2 Dynamic Seat Assignment

In dynamic seat assignment, the decision to either reject or accept-and-assign groups is made at each stage upon their arrival. This problem can be regarded as a special case of the dynamic multiple knapsack problem. When there is one row, the related problem is dynamic knapsack problem [16]. Our model in its static form can be viewed as a specific instance of the multiple knapsack problem [19]. To the best of our knowledge, there are no existing studies that have specifically addressed the dynamic multiple knapsack problem.

Dynamic seat assignment has applications in the transportation industry, such as for airplanes, trains, and buses [2, 13, 24]. It is a process of assigning seats to passengers in a way that maximizes the efficiency and convenience of the seating arrangements.

Our problem is closely related to the group-based network revenue management (RM) problem [23], which is typically formulated as a dynamic programming (DP) problem. However, for large-scale problems, the exponential growth of the state space and decision set makes the DP approach computationally intractable. One of the characteristics we are studying is that the decision should be made on an all-or-none basis for each group, which is the real complication in group arrivals [21]. The group trait reflected in the RM is the hotel revenue management considering multiple day stays [1, 3].

Another characteristic in our study is the significance of seat assignment. This sets it apart from traditional revenue management focusing on accepting or rejecting a request [11]. The assign-to-seat feature introduced by Zhu et al. [24] also highlights the importance of seat assignment in revenue management. This trait addresses the challenge of selling high-speed train tickets in China, where each request must be assigned to a single seat for the entire journey and takes into account seat reuse.

Our work not only focuses on how to make the group-based seat assignment with non-reusable seats, also contribute the revenue insights with social distancing.

3 Seat Planning Problem with Social Distancing

This section integrates social distancing measures into the seat planning process. By introducing some concepts, we present the deterministic seat planning problem. For the seat planning that does not utilize all available seats, we propose to improve the seat planning.

3.1 Concepts

Consider a seat layout comprising N rows, with each row containing L_j^0 seats, where $j \in \mathcal{N} := \{1, 2, \ldots, N\}$. There are M distinct group types, denoted by group type i, where each group type consists of i individuals. The set of all group types is denoted by $\mathcal{M} := \{1, 2, \ldots, M\}$. The demand for each group type is represented by a demand vector $\mathbf{d} = (d_1, d_2, \ldots, d_M)^\intercal$, where d_i is the demand of group type i.

We incorporate social distancing in the seat planning problem. To comply with the social distancing requirements, individuals from the same group must sit together in one specific row, while maintaining a distance from other groups. 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 add the parameter, δ, to the original group types, resulting in the size of group type i being denoted as $n_i = i + \delta$, where $i \in \mathcal{M}$. Accordingly, the size of each row is also adjusted to accommodate the group sizes. Consequently, $L_j = L_j^0 + \delta$ represents the size of row j, where L_j^0 indicates the number of seats in row j. By incorporating the additional seat(s) and designating certain seat(s) for social distancing, we can integrate social distancing measures into the seat planning problem.

We introduce the term pattern to represent 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 is the number of group 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 $\boldsymbol{h} \in \mathbb{N}^M$. Then a seat planning with N rows can be expressed by $\boldsymbol{H} = \{\boldsymbol{h}_1; \ldots; \boldsymbol{h}_N\}$.

Let $|\boldsymbol{h}|$ indicate the maximum number of people that can be assigned according to pattern $\boldsymbol{h}$, i.e., $|\boldsymbol{h}| = \sum_{i=1}^{M} i h_i$. The size of $\boldsymbol{h}$, $|\boldsymbol{h}|$, provides a measure of the maximum number of seats that can be taken due to the implementation of social distancing constraints. By examining $|\boldsymbol{h}|$ associated with different patterns, we can assess the effectiveness of various seat planning configurations for accommodating the desired number of individuals while adhering to social distancing requirements.

Example 1. *Consider the given values: $\delta = 1$, $L^0 = 10$, and $M = 4$. By adding one seat to each group and the original row, we can realize the conversion, as shown in the following figure.*

After the conversion, $L = L^0 + 1 = 11$, $n_i = i + 1$ for $i = 1, 2, 3, 4$. For the first group, we use a group of three seats to indicate the original group of two and one seat used as the social distancing. We do the same procedure for the other groups. Then, the row can be represented by $\boldsymbol{h} = (2, 1, 1, 0)$. The maximum number of people that can be accommodated is $|\boldsymbol{h}| = 7$.

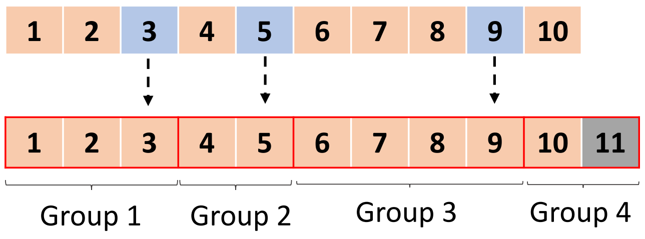


Figure 1: Illustration of Groups with Social Distancing

Let x_{ij} represent the number of group type i planned in row j. The deterministic seat planning problem (SPP($\{d_i\}$)) is formulated below, with the objective of maximizing the number of people accommodated.

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

In this context, we define $\boldsymbol{X}_i = \sum_{j=1}^{N} x_{ij}$ as the supply to group type i for $i \in \mathcal{M}$. Then $\boldsymbol{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_M)^T$ is the aggregate solution. In other words, $\boldsymbol{X}$ captures the number of each group type that can be allocated to the seat layout by summing up the supplies across all rows. By considering the monotone ratio between the original group sizes and the adjusted group sizes, we can determine the upper bound of supply corresponding to the optimal solution of the LP relaxation of Problem (1), as demonstrated in Proposition 1.

Proposition 1. *For the LP relaxation of problem* (1)*, 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* $i = \tilde{i}$*.*

For $i = 1, \ldots, \tilde{i}-1$, the optimal solutions have $x_{ij}^* = 0$ for all rows, indicating that no group type i lower than index $\tilde{i}$ are assigned to any rows. For $i = \tilde{i}+1, \ldots, M$, the optimal solution assigns $\sum_j x_{ij}^* = d_i$ group type i to meet the demand for group type i. For $i = \tilde{i}$, the optimal solution assigns $\sum_j x_{ij}^* = \frac{\sum_{j=1}^{N} L_j - \sum_{i=\tilde{i}+1}^{M} d_i n_i}{n_{\tilde{i}}}$ group type $\tilde{i}$ to the rows. This quantity is determined by the available supply, which is calculated as the remaining seats after accommodating the demands for group types $\tilde{i}+1$ to M, divided by the size of group type $\tilde{i}$, denoted as $n_{\tilde{i}}$.

Hence, the corresponding supply associated with the optimal solutions can be summarized as follows: $X_{\tilde{i}} = \frac{\sum_{j=1}^{N} L_j - \sum_{i=\tilde{i}+1}^{M} d_i n_i}{n_{\tilde{i}}}$, $X_i = d_i$ for $i = \tilde{i}+1, \ldots, M$, and $X_i = 0$ for $i = 1, \ldots, \tilde{i}-1$.

3.2 Seat Planning Composed of Full or Largest Patterns

The seat planning obtained from problem (1) may not utilize all available seats, as it depends on the given demand. To improve a given seat planning and utilize all seats, we aim to generate a new seat planning composed of 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 with size L. We refer to $\boldsymbol{h}$ as a full pattern if $\sum_{i=1}^{M} n_i h_i = L$, $\boldsymbol{h}$ 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 is one in which the sum of the product of the number of occurrences h_i and the size n_i of each group in the pattern is equal to the size of the row L. This ensures that the pattern fully occupies the available row seats. A largest pattern is one that either has the maximum size or is equal in size to other patterns, ensuring that it can accommodate the maximum number of people 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 the size of the row is L, the social distancing is δ, the number of distinct group types is M, $q = \lfloor \frac{L}{M+\delta} \rfloor$, and $r \equiv L \bmod (M + \delta)$, then this pattern is a largest pattern.*

Proposition 2 states that if the size of a pattern is $|\boldsymbol{h}| = qM + \max\{r - \delta, 0\}$, then this pattern is the largest. Furthermore, according to the definition of the largest pattern, the size of one possible largest pattern is given by $qM + \max\{r - \delta, 0\}$.

When $r = 0$, the largest pattern $\boldsymbol{h}$ is unique and full, indicating that only one pattern can accommodate the maximum number of people. On the other hand, if $r > \delta$, the largest pattern $\boldsymbol{h}$ is full, as it utilizes the available space up to the social distancing requirement.

Example 2. *Consider the given values: $\delta = 1$, $L = 21$, and $M = 4$. In this case, we have $n_i = i+1$ for $i = 1, 2, 3, 4$. The size of the largest pattern can be calculated as $qM + \max\{r - \delta, 0\} = 4 \times 4 + 0 = 16$. The largest patterns are the following: $(1, 0, 1, 3)$, $(0, 1, 2, 2)$, $(0, 0, 0, 4)$, $(0, 0, 4, 1)$, and $(0, 2, 0, 3)$.*

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

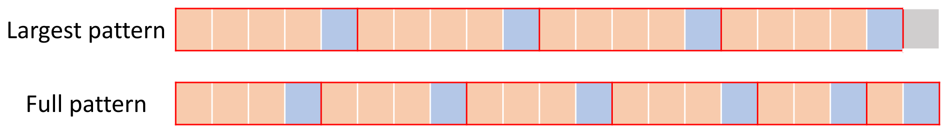


Figure 2: Largest and Full Patterns

The first row can be represented by $(0, 0, 0, 4)$. It is a largest pattern as its size is 16. However, it does not satisfy the requirement of fully utilizing all available seats since $4 \times 5 \neq 21$. The second row can be represented by $(1, 1, 4, 0)$, which is a full pattern as it utilizes all available seats. However, its size is 15, indicating that it is not a largest pattern.

