Dynamic Seat Assignment With Social Distancing

October 9, 2023

Abstract

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

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

1 Introduction

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

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

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

We aim to provide a practical tool for venues to optimize seat assignments while ensuring the safety of groups by proposing a seat assignment policy that takes into account social distancing requirements and the given seating layout. We strive to enable venues to implement social distancing measures effectively by providing a tailored solution that accommodates their specific seating arrangements and operational constraints.

To avoid confusion regarding the terms 'seat planning' and 'seat assignment', it is important to clarify the distinction between them. In our context, seat planning involves determining the arrangement of seats within a venue or space based on the size or layout of the room. This includes deciding on seats partition with social distancing. On the other hand, seat assignment involves the specific allocation of seats to individual attendees or groups based on seat availability. This task is typically performed when the seller needs to make a decision each time there is a request.

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

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

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

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

Our main contributions in this paper are summarized as follows:

First, this study presents the first attempt to consider the arrangement of seat assignments with social distancing under dynamic arrivals. While many studies in the literature highlight the importance of social distancing in controlling the spread of the virus, they often focus too much on the model and do not provide much insight into the operational significance behind social distancing [1, 11]. Recent studies have explored the effects of social distancing on health and economics, mainly in the context of aircraft [13, 27, 28]. Our study provides a new perspective to help the government adopt a mechanism for setting seat assignments to protect people 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 consider the stochastic demand situation where the demands of different group types are random. By using Benders decomposition methods, we can obtain the optimal linear solution.

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

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

2 Literature Review

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

2.1 Seat Planning with Social Distancing

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

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

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

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

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

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

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

2.2 Dynamic Seat Assignment

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

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

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

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

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

3 Problem Description

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

3.1 Seat Planning Problem with Social Distancing

We consider a layout comprising N rows, with each row containing S_j seats, where $j \in \mathcal{N} := \{1, 2, \ldots, N\}$. The seating arrangement is intended for various groups, where each group consists of no more than M individuals. There are M distinct group types, denoted by group type i containing i people, where $i \in \mathcal{M} := \{1, 2, \ldots, M\}$. Represented by a demand vector $\mathbf{d} = [d_1, \ldots, d_M]$, each element d_i represents the number of group type i.

In order to comply with the social distancing requirements, individuals from the same group may sit together, 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 that there are empty seat(s) between them and adjacent groups. Importantly, the seating arrangement of different rows does not affect each other, meaning that individuals from one group can be seated directly behind individuals from another group.

To incorporate the social distancing requirements into the seat planning process, we adjust the original group sizes by adding δ. Consequently, the new size of group type i is denoted as $n_i = i + \delta$, where $i \in \mathcal{M}$. Similarly, to accommodate the adjusted group sizes, the seat layout is modified by adding δ to the length of each row. Thus, $L_j = S_j + \delta$ represents the length of row j, where S_j 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.

The deterministic seat planning problem is formulated below, with the objective of maximizing the number of people accommodated.

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

This seat planning problem can be viewed as a special case of the multiple knapsack problem. Given the small size of the problem, it is relatively easy to obtain the optimal solution.

To simplify the discussion, we will use a vector $(t_1, \ldots, t_i, \ldots, t_M)$ to represent the pattern, where t_i corresponds to the number of group type i. To quantify the effect of each pattern, we introduce the notion of loss, denoting the number of unoccupied seats.

Lemma 1. *When given the length of a row, L, the social distancing, δ, the adjusted size of the largest group allowed, n_M, the loss of the largest pattern is $\lfloor \frac{L}{n_M} \rfloor \delta - \delta + g(L - \lfloor \frac{L}{n_M} \rfloor n_M)$, where $g(r) = 0$ if $r > \delta$, and $g(r) = r$ if $r \leq \delta$.*

The loss provides a measure of the number of people who cannot be seated due to the implementation of social distancing constraints. By examining the losses associated with different patterns, we can assess the effectiveness of various seat planning configurations with respect to accommodating the desired number of individuals while adhering to social distancing requirements.

Definition 1. *Given the length of row and the largest group size, we call the pattern with the minimal loss as the largest patterns. Additionally, we refer to patterns that have no empty seats, except for those reserved for social distancing purposes, as full patterns.*

The largest patterns are characterized by having the least number of people unable to be seated due to social distancing requirements. The full patterns are designed to maximize seating capacity while still maintaining the necessary spacing between groups.

In most cases, we observe that the optimal solution tends to consist of rows with either full patterns or the largest patterns. By distinguishing the largest and full patterns from other configurations, we can gain valuable insights into the most efficient seat planning strategies that prioritize accommodating the maximum number of people while adhering to social distancing guidelines.

In scenarios where the demand for seats is high, it is advantageous to adopt the largest pattern, as it allows for the accommodation of a larger number of individuals. The largest pattern becomes particularly beneficial when the demand exceeds the capacity of full patterns. In scenarios where the demand is moderate, adopting the full pattern becomes more feasible. The full pattern maximizes seating capacity by utilizing all available seats, except those required for social distancing measures. Overall, by considering both the largest and full patterns, we can optimize seat planning configurations to efficiently accommodate a significant number of individuals while adhering to social distancing guidelines.