Suppose the original seat planning is $\boldsymbol{H}$, the desired seat planning is $\boldsymbol{H'}$. To meet the original group type requirements, for each group type i, the total quantity from group i to group M in $\boldsymbol{H'}$ should be no less than the total quantity from group type i to group type M in $\boldsymbol{H}$. Mathematically, we aim to find a feasible seat planning $\boldsymbol{H'}$ such that $\sum_{k=i}^{M}\sum_{j=1}^{N} H_{ji} \leq \sum_{k=i}^{M}\sum_{j=1}^{N} H^{'}_{ji}, \forall i \in \mathcal{M}$. We say $\boldsymbol{H} \subseteq \boldsymbol{H}^{'}$ if this condition holds.

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

$$\begin{aligned}
\max \quad & \sum_{i=1}^{M}\sum_{j=1}^{N}(n_i - \delta)x_{ij} \\
s.t. \quad & \sum_{j=1}^{N}\sum_{k=i}^{M} x_{kj} \geq \sum_{k=i}^{M}\sum_{j=1}^{N} H_{ji}, i \in \mathcal{M} \\
& \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} \\
& x_{ij} \in \mathbb{N}, i \in \mathcal{M}, j \in \mathcal{N}
\end{aligned} \tag{2}$$

Proposition 3. *Given a feasible seat planning $\boldsymbol{H}$, the solution to problem (2) corresponds to a seat planning $\boldsymbol{H'}$ such that $\boldsymbol{H} \subseteq \boldsymbol{H'}$ and $\boldsymbol{H}^{'}$ is composed of full or largest patterns.*

This approach guarantees efficient seat allocation by constructing full or largest patterns while still accommodating the original groups' requirements. Furthermore, the improved seat planning can be used for the seat assignment when the demand arrives sequentially.

4 Dynamic Seat Assignment with Social Distancing

In many commercial situations, groups arrive sequentially over time, and the seller must promptly make group assignments upon each arrival while maintaining the required spacing between groups. When a group is accepted, the seller must also determine which seats should be assigned to that group. It is essential to note that each group must be either accepted in its entirety or rejected entirely. Once the seats are confirmed and assigned to a group, they cannot be changed or reassigned to other groups.

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 group arrives and the probability of an arrival for a group of size i is denoted as p_i, where i belongs to the set $\mathcal{M}$. The probabilities satisfy the constraint $\sum_{i=1}^{M} p_i \leq 1$, indicating that the total probability of any group arriving in a single period does not exceed one. We introduce the probability $p_0 = 1 - \sum_{i=1}^{M} p_i$ to represent the probability of no arrival in a given period t. 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.

At time t, the state of remaining capacity in each row is represented by a vector $\mathbf{L}^t = (l_1^t, l_2^t, \ldots, l_N^t)$, where l_j^t denotes the number of remaining seats in row j at time t. 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

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

Here, $\mathbf{e}_j$ represents an N-dimensional unit column vector with the j-th element being 1, i.e., $\mathbf{e}_j = (\underbrace{0, \cdots, 0}_{j-1}, 1, \underbrace{0, \cdots, 0}_{n-j})$. In other words, the decision set $U(\mathbf{L}^t)$ 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 $V^t(\mathbf{L}^t)$ denote the maximum expected revenue earned by the best decisions regarding group seat assignments in period t, given remaining capacity $\mathbf{L}^t$. Then, the dynamic programming formula for this problem can be expressed as:

$$V^t(\mathbf{L}^t) = \max_{u_{i,j}^t \in U^t(\mathbf{L})} \left\{ \sum_{i=1}^{M} p_i \Big(\sum_{j=1}^{N} i u_{i,j}^t + V^{t+1}(\mathbf{L}^t - \sum_{j=1}^{N} n_i u_{i,j}^t \mathbf{e}_j) \Big) + p_0 V^{t+1}(\mathbf{L}^t) \right\} \tag{3}$$

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.

At the beginning of period t, we have the current remaining capacity vector denoted as $\mathbf{L} = (L_1, L_2, \ldots, L_N)$. 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 $V^1(\mathbf{L})$.

Solving the dynamic programming problem described in equation (3) 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 planning, as outlined in section 5. This seat planning acts as a foundation for the seat assignment. In section 6, building upon the generated seat planning, we further develop a dynamic seat assignment policy which guides the allocation of seats to the incoming groups sequentially.

5 Seat Planning with Stochastic Demand

In this section, we aim to obtain a seat planning which is suitable to the dynamic seat assignment.Specially, we develop the Scenario-based Stochastic Programming (SSP) to obtain the seat planning with available capacity. Due to the well-structured nature of SSP, we implement Benders decomposition to solve it efficiently. However, in some cases, solving the integer programming with Benders decomposition remains still computationally prohibitive. Thus, we can consider the LP relaxation first, then obtain a feasible seat planning by deterministic model. Based on that, we construct a seat planning composed of full or largest patterns to fully utilize all seats.

5.1 Scenario-based Stochastic Programming Formulation

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

The seat planning can be represented by decision variables $\mathbf{x} \in \mathbb{N}^{M \times N}$. Here, x_{ij} represents the number of group type i assigned to row j in the seat planning. As mentioned earlier, we calculate the supply for group type i as the sum of x_{ij} over all rows j, denoted as $\sum_{j=1}^{N} x_{ij}$. However, considering the variability across different scenarios, it is necessary to model the potential excess or shortage of supply. To capture this characteristic, we introduce a scenario-dependent decision variable, denoted as $\mathbf{y}$. It includes two vectors of decisions, $\mathbf{y}^+ \in \mathbb{N}^{M \times |\Omega|}$ and $\mathbf{y}^- \in \mathbb{N}^{M \times |\Omega|}$. Each component of $\mathbf{y}^+$, denoted as $y^+_{i\omega}$, represents the excess supply for group type i for each scenario ω. On the other hand, $y^-_{i\omega}$ represents the shortage of supply for group type i for each scenario ω.

Taking into account the possibility of groups occupying seats planned for larger group types when the corresponding supply is insufficient, we make the assumption that surplus seats for group type i can be occupied by smaller group types $j < i$ in descending order of group size. This means that if there are excess supply available after assigning groups of type i to rows, we can provide the supply to groups of type $j < i$ in a hierarchical manner based on their sizes. That is, for any ω, $i \leq M - 1$,

$$y^+_{i\omega} = \left(\sum_{j=1}^{N} x_{ij} - d_{i\omega} + y^+_{i+1,\omega} \right)^+, \; y^-_{i\omega} = \left(d_{i\omega} - \sum_{j=1}^{N} x_{ij} - y^+_{i+1,\omega} \right)^+,$$

where $(x)^+$ equals x if $x > 0$, 0 otherwise. Specially, for the largest group type M, we have $y^+_{M\omega} = (\sum_{j=1}^{N} x_{Mj} - d_{M\omega})^+$, $y^-_{M\omega} = (d_{M\omega} - \sum_{j=1}^{N} x_{Mj})^+$. Based on the above mentioned considerations, 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:

$$\max \quad E_\omega\left[(n_M-\delta)(\sum_{j=1}^{N}x_{Mj}-y^+_{M\omega})+\sum_{i=1}^{M-1}(n_i-\delta)(\sum_{j=1}^{N}x_{ij}+y^+_{i+1,\omega}-y^+_{i\omega})\right] \tag{4}$$

$$\text{s.t.} \quad \sum_{j=1}^{N}x_{ij}-y^+_{i\omega}+y^+_{i+1,\omega}+y^-_{i\omega}=d_{i\omega}, \quad i=1,\dots,M-1,\omega\in\Omega \tag{5}$$

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

$$\sum_{i=1}^{M}n_ix_{ij}\le L_j, j\in\mathcal{N} \tag{7}$$

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

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

We use SSP$(\mathbf{L},\Omega)$ to represent the above stochastic programming, and similarly, we use RSSP$(\mathbf{L},\Omega)$ to represent the LP relaxation of SSP.

The objective function consists of two parts. The first part represents the number of people in the 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 people in group type i, excluding M, that can be accommodated, given by $(n_i-\delta)(\sum_{j=1}^{N}x_{ij}+y^+_{i+1,\omega}-y^+_{i\omega}), i=1,\dots,M-1$. 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

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

Here, $\sum_{j=1}^{N}\sum_{i=1}^{M}i\cdot x_{ij}$ indicates the maximum number of people that can be accommodated in the seat planning $\{x_{ij}\}$. The second part, $\sum_{\omega=1}^{|\Omega|}p_\omega\sum_{i=1}^{M}y^+_{i\omega}$ indicates the expected excess 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 (5) and (6) 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.

Considering the analysis provided earlier, we find it advantageous to obtain a seat planning that only consists of full or largest patterns. However, the seat planning associated with the optimal solution obtained by solver to SSP may not consist of the largest or full patterns. We can convert the optimal solution to another optimal solution which is composed of the largest or full patterns.

Proposition 4. *There exists an optimal solution to the stochastic programming problem such that the patterns associated with this optimal solution are composed of the full or largest patterns under any given scenarios.*

Proposition 4 is different from proposition 3. Proposition 3 can make sure that we can obtain a seat planning composed of full or largest patterns. Here, proposition 4 states that there always exists an optimal solution where the corresponding patterns are either full or largest. However, we may not be able to directly find such an optimal solution as described in Proposition 4. Instead, we try to obtain the seat planning composed of full or largest patterns, as stated in Section 5.3.

Then, we reformulate SSP($\mathbf{L}, \Omega$) 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 (7) 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 (5) and (6), 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 $\omega \in \Omega$,

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

As we can find, this deterministic equivalent form is a large-scale problem even if the number of possible scenarios Ω is moderate. However, the structured constraints allow us to simplify the problem by applying Benders decomposition approach. Before using this approach, we could reformulate this problem as the following form. 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 SSP formulation can be expressed as below,

$$\begin{aligned} \max \quad & \mathbf{c}^\mathsf{T}\mathbf{x} + z(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x}^\mathsf{T}\mathbf{n} \le \mathbf{L} \\ & \mathbf{x} \in \mathbb{N}^{M\times N}, \end{aligned} \tag{8}$$

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 $\omega \in \Omega$,

$$\begin{aligned} z_\omega(\mathbf{x}) := \max \quad & \mathbf{f}^\mathsf{T}\mathbf{y}_\omega \\ \text{s.t.} \quad & \mathbf{V}\mathbf{y}_\omega = \mathbf{d}_\omega - \mathbf{x}\mathbf{1} \\ & \mathbf{y}_\omega \ge 0. \end{aligned} \tag{9}$$

We can solve problem (8) quickly if we can efficiently solve problem (9). Next, we will mention how to solve problem (9).

5.2 Solve SSP by Benders Decomposition

We reformulate problem (8) into a master problem and a subproblem (9). The iterative process of solving the master problem and subproblem is known as Benders decomposition. The solution obtained from the master problem provides inputs for the subproblem, and the subproblem solutions help update the master problem by adding constraints, iteratively improving the overall solution until convergence is achieved. Firstly, we generate a closed-form solution to problem (9), then we obtain the solution to the LP relaxation of problem (8) by the constraint generation.

5.2.1 Solve The Subproblem

Notice that the feasible region of the dual of problem (9) remains unaffected by $\mathbf{x}$. This observation provides insight into the properties of this problem. Let $\boldsymbol{\alpha}$ denote the vector of dual variables. For each ω, we can form its dual problem, which is

$$\begin{aligned} \min \quad & \boldsymbol{\alpha}_\omega^\mathsf{T}(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \\ \text{s.t.} \quad & \boldsymbol{\alpha}_\omega^\mathsf{T}\mathbf{V} \ge \mathbf{f}^\mathsf{T} \end{aligned} \tag{10}$$

Lemma 1. *The feasible region of problem (10) is nonempty and bounded. Furthermore, all the extreme points of the feasible region are integral.*

Let $\mathbb{P}$ indicate the feasible region of problem (10). According to Lemma 1, the optimal value of the problem (9), $z_\omega(\mathbf{x})$, 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 $z_\omega(\mathbf{x}) = \min_{\boldsymbol{\alpha}_\omega\in\mathcal{O}} \boldsymbol{\alpha}_\omega^\mathsf{T}(\mathbf{d}_\omega - \mathbf{x}\mathbf{1})$.

Alternatively, $z_\omega(\mathbf{x})$ is the largest number z_ω such that $\boldsymbol{\alpha}_\omega^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_w, \forall \boldsymbol{\alpha}_\omega \in \mathcal{O}$. We use this characterization of $z_w(\mathbf{x})$ in problem (8) and conclude that problem (8) can thus be put in the form by setting z_w as the variable:

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} p_\omega z_\omega \\ \text{s.t.} \quad & \mathbf{x}^\intercal \mathbf{n} \leq \mathbf{L} \\ & \boldsymbol{\alpha}_\omega^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega, \forall \boldsymbol{\alpha}_\omega \in \mathcal{O}, \forall \omega \\ & \mathbf{x} \in \mathbb{N}^{M \times N} \end{aligned} \tag{11}$$

Before applying Benders decomposition to solve problem (11), it is important to address the efficient computation of the optimal solution to problem (10). When we are given $\mathbf{x}^*$, the demand that can be satisfied by the seat planning is $\mathbf{x}^*\mathbf{1} = \mathbf{d}_0 = (d_{1,0}, \ldots, d_{M,0})^\intercal$. By plugging them in the subproblem (9), we can obtain the value of $y_{i,\omega}$ recursively:

$$\begin{aligned} & y_{M\omega}^- = (d_{M\omega} - d_{M0})^+ \\ & y_{M\omega}^+ = (d_{M0} - d_{M\omega})^+ \\ & y_{i\omega}^- = \left(d_{i\omega} - d_{i0} - y_{i+1,\omega}^+\right)^+, i = 1, \ldots, M-1 \\ & y_{i\omega}^+ = \left(d_{i0} - d_{i\omega} + y_{i+1,\omega}^+\right)^+, i = 1, \ldots, M-1 \end{aligned} \tag{12}$$

The optimal solutions to problem (10) can be obtained according to the value of $\mathbf{y}_\omega$.

Proposition 5. *The optimal solutions to problem* (10) *are given by*

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

Instead of solving this linear programming directly, we can compute the values of α_ω by performing a forward calculation from $\alpha_{1\omega}$ to $\alpha_{M\omega}$.

5.2.2 Constraint Generation

Due to the computational infeasibility of solving problem (11) 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 (11). This results in a modified problem known as the Restricted Benders Master Problem (RBMP). To find the optimal solution to problem (11), we employ the technique of constraint generation. It involves iteratively solving the RBMP and incrementally adding more constraints until the optimal solution to problem (11) 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 $\mathbf{z}^* = (z_1^*, \ldots, z_{|\Omega|}^*)$. 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 (10) 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 (11).

If for every scenario ω, the optimal value of the corresponding problem (10) 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 (11). However, if there exists at least one scenario ω for which the optimal value of problem (10) 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 always bounded with 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

```
   Input: Initial problem (14) with $\alpha_\omega = 0, \forall\omega$, $LB = 0$, $UB = \infty$, $\epsilon$.
   Output: $\mathbf{x}^*$
1  while $UB - LB > \epsilon$ do
2      Obtain $(\mathbf{x}^*, \mathbf{z}^*)$ from problem (14);
3      $UB \leftarrow c^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega z_\omega^*$;
4      for $\omega = 1, \ldots, |\Omega|$ do
5          Obtain $\tilde{\boldsymbol{\alpha}}_\omega$ from Proposition 5;
6          $\tilde{z}_\omega = (\tilde{\boldsymbol{\alpha}}_\omega)^\intercal (\mathbf{d}_\omega - \mathbf{x}^*\mathbf{1})$;
7          if $\tilde{z}_\omega < z_\omega^*$ then
8              Add one new constraint, $(\tilde{\boldsymbol{\alpha}}_\omega)^\intercal (\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega$, to problem (14);
9          end
10     end
11     $LB \leftarrow c^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega \tilde{z}_\omega$;
12 end
```

However, solving problem (14) even with the simplified constraints directly can be computationally challenging in some cases, so practically we first obtain the optimal solution to the LP relaxation of problem (8). Then, we generate an integral seat planning from this solution.

5.3 Obtain The Feasible Seat Planning

We may obtain a fractional optimal solution when we solve the LP relaxation of problem (8). 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 planning. 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 planning composed of full or largest patterns based on the feasible seat planning 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 SPP($\{\tilde{X}_i\}$) to obtain the optimal solution, $\tilde{\mathbf{x}}$, and the corresponding pattern, $\boldsymbol{H}$, then generate the seat planning by problem (2) with $\boldsymbol{H}$.

Algorithm 2: Seat Planning Construction

```
1 Obtain the optimal solution, $\mathbf{x}^*$, from the LP relaxation of problem (14);
2 Aggregate $\mathbf{x}^*$ to the number of each group type, $\tilde{X}_i = \sum_j x^*_{ij}, i \in \mathbf{M}$;
3 Obtain the optimal solution, $\tilde{\mathbf{x}}$, and the corresponding pattern, $\boldsymbol{H}$, from SPP($\{\tilde{X}_i\}$);
4 Construct the seat planning by problem (2) with $\boldsymbol{H}$;
```

6 Seat Assignment with Dynamic Demand

In this section, we address the assignment of arriving groups in the dynamic situation. Our policy involves making seat allocation decisions for each arriving group based on a seat planning strategy outlined in Section 5. We propose two approaches to make the decision. The first method is based on an modified SSP, while the second method is based on the LP relaxation of SSP. Typically, we prefer the former method if time permits. However, to ensure consistent computation time, we opt for the latter method.

6.1 Assignment Based on The Modified SSP

Suppose the supply is $[X_1, \ldots, X_M]$. When a group type i' arrives, if $X_{i'} > 0$, we accept the group directly and assign it the seats originally planned for group type i, adhering to the break-tie rule mentioned in 6.2.2. If $X_{i'} = 0$, we make the decision based on the modified SSP.

We introduce the decision variables $I_j, j \in \mathcal{N}$ to determine the appropriate row assignment. If we accept this group and assign it to row j, then I_j is equal to 1, and 0 otherwise. We need to add the constraint $\sum_{j=1}^{N} I_j \leq 1$ to the original SSP to ensure that only one row can be assigned to the group. The capacity constraint and objective function will be changed correspondingly to accommodate this new decision variable and constraint. The modified SSP can be expressed as:

$$
\begin{aligned}
\max \quad & \sum_{j} i' I_j + E_\omega \left[\sum_{i=1}^{M-1} (n_i - \delta)(\sum_{j=1}^{N} x_{ij} + y^+_{i+1,\omega} - y^+_{i\omega}) + (n_M - \delta)(\sum_{j=1}^{N} x_{Mj} - y^+_{M\omega}) \right] \\
\text{s.t.} \quad & \sum_{j=1}^{N} x_{ij} - y^+_{i\omega} + y^+_{i+1,\omega} + y^-_{i\omega} = d_{i\omega}, \quad i = 1, \ldots, M-1, \omega \in \Omega \\
& \sum_{j=1}^{N} x_{ij} - y^+_{i\omega} + y^-_{i\omega} = d_{i\omega}, \quad i = M, \omega \in \Omega \\
& \sum_{i=1}^{M} n_i x_{ij} \leq L_j - n_{i'} I_j, j \in \mathcal{N} \\
& \sum_{j=1}^{N} I_j \leq 1 \\
& x_{ij} \in \mathbb{N}, \quad i \in \mathcal{M}, j \in \mathcal{N}, y^+_{i\omega}, y^-_{i\omega} \in \mathbb{N}, \quad i \in \mathcal{M}, \omega \in \Omega, I_j \in \{0,1\}, j \in \mathcal{N}.
\end{aligned} \tag{15}
$$

After making the decision for group type i', we proceed to assign seats to the next incoming group based on the updated seat planning. The algorithm is shown as follow:

Algorithm 3: Seat Assignment with Modified SSP

```
1  for $t = 1, \ldots, T$ do
2      Observe group type $i'$;
3      if $X_{i'} > 0$ then
4          Find row $k$ such that $H_{ki'} > 0$ according to tie-breaking rule;
5          Accept group type $i$ in row $k$, update $L_k$, $H_{ki'}$, $X_{i'}$;
6      else
7          Obtain $I_j, j \in \mathcal{N}$ from problem (15);
8          if $I_j > 0$ then
9              Accept group type $i'$ in row $j$;
10             Update $L_j$, $\boldsymbol{H}$, $\boldsymbol{X}$;
11         else
12             Reject group type $i'$;
13             Update $\boldsymbol{H}$, $\boldsymbol{X}$;
14         end
15     end
16 end
```

6.2 Assignment Based on The LP Relaxation of SSP

Since solving the modified SSP may be slow, we assign the groups based on the LP relaxation of SSP in this section. Similarly, suppose the supply is $[X_1, \ldots, X_M]$. When a group type i' arrives, if $X_{i'} > 0$, we assign it the seats according to the break-tie rule that we will mention later. When $X_{i'} = 0$, we have three steps to assign the group arrival. First, we develop the group-type control policy to determine the group type that can accommodate the arriving group. Second, we choose a specific row based on the group type and the break-tie rule. In the third step, we make the decision by comparing the values of the relaxed SSPs, considering both accepting and rejecting the arriving group. In the following part, we will refer to this policy as Dynamic Seat Assignment (DSA).

In order to mitigate the computational challenges, we utilize the LP relaxation of SSP as an approximation to compare the values when deciding whether to accept or reject a group. However, one challenge arises from the fact that the LP relaxation results in the same objective values for the acceptance group in any possible row. This poses the question of determining which row to place the group in when we accept it. To address this challenge, we developed the group-type control policy.

6.2.1 Group-type Control

The group-type control aims to find the group type to assign the arriving group, that helps us narrow down the option of rows for seat assignment. The policy considers whether to use a larger group's supply to meet the need of the arriving group when given a seat planning. The group type is selected based on the tradeoff between the social distancing and the future demand. When a group is accepted and assigned to larger-size seats, the remaining empty seat(s) can be reserved for future demand without affecting the rest of the seat planning. To determine whether to use larger seats to accommodate the incoming group, we compare the expected values of accepting the group in the larger seats and rejecting the group based on the current seat planning. Then we identify the possible rows where the incoming group can be assigned based on the group types and seat availability.

Specifically, suppose the supply is $(X_1, \dots, X_M)$ at period t, the number of remaining periods is $(T-t)$. For the arriving group type i' when $X_{i'} = 0$, we demonstrate how to decide whether to accept the group to occupy larger-size seats. For any $\hat{\imath} = i'+1, \dots, M$, we can use one supply of group type $\hat{\imath}$ to accept a group type i'. In that case, when $\hat{\imath} = i'+1, \dots, i+\delta$, the expected number of accepted people 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 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, $P(D_{\hat{\imath}-i'-\delta}^{T-t} \geq X_{\hat{\imath}-i'-\delta} + 1)$, indicates the probability that the demand of group type $(\hat{\imath} - i' - \delta)$ in $(T-t)$ periods is no less than its current remaining supply plus 1.

Similarly, when we retain the supply of group type $\hat{\imath}$ by rejecting the group type i', the expected number of accepted people is $\hat{\imath} P(D_{\hat{\imath}}^{T-t} \geq X_{\hat{\imath}})$. The term, $P(D_{\hat{\imath}}^{T-t} \geq X_{\hat{\imath}})$, indicates the probability that the demand of group type $\hat{\imath}$ in $(T-t)$ periods is no less than its current remaining supply.

Let $d^t(i', \hat{\imath})$ be the difference of expected number of accepted people between acceptance and rejection on group type i' occupying $(\hat{\imath}+\delta)$-size seats at period t. Then we have

$$d^t(i',\hat{\imath}) = \begin{cases} i' + (\hat{\imath} - i' - \delta)P(D^{T-t}_{\hat{\imath}-i'-\delta} \geq X_{\hat{\imath}-i'-\delta}+1) - \hat{\imath}P(D^{T-t}_{\hat{\imath}} \geq X_{\hat{\imath}}), & \text{if } \hat{\imath} = i'+\delta+1,\ldots,M \\ i' - \hat{\imath}P(D^{T-t}_{\hat{\imath}} \geq X_{\hat{\imath}}), & \text{if } \hat{\imath} = i'+1,\ldots,i'+\delta. \end{cases}$$

One intuitive decision is to choose $\hat{\imath}$ with the largest difference. For all $\hat{\imath} = i'+1,\ldots,M$, find the largest $d^t(i',\hat{\imath})$, denoted as $d^t(i',\hat{\imath}^*)$. If $d^t(i',\hat{\imath}^*) > 0$, we will plan to assign the group type i' in $(\hat{\imath}^*+\delta)$-size seats. Otherwise, reject the group.

Group-type control policy can only tell us which group type's seats are planned to provide for the smaller group based on the current planning, we still need to further compare the values of the stochastic programming problem when accepting or rejecting a group on the specific row.

6.2.2 Break Tie for Determining A Specific Row

To determine the appropriate row for seat assignment, we can apply a tie-breaking rule among the possible options obtained by the group-type control. This rule helps us decide on a particular row when there are multiple choices available.

A tie occurs when there are serveral rows to accommodate the group. Suppose one group type i' arrives, the current seat planning is $\boldsymbol{H} = \{\boldsymbol{h}_1;\ldots;\boldsymbol{h}_N\}$, the corresponding supply is $\boldsymbol{X}$. Let $\beta_j = L_j - \sum_i (i+\delta)H_{ji}$ represent the remaining number of seats in row j after considering the seat allocation for other groups. When $X_{i'} > 0$, we assign the group to row $k \in \arg\min_j\{\beta_j\}$. That allows us to fill in the row as much as possible. When $X_{i'} = 0$ and we plan to assign the group to seats designated for group type $\hat{\imath}, \hat{\imath} > i$, we assign the group to a row $k \in \arg\max_j\{\beta_j | H_{j\hat{\imath}} > 0\}$. That helps to reconstruct the pattern with less unused seats. When there are multiple rows available, we can choose randomly. This rule in both scenarios prioritizes filling rows and leads to better capacity management.

As an example to illustrate group-type control and the tie-breaking rule, consider a situation where $L_1 = 3, L_2 = 4, L_3 = 5, L_4 = 6$, $M = 4$, $\delta = 1$. The corresponding patterns for each row are $(0,1,0,0)$, $(0,0,1,0)$, $(0,0,0,1)$ and $(0,0,0,1)$, respectively. Thus, $\beta_1 = \beta_2 = \beta_3 = 0$, $\beta_4 = 1$. Now, a group of one arrives, and the group-type control indicates the possible rows where the group can be assigned. We assume this group can be assigned to the seats of the largest group according to the group-type control, then we have two choices: row 3 or row 4. To determine which row to select, we can apply the breaking tie rule. The β value of the rows will be used as the criterion, we would choose row 4 because β_4 is larger. Because when we assign it in row 4, there will be two seats reserved for future group of one, but when we assign it in row 3, there will be one seat remaining unused.

In the above example, the group of one can be assigned to any row with the available seats. The group-type control can help us find the larger group type that can be used to place the arriving group while maximizing the expected values. Maybe there are multiple rows containing the larger group type. Then we can choose the row containing the larger group type according to the breaking tie rule. Finally, we compare the values of stochastic programming when accepting or rejecting the group, then make the corresponding decision.

6.2.3 Compare The Values of Relaxed SSP

After determining a specific row, we compare the values of the relaxed stochastic programming when accepting the group at the chosen row versus rejecting it. This evaluation allows us to assess the potential revenues and make the final decision. Simultaneously, after this calculation, we can generate a new seat planning according to Algorithm 2. Specifically, after we plan to assign the arriving group in a specific row, we determine whether to place the arriving group in the row based on the values of the stochastic programming problem. For the objective values of the relaxed stochastic programming, we consider the potential revenues that could result from accepting the current arrival, i.e., the Value of Acceptance (VoA), as well as the potential outcomes that could result from rejecting it, i.e., the Value of Rejection (VoR).

Suppose a group type i arrives at period t. The set of scenarios at period t is denoted as Ω^t. The available supply at period t before making the decision is $\mathbf{L}_r = (L_1, \ldots, L_N)$. The VoR is the value of RSSP with the scenario set Ω^t and the capacity $\mathbf{L}_r$ when we reject group type i at period t, denoted as RSSP$(\mathbf{L}_r, \Omega^t)$. If we plan to accept group type i in row j, we need to assign seats from row j to group type i. Let $\mathbf{L}_a = (L_1, \ldots, L_j - n_i, \ldots, L_N)$, then the VoA is calculated by RSSP$(\mathbf{L}_a, \Omega^t)$.

In each period, we can calculate the relaxed stochastic programming values only twice: once for the acceptance option (VoA) and once for the rejection option (VoR). By comparing the values of VoA and VoR, we can determine whether to accept or reject the group arrival. The decision will be based on selecting the option with the higher expected value, i.e., if the VoA is larger than the VoR, we accept the arrival; if the VoA is less than the VoR, we will reject the incoming group.

By combining the group-type control strategy with the evaluation of relaxed stochastic programming 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 groups, as well as determine the appropriate row for the assignment when acceptance is made. By considering both computation time savings and potential revenues, we can optimize the overall performance of the seat assignment process.

6.2.4 Regenerate The Seat Planning

To optimize computational efficiency, it is not necessary to regenerate the seat planning for every period. Instead, we can employ a more streamlined approach. Considering that largest group type can meet the needs of all smaller group types, thus, if the supply for the largest group type diminishes from one to zero, it becomes necessary to regenerate the seat planning. This avoids rejecting the largest group due to infrequent regenerations. Another situation that requires seat planning regeneration is when we determine whether to assign the arriving group seats planned for the larger group. In such case, we can obtain the corresponding seat planning after solving the relaxed stochastic programmings. By regenerating the seat planning in these situations, we can achieve real-time seat assignment while minimizing the frequency of planning updates.

The algorithm is shown below.

Algorithm 4: Dynamic Seat Assignment

```
$\boldsymbol{X} = [X_1, \ldots, X_M]$;
for $t = 1, \ldots, T$ do
    Observe group type $i'$;
    if $X_{i'} > 0$ then
        Find row $k$ such that $H_{ki'} > 0$ according to tie-breaking rule;
        Accept group type $i$ in row $k$, update $L_k$, $H_{ki'}$, $X_{i'}$;
        if $i' = M$ and $X_M = 0$ then
            Generate seat planning $\boldsymbol{H}$ 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
            Find row $k$ such that $H_{k\hat{i}^*} > 0$ according to the tie-breaking rule;
            Calculate the VoA under scenario $\Omega_A^t$ and the VoR under scenario $\Omega_R^t$;
            if $VoA \geq VoR$ then
                Accept group type $i'$, update $L_k$;
                Regenerate $\boldsymbol{H}$ from Algorithm 2, update the corresponding $\boldsymbol{X}$;
            else
                Reject group type $i'$;
                Regenerate $\boldsymbol{H}$ from Algorithm 2, update the corresponding $\boldsymbol{X}$;
            end
        else
            Reject group type $i'$;
        end
    end
end
```

7 Computational Experiment

We carried out several experiments, including analyzing the performances of different policies, evaluating the impact of implementing social distancing. In the experiment, we set the following parameters. The seat layout consists of 10 rows, each with the same size of 21 seats. To account for social distancing measures, one seat is designated as one dummy seat, i.e., $\delta = 1$. The group sizes considered range from 1 to 4 people, i.e., $M = 4$. In our experiments, we simulate the arrival of exactly one group in each period, i.e., $p_0 = 0$. The average number of people per period, denoted as γ, can be expressed as $\gamma = p_1 \cdot 1 + p_2 \cdot 2 + p_3 \cdot 3 + p_4 \cdot 4$, where p_1, p_2, p_3, and p_4 represent the probabilities of groups with one, two, three, and four individuals, respectively. We assume that p_4 always has a positive value.

7.1 Performances of Different Policies

In this section, we compare the performance of five assignment policies to the optimal one, which can be obtained by solving the deterministic model after observing all arrivals. The policies under examination are the dynamic seat assignment policy, DP-based heuristic, bid-price control, booking limit control and FCFS policy.

Parameters Description

We give the description of the parameters that will be used in the numerical results. M $= 4$, $\delta = 1$, N $= 10$, $L_j = 21, j \in \mathcal{N}$. Consider three sets of probability distributions with the same expectation of demand each period: D1: [0.25, 0.25, 0.25, 0.25], D2: [0.25, 0.35, 0.05, 0.35], D3: [0.15, 0.25, 0.55, 0.05].

These distributions have a common mean $\gamma = 2.5$, ensuring that the expected number of people for each period remained consistent. Each entry in the result columns is the average of 100 instances. The number of scenarios is $|\Omega| = 1000$. To assess the effectiveness of different policies across varying demand levels, we conducted experiments spanning a range of 60 to 100 periods. When the total number of periods, T, is set to 60, the expected total demand is $T \cdot \gamma = 150$. This expected demand would, on average, require $150 \times \frac{\gamma+\delta}{\gamma} = 210$ seats to accommodate. As the value of T increases, the total expected demand will eventually exceed the available supply. For example, with $T = 70$, the expected demand would be $T \cdot \gamma = 175$, which would require an average of $175 \times \frac{\gamma+\delta}{\gamma} = 245$ seats. By conducting experiments across this range of periods, we can observe how the different policies perform under both moderate and high-demand situations.

The following table presents the performance results of five different policies: DSA, DP, Bid-price, Booking, and FCFS, which stand for dynamic seat assignment, dynamic programming based heuristic, bid-price, booking-limit, and first come first served, respectively. The procedures for the last four policies are detailed in the appendix 9. Each entry in the table represents the average performance across 100 instances. Performance is evaluated by comparing the ratio of the number of accepted people under each policy to the number of accepted people under the optimal policy, which assumes complete knowledge of all incoming groups before making seat assignments.

We can find that DSA is better than DP-based heuristic, bid-price policy and booking limit policy consistently, and FCFS policy works worst. DP-based heuristic and bid-price policy can only make the decision to accept or deny, cannot decide which row to assign the group to. Booking limit policy does not consider satisfying the group demand with the larger planning. FCFS accepts groups in sequential order until the capacity cannot accommodate more.

The performance of DSA, DP-based heuristic, and bid-price policies follows a pattern where it initially decreases and then gradually improves as T increases. When T is small, the demand for capacity is generally low, allowing these policies to achieve relatively optimal performance. However, as T increases, it becomes more challenging for these policies to consistently achieve a perfect allocation plan, resulting in a decrease in performance. Nevertheless, as T continues to grow, these policies tend to accept larger groups, thereby narrowing the gap between their performance and the optimal value. Consequently, their

Table 1: Performances of Different Policies

Distribution	T	DSA (%)	DP (%)	Bid (%)	Booking (%)	FCFS (%)
D1	60	99.12	98.42	98.38	96.74	98.17
	70	98.34	96.87	96.24	97.18	94.75
	80	98.61	95.69	96.02	98.00	93.18
	90	99.10	96.05	96.41	98.31	92.48
	100	99.58	95.09	96.88	98.70	92.54
D2	60	98.94	98.26	98.25	96.74	98.62
	70	98.05	96.62	96.06	96.90	93.96
	80	98.37	96.01	95.89	97.75	92.88
	90	99.01	96.77	96.62	98.42	92.46
	100	99.23	97.04	97.14	98.67	92.00
D3	60	99.14	98.72	98.74	96.61	98.07
	70	99.30	96.38	96.90	97.88	96.25
	80	99.59	97.75	97.87	98.55	95.81
	90	99.53	98.45	98.69	98.81	95.50
	100	99.47	98.62	98.94	98.90	95.25

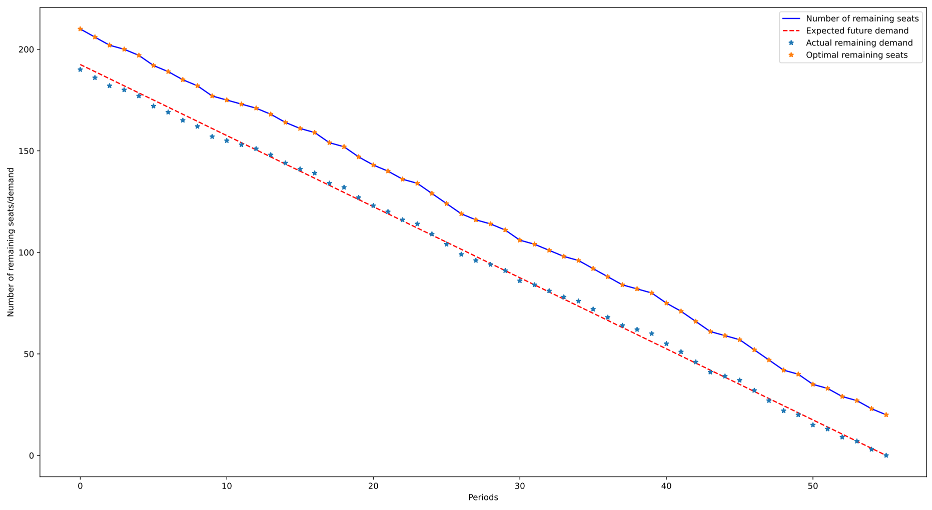
performances improve. In contrast, the booking limit policy shows improved performance as T increases because it reduces the number of unoccupied seats reserved for the largest groups. When T increases to infinity, DSA can always generate the largest pattern in each row, thus, the performance will converge to 100% compared to the optimal solution.

The performance of the policies can vary based on different probabilities. For different probability distributions listed, DSA performs more stably and consistently for the same demand under different probabilities, while for DP and bid price, their performance fluctuates more.

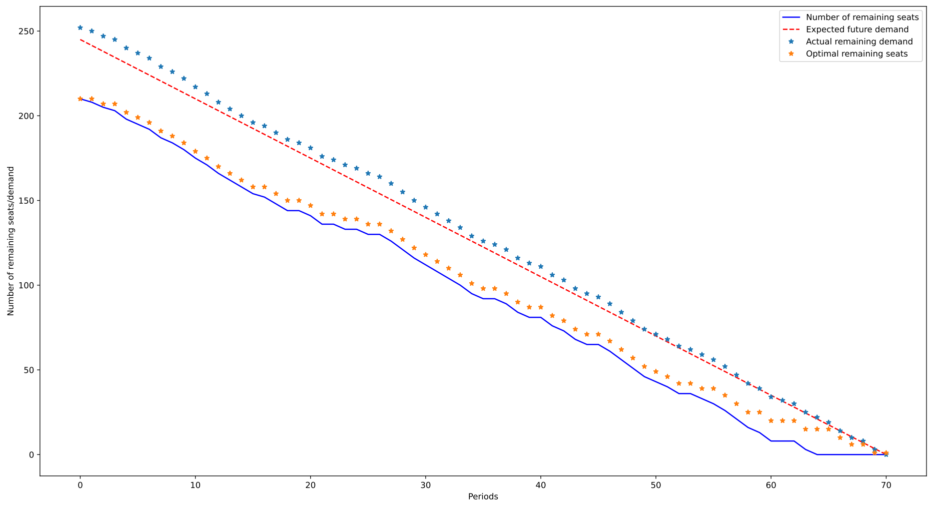
According to the results of policies, we can develop an easy-to-implement policy. When $T \leq \frac{\sum_j L_j}{\sum_i i p_i + 1}$, we accept the groups based on the first come first served policy. When $T > \frac{\sum_j L_j}{\sum_i i p_i + 1}$, we adopt the booking-limit policy, i.e., assign the seats according to the seat planning obtained from problem (1).

Analysis of DSA

We explore the arrival path of one instance under DSA and the optimal solution. The figures show two arrival paths when $T = 55$ and $T = 70$ at the even probability distribution. In the figures, we plot four lines over periods, number of remaining seats, the expected future demand, optimal remaining seats and optimal remaining demand. The horizontal parts of remaining seats represent the rejection at the period. We can observe that when the demand is larger than supply (T =70), even at the beginning we still reject the group. When the demand is lower than supply (T =55), we will accept all groups.



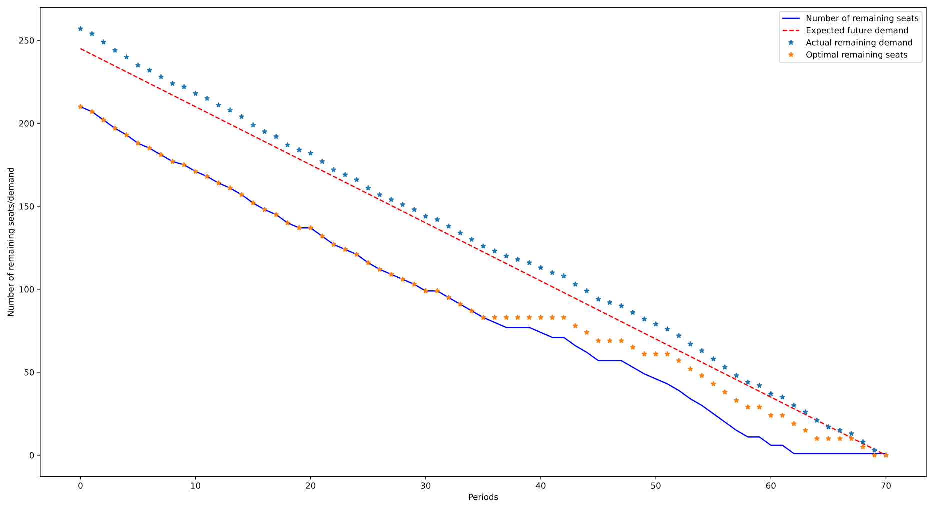
(a) T = 55



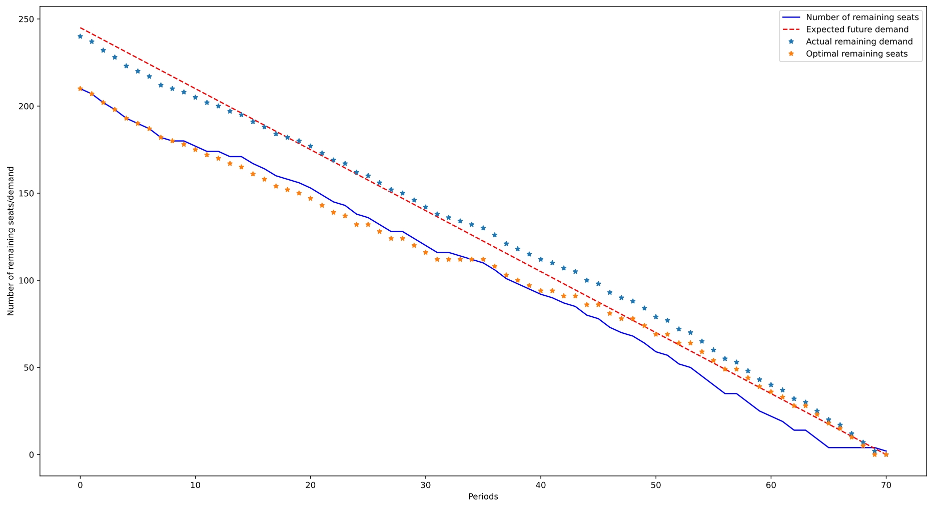
(b) T = 70

Figure 3: Arrival paths

We also examine the situation where the actual demand fluctuates around the expected demand.



(a)



(b)

Figure 4: Tow instances when T= 70

Based on the figures provided, we can draw several key conclusions:

1. Even at the very beginning of the time period, both DSA approach and the optimal solution reject some groups, as indicated by the horizontal segments in the lines.

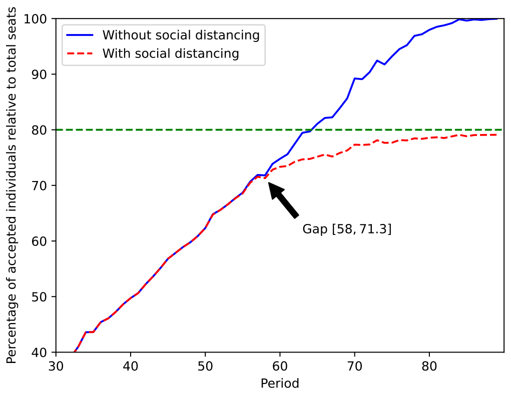
2. DSA approach is inclined to accept groups earlier compared to the optimal solution. This is because DSA makes decisions based on the expected future demand, rather than the actual remaining demand.

3. When the actual remaining demand is lower than the expected demand, the optimal remaining seats will be lower than that of DSA approach. Conversely, when the actual remaining demand is higher than expected, the optimal remaining seats will be higher than DSA.

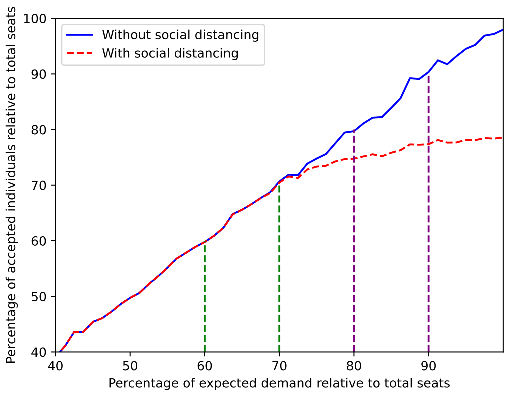
7.2 Impact of Social Distancing under Even Probability Distribution

Now, we explore the impact of social distance as the demand increases. Specifically, we consider the even probability distribution: $[0.25, 0.25, 0.25, 0.25]$. In the next section, we also consider other distributions. Here, T varies from 30 to 90, the step size is 1. Other parameters are set the same as before.

The figure below displays the outcomes of groups who were accepted under two different conditions: with social distancing measures and without social distancing measures. For the former case, we employ DSA to obtain the results. In this case, we consider the constraints of social distancing and optimize the seat allocation accordingly. For the latter case, we adopt the optimal solution assuming that all arrivals are known and that there are no social distancing constraints. The occupancy rate at different demands is calculated as the mean of these 100 instances. The figures depicting the results are presented below. The difference between these two figures is the x-axis, the left one is period, while the right one is the percentage of expected demand relative to total seats.



(a)



(b)

Figure 5: The occupancy rate over demand

Intuitively, when demand is low, we will accept all arrivals, and there will be no difference in the number of accepted individuals whether we implement social distancing or not. The interesting case is when the difference in the number of accepted individuals due to social distancing starts to occur.

Let T denote the total number of periods. We use $E(T; DSA)$ to denote the expected number of people accepted across 100 instances by DSA with one seat as social distancing. Similarly, we use $E(T; OPT)$ to denote the expected number of people accepted across 100 instances by the optimal solution when there is no social distancing. The gap point $\tilde{T}$ is defined as the lowest point that satisfies $E(\tilde{T}; DSA) + 1 \leq E(\tilde{T}; OPT)$. This period marks the threshold where the difference in the number of accepted individuals due to social distancing becomes significant. Additionally, the corresponding occupancy rate at the gap point is denoted by $\Gamma(\tilde{T})$.

Under the even probability distribution, we obtain the gap point is 58, the corresponding occupancy rate is 71.3%. To better analyze the situation under different demands, we plot the second figure. When the capacity is sufficient, in this case when the expected demand is less than 71.3%, the outcome remains unaffected by the implementation of social distancing measures. When the expected demand is larger than 71.3%, the difference between the outcomes with and without social distancing measures becomes more pronounced. As the expected demand continues to increase, both situations reach their maximum capacity acceptance. At this point, the gap between the outcomes with and without social distancing measures begins to converge. For the social distancing situation, according to Proposition 2, when the largest pattern is assigned to each row, the resulting occupancy rate is $\frac{16}{20} = 80\%$, which is the upper bound of occupancy rate. Therefore, the policy requires an occupancy rate larger than 80% will be invalid.

7.3 Estimation of Gap Points

To estimate the gap point, we aim to find the maximum period such that all the groups can be assigned into the seats during these periods, i.e., for each group type i, we have $\sum_j x_{ij} \geq d_i$, where x_{ij} is the number of group type i in row j. 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 let $\gamma = \sum_i i p_i$ represent the average number of people arriving in each period. From this, we can derive the inequality $T \leq \frac{\tilde{L}}{\gamma+\delta}$. Therefore, the upper bound for the expected maximum period is given by $T' = \frac{\tilde{L}}{\gamma+\delta}$.

Assuming that we accept all incoming groups within T' periods, filling all the available seats, the corresponding occupancy rate at this period 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 maximum period will be smaller than T' because it is 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 rate compared to the ideal assumption. Similarly, we can estimate the corresponding occupancy rate as $y_2 = c_2\frac{\gamma}{\gamma+\delta}\frac{\tilde{L}}{\tilde{L}-N\delta}$, where c_2 is a discount rate for the occupancy rate compared to the ideal scenario.

To analyze the relation between the increment of 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 the increment of γ, along with the corresponding estimations. For each probability combination, we considered 100 instances and plotted the gap point as blue points. Additionally, the occupancy rate at the gap point is represented by red points.

To provide estimations, we utilize the equations $y_1 = \frac{c_1\tilde{L}}{\gamma+\delta}$ (blue line in the figure) and $y_2 = c_2\frac{\gamma}{\gamma+\delta}\frac{\tilde{L}}{\tilde{L}-N\delta}$ (orange line in the figure), which are fitted to the data. These equations capture the relation between the gap point and the increment of γ, allowing us to approximate the values. By examining the relation between the gap point and the increment of γ, we can find that γ can be used to estimate gap point.

The fitting values of c_1 and c_2 can be affected by different seat layouts. To investigate this impact, we conduct several experiments using different seat layouts, specifically with the number of rows $\times$ the number of seats configurations set as 10×16, 10×21, 10×26 and 10×31. Similarly, we perform an analysis using a sample of 100 probability combinations, each with a mean equal to γ. The values of γ range from 1.5 to 3.4. We employed an Ordinary Least Squares (OLS) model to fit the data and derive

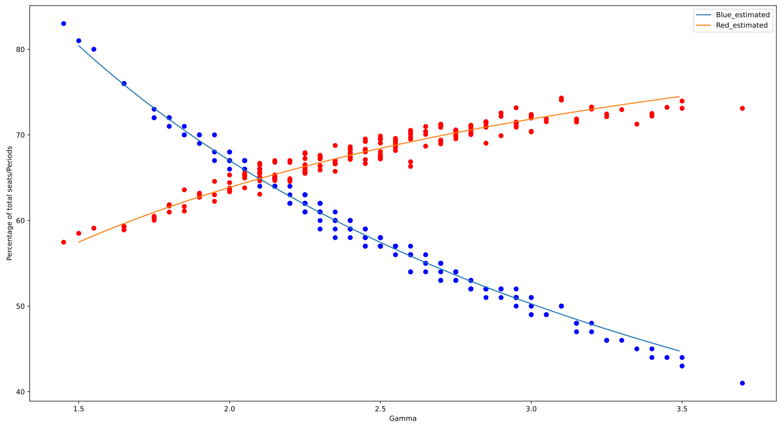


Figure 6: Gap points and their estimation under 200 probabilities

the parameter values. The goodness of fit is assessed using the R-square values, which are found to be 1.000 for all models, indicating a perfect fit between the data and the models.

The results of the estimation of c_1 and c_2 are presented in the table below:

Table 2: Fitting values of c_1 and c_2

Seat layout(# of rows $\times$ # of seats)	Fitting Values of c_1	Fitting Values of c_2
10 $\times$ 11	0.909 $\pm$ 0.013	89.89 $\pm$ 1.436
10 $\times$ 16	0.948 $\pm$ 0.008	94.69 $\pm$ 0.802
10 $\times$ 21	0.955 $\pm$ 0.004	95.44 $\pm$ 0.571
10 $\times$ 26	0.966 $\pm$ 0.004	96.23 $\pm$ 0.386
10 $\times$ 31	0.965 $\pm$ 0.003	96.67 $\pm$ 0.434
10 $\times$ 36	0.968 $\pm$ 0.003	97.04 $\pm$ 0.289

As the number of seats in each row increases, the fitting values of c_1 and c_2 will gradually increase.

7.4 Impact of Social Distancing under Different Demands

We present a table that shows the occupancy rate differences between DSA approach with social distancing and the optimal solution without social distancing for different expected demand levels (120, 140, 160, 180, 200).

Table 3: Gap points and occupancy rate differences under different demands of different gammas

γ	$\tilde{T}$	$\Gamma(\tilde{T})$	$\Delta\Gamma(T)$ under different demands				
			120	140	160	180	200
1.9	69	65.52	0	2.12	9.61	18.61	23.65
2.1	64	67.74	0	1.07	8.25	15.81	22.05
2.3	61	69.79	0	0.80	6.82	14.32	20.01
2.5	57	70.89	0	0.16	5.02	12.68	19.14
2.7	53	71.28	0	0.04	4.23	11.92	17.99

Since the gap points of different probability distributions with the same γ show little variation, we use the following probability distributions for γ values ranging from 1.9 to 2.7: [0.45, 0.35, 0.05, 0.15], [0.35, 0.35, 0.15, 0.15], [0.35, 0.25, 0.15, 0.25], [0.3, 0.2, 0.2, 0.3], [0.25, 0.15, 0.25, 0.35]. For each distribution, we simulate 100 instances to calculate the occupancy rate. As the value of γ increases, the gap point will decrease and the corresponding occupancy rate will increase. Consequently, the occupancy rate increases due to the allocation of seats to larger groups. The percentage difference is negligible when the demand is small, but it becomes more significant as the demand increases.

We examine the impact of implementing social distancing on the occupancy rate and explore strategies to minimize revenue loss. Consider the situation where the gap point is $\tilde{T}$ and is determined by the parameters δ, γ, M, and $\tilde{L}$. The corresponding occupancy rate is $\Gamma(\tilde{T})$. When the actual number of people (demand) is less than $\tilde{L} \cdot \Gamma(\tilde{T})$, implementing social distancing does not affect the revenue. However, if the actual number of people exceeds $\tilde{L} \cdot \Gamma(\tilde{T})$, enforcing social distancing measures will lead to a reduction in revenue. The extent of this loss can be assessed through simulations by using the specified parameters. To mitigate the potential loss, the seller can increase the value of γ. This can be achieved by implementing certain measures, such as setting a limit on the number of single-person groups or allowing for larger group sizes. The government can set a requirement for a higher occupancy rate limit, for example, for family movies in cinemas, γ will be relatively large, so a higher occupancy rate limit can be set. By doing so, the objective is to minimize the negative impact of social distancing while maximizing revenue within the constraints imposed by the occupancy rate.

8 Conclusion

Our paper focuses on the problem of dynamic seat assignment with social distancing in the context of a pandemic. To tackle this problem, we propose a scenario-based stochastic programming approach to obtain a seat planning that adheres to social distancing constraints. We utilize the benders decomposition method to solve this model efficiently, leveraging its well-structured property. However, solving the integer programming formulation directly can be computationally prohibitive in some cases. Therefore, in practice, we consider the linear programming relaxation of the problem and devise an approach to obtain the seat planning, which consists of full or largest patterns. In our approach, seat planning can be seen as the supply for each group type. We assign groups to seats when the supply is sufficient. However, when the supply is insufficient, we employ the dynamic seat assignment policy to make decisions on whether to accept or reject group requests.

We conducted several experiments to investigate various aspects of our approach. These experiments include analyzing different policies for dynamic seat assignment and evaluating the impact of implementing social distancing. 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 performance.

Building upon our policies, we further evaluated the impact of implementing social distancing. By defining the gap point as the period at which the difference between applying and not applying social distancing becomes evident, we established a relationship between the gap point and the expected number of people in each period. We observed that as the expected number of people in each period increased, the gap point occurred earlier, resulting in a higher occupancy rate at the gap point.

Overall, our study emphasizes the operational significance of social distancing in the seat allocation and offers a fresh perspective for sellers and governments to implement seat assignment mechanisms that effectively promote social distancing during the pandemic.

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

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

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

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

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

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

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

[17] 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.

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

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

[20] 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.

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

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

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

[24] 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

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 problem (1) is:

$$\begin{aligned}\min \quad & \sum_{i=1}^{M} d_i z_i + \sum_{j=1}^{N} L_j \beta_j \\ \text{s.t.} \quad & z_i + \beta_j n_i \geq (n_i - \delta), \quad i \in \mathcal{M}, j \in \mathcal{N} \\ & z_i \geq 0, i \in \mathcal{M}, \beta_j \geq 0, j \in \mathcal{N}.\end{aligned} \tag{16}$$

In (16), 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 above the sum of the bid prices of the seats it uses. Thus, if its revenue is more than its opportunity costs, i.e., $i - \beta_j n_i \geq 0$, 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* (16) *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 means 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 can result in a large number of empty seats.

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

Algorithm 5: Bid-price Control Algorithm

```
1  for $t = 1, \ldots, T$ do
2      Observe group type $i$;
3      Solve the LP relaxation of problem (1) with $d_i^t = (T-t) \cdot p_i$ and $\mathbf{L}^t$;
4      Obtain $\tilde{i}$ such that the aggregate optimal solution is $x e_{\tilde{i}} + \sum_{i=\tilde{i}+1}^{M} d_i e_i$;
5      if $i \geq \tilde{i}$ and $\max_j L_j^t \geq n_i$ then
6          Accept the group and assign the group to row $k$ such that $L_k^t \geq n_i$;
7      else
8          Reject the group;
9      end
10 end
```

Booking Limit Control

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

In this policy, we replace the real demand by the expected one and solve the corresponding static problem using the expected demand. Then for every type of requests, we only allocate a fixed amount according to the static solution and reject all other exceeding requests. When we solve the linear relaxation of problem (1), the aggregate optimal solution is the limits for each group type. Interestingly, the bid-price control policy is found to be equivalent to the booking limit control policy.

When we solve problem (1) directly, we can develop the booking limit control policy.

Algorithm 6: Booking Limit Control Algorithm

```
1  for $t = 1, \dots, T$ do
2      Observe group type $i$;
3      Solve problem (1) with $d_i^t = (T - t) \cdot p_i$ and $\mathbf{L}^t$;
4      Obtain the optimal solution, $x_{ij}^*$ and the aggregate optimal solution, $\mathbf{X}$;
5      if $X_i > 0$ then
6          Accept the group and assign the group to row $k$ such that $x_{ik} > 0$;
7      else
8          Reject the group;
9      end
10 end
```

DP-based Heuristic

To simplify the complexity of the original dynamic programming problem, we can consider a simplified version by relaxing all rows to a single row with the same total capacity, denoted as $\tilde{L} = \sum_{j=1}^{N} L_j$. With this simplification, we can make decisions for each group arrival based on the relaxed dynamic programming. By relaxing the rows to a single row, we aggregate the capacities of all individual rows into a single capacity value. This allows us to treat the seat assignment problem as a one-dimensional problem, reducing the computational complexity. Using the relaxed dynamic programming approach, we can determine the seat assignment decisions for each group arrival based on the simplified problem.

Let u denote the decision, where $u^t = 1$ if we accept a request in period t, $u^t = 0$ otherwise. Similar to the DP in section 4, the DP with one row can be expressed as:

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

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

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

Algorithm 7: DP-based Heuristic Algorithm

```
1  Calculate $V^t(l)$, $\forall t = 2, \ldots, T; \forall l = 1, \ldots, L$;
2  $l^1 \leftarrow L$;
3  for $t = 1, \ldots, T$ do
4      Observe group type $i$;
5      if $V^{t+1}(l^t) \leq V^{t+1}(l^t - n_i) + i$ then
6          Accept the group and assign the group to an arbitrary row $k$ such that $L_k^t \geq n_i$;
7      else
8          Reject the group;
9      end
10 end
```

First Come First Served (FCFS) Policy

For dynamic seat assignment for each group arrival, the intuitive but trivial method will be on a first-come-first-served basis. Each accepted request will be assigned seats row by row. If the capacity of a row is insufficient to accommodate a request, we will allocate it to the next available row. If a subsequent request can fit exactly into the remaining capacity of a partially filled row, we will assign it to that row immediately. Then continue to process requests in this manner until all rows cannot accommodate any groups.

Algorithm 8: FCFS Policy Algorithm

```
1  for $t = 1, \ldots, T$ do
2      Observe group type $i$;
3      if $\exists k$ such that $L_k^t \geq n_i$ then
4          Accept the group and assign the group to row $k$;
5      else
6          Reject the group;
7      end
8  end
```

Tie-Breaking Rule

These policies will encounter ties when the group can be assigned to two or more rows. For the booking limit control, we assign the group according to the seat planning. The same tie-breaking rule used in the DSA approach can be applied for the booking limit control policy. For the other policies besides the booking limit control, we adopt the following rule for assigning groups to rows. We prioritize assigning the group to rows that have at least n_M seats available. If the number of remaining seats for all rows are less than n_M, we assign the group to an arbitrary row that has enough capacity to accommodate the group.

Proof

(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. We can utilize a greedy approach to construct a pattern, denoted as* $\boldsymbol{h}_g$*, by following the steps outlined below. This approach aims to generate a pattern that maximizes the number of people accommodated within the given constraints.*

- *Begin by selecting the maximum group size, denoted as* n_M*, as many times as possible to fill up the available seats in the row.*
- *Allocate the remaining seats (if possible) in the row to the group with the corresponding size.*

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:*

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

Breaking down the above inequality into two cases:

(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 pattern,* $\boldsymbol{h}_g$*, is a largest pattern. The maximum number of people that can be accommodated in the largest pattern is* $qM + \max\{r - \delta, 0\}$*.*

□

(Proof of Proposition 1)**.** *Treat the groups as the items, the rows as the knapsacks. There are* M *types of items, the total number of which is* $K = \sum_i d_i$*, each item* k *has a profit* p_k *and weight* w_k*.*

Then this Integer Programming is a special case of the Multiple Knapsack Problem (MKP). Consider the solution to the linear relaxation of (1)*. 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. Then the Dantzig upper bound* [8] *becomes* $u_{\text{MKP}} = \sum_{j=1}^{b-1} p_j + \left(\tilde{L} - \sum_{j=1}^{b-1} w_j\right) \frac{p_b}{w_b}$*. 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 4)**.** *In any optimal solution where one of the corresponding patterns is not full or largest, we have the flexibility to allocate the remaining unoccupied seats. These seats can be assigned to either a new seat planning or added to an existing seat planning. Importantly, since a group can utilize the seat planning of a larger group, the allocation scheme based on the original optimal solution will not affect the optimality of the solution. For each row, there are three situations to allocate the seats. First, when the rest seats can be allocated to the existing groups, then the corresponding pattern becomes a full pattern. Second, when all the existing groups are the largest groups and the rest seats cannot construct a new group, the pattern becomes the largest. Third, when all the existing groups are the largest groups and the rest seats can construct new groups, the rest seats can be used to construct the largest groups until there is no enough capacity, then the pattern becomes the largest. Finally, we can allocate the seats such that each row in the seat planning becomes either full or largest.* □

(Proof of Lemma 1)**.** *Note that* $\mathbf{f}^\intercal = [-\mathbf{1},\ \mathbf{0}]$ *and* $V = [W,\ I]$*. Based on this, we can derive the following inequalities:* $\boldsymbol{\alpha}^\intercal W \geq -\mathbf{1}$ *and* $\boldsymbol{\alpha}^\intercal I \geq \mathbf{0}$*. These inequalities indicate that the feasible region is nonempty and bounded. Moreover, let* $\alpha_0 = 0$*. From this, we can deduce that* $0 \leq \alpha_i \leq \alpha_{i-1} + 1$ *for* $i \in \mathcal{M}$*. Consequently, 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* $\alpha_i = \alpha_{i-1} + 1$*. Let* $\Delta d = d_\omega - d_0$*, then the elements of* Δd *will be a negative integer, positive integer and zero. When* $y_{i\omega}^+ = y_{i\omega}^- = 0$*, if* $i = M$*,* $\Delta d_M = 0$*, the value of objective function associated with* 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* $\boldsymbol{\alpha}_\omega^0$ *for each scenario. Then we have the* *following problem.*

$$\begin{aligned}\max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega\in\Omega} p_\omega z_\omega \\ s.t. \quad & \mathbf{n}\mathbf{x} \le \mathbf{L} \\ & (\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{d}_\omega \ge (\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{x}\mathbf{1} + z_\omega, \forall \omega \\ & \mathbf{x} \in \mathbb{N}^{M\times N}\end{aligned} \tag{17}$$

Problem (17) *reaches its maximum when* $(\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{d}_\omega = (\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{x}\mathbf{1} + z_\omega, \forall \omega$. *Substitute* z_ω *with these equations, we have*

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

Notice that $\mathbf{x}$ *is bounded by* $\mathbf{L}$, *then the problem* (17) *is bounded. Adding more constraints will not make the optimal value larger. Thus, RBMP is bounded.* □

(Proof of Proposition 3)**.** *First of all, we demonstrate the feasibility of problem* (2)*. Given the feasible seat planning* $\boldsymbol{H}$ *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* (2) *always has a feasible solution.*

Suppose there exists at least one pattern $\boldsymbol{h}$ *is neither full nor largest in the optimal seat planning obtained from problem* (2)*. Let* $\beta = L - \sum_i n_i h_i$, *and denote the smallest group type in pattern* $\boldsymbol{h}$ *by* k*. If* $\beta \ge n_1$, *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* $j = \min\{M, \beta + i\}$*. 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* (2) *always generate a seat planning composed of full or largest patterns.*

(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* $\beta_j \ge \frac{n_i - \delta}{n_i}$ *for* $i = 1, \ldots, \tilde{i}$*. Since* $\frac{n_i-\delta}{n_i}$ *increases with* i, *we have* $\beta_j \ge \frac{n_{\tilde{i}}-\delta}{n_{\tilde{i}}}$*. Consequently, we obtain* $z_i \ge n_i - \delta - n_i \frac{n_{\tilde{i}}-\delta}{n_{\tilde{i}}} = \frac{\delta(n_i - n_{\tilde{i}})}{n_{\tilde{i}}}$ *for* $i = h+1, \ldots, M$*.*

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