Example 1. *Suppose that the social distancing requirement is one seat, and there are four types of groups. In this case, the new sizes of the groups would be 2, 3, 4, and 5, respectively. Additionally, the length of a single row is determined to be $L = 21$. Then the loss of the largest pattern is $\lfloor \frac{21}{5} \rfloor - 1 + g(1) = 5$.*

We find that the patterns (0, 0, 0, 4), (0, 0, 4, 1), and (0, 2, 0, 3) are the largest patterns with the same loss value of 5. However, it's important to note that the largest pattern may not necessarily be a full pattern. For instance, the pattern (0, 0, 0, 4) is the largest pattern in terms of accommodating the maximum number of individuals, but it does not meet the requirement of fully utilizing all available seats since $4 \times 5 \neq 21$. Conversely, a full pattern may not always be the largest pattern. For example, the pattern (1, 1, 4, 0) is considered a full pattern since it utilizes all seats, but its loss value is 6.

The optimal solution to the seat planning problem can be complex. However, the LP relaxation of problem (1) has nice properties.

Lemma 2. *For the LP relaxation of problem (1), there exists h such that the optimal solutions $x_{ij}^* = 0$ when $i < h$; $\sum_j x_{ij}^* = d_i$, when $i > h$; $\sum_j x_{ij}^* = (L - \sum_{i=h+1}^{M} d_i n_i)/n_h$, when $i = h$.*

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

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

3.2 Dynamic Seat Assignment with Social Distancing

We address the problem of dynamic seat assignment with social distancing, which involves the real-time allocation of seats to incoming groups while ensuring adherence to social distancing guidelines. The decision-maker must make accept or reject decisions for each group and assign them to available seats in rows, while guaranteeing the required spacing between groups. Recalling the seat layout, it consists of N rows, with each row having a length denoted by L_j. Additionally, there are M distinct group types, where each group type i consists of i individuals.

To model this problem, we adopt a discrete-time framework. The time is divided into T periods, with each period representing the arrival of exactly one group. Time is discretized as $1, \ldots, T$, where the first period corresponds to the beginning of the selling horizon and the last period represents the end. In each period t, a group of size i arrives with a probability denoted as p_i. It is assumed that the arrivals of different group types are independent. During each period, the decision-maker determines whether to accept or reject the incoming group and assigns them to a specific row. Once seats are confirmed and assigned to a group, they cannot be changed.

To keep track of the remaining capacity of rows, we utilize a vector $\mathbf{L} = (l_1, l_2, \ldots, l_N)$, where l_j represents the number of remaining seats in row j. Specifically, we define $V_t(\mathbf{L})$ as the maximum expected value at period t, given the current capacity $\mathbf{L}$. Additionally, we introduce the decision variable $u_{i,j}$, where $u_{i,j}(t) = 1$ denotes the decision to accept group type i in row j at period t, while $u_{i,j}(t) = 0$ indicates the rejection of that group type in row j at that period. The decision set is defined as: $U(\mathbf{L}) = \{u_{i,j} \in \{0, 1\}, \forall i, j | \sum_{j=1}^{N} u_{i,j} \leq 1, \forall i; n_i u_{i,j} \mathbf{e}_j^\top \leq \mathbf{L}, \forall i, j\}$. Essentially, $U(\mathbf{L})$ represents the feasible assignment decisions, where each group type i can be assigned to at most one row, and the corresponding capacity requirements are satisfied.

The dynamic programming formula for this problem can be expressed as:

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

Here, $\mathbf{e}_j$ represents an N-dimensional unit row vector with j-th element being 1. In the above formula, the decision set $U(\mathbf{L})$ represents the feasible set of decisions for a given seat availability $\mathbf{L}$.

Initially, we have $\mathbf{L} = (L_1, L_2, \ldots, L_N)$. The objective function is $V_1(\mathbf{L})$. By applying the dynamic programming formula, we can recursively compute the optimal value function $V_t(\mathbf{L})$, representing the maximum expected value at time t for a given seat availability $\mathbf{L}$. This approach allows us to make optimal decisions regarding group acceptance and seat assignment in order to maximize the overall value while considering social distancing constraints and group arrival probabilities. However, this leads to the curse of dimensionality due to the numerous seat planning combinations. To avoid this complexity, we propose an approach that directly targets the final seat planning and then formulate a policy to assign arriving groups. To obtain the final seat planning firstly, we develop the scenario-based stochastic programming.

4 Seat Planning Composed of Full or Largest Patterns

This section focuses on obtaining seat planning with available capacity. We begin by introducing a scenario-based stochastic programming formulation. Due to its time-consuming nature, we reformulate it into the master problem and subproblem. We could obtain the optimal solution by implementing benders decomposition. However, in some cases, solving the IP directly remains still computationally prohibitive. Thus, we can consider the LP relaxation first, then obtain a feasible seat planning by deterministic model. To fully utilize all seats, we construct a seat planning composed of full or largest patterns.

4.1 Scenario-based Stochastic Programming(SSP) 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 $\omega \in \Omega$. 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 value of people over all the scenarios, we propose a scenario-based stochastic programming.

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

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

equals x if $x > 0$, 0 otherwise. Specially, for the largest group type M, we have $y^+_{M\omega} = (\sum_{j=1}^N x_{ij} - d_{i\omega})^+$, $y^-_{M\omega} = (d_{i\omega} - \sum_{j=1}^N x_{ij})^+$.

Then we have the formulation of SSP:

$$\max \quad E_\omega \left[\sum_{i=1}^{M-1} (n_i - \delta)(\sum_{j=1}^N x_{ij} + y^+_{i+1,\omega} - y^+_{i\omega}) + (n_M - \delta)(\sum_{j=1}^N x_{Mj} - y^+_{M\omega}) \right] \tag{2}$$

$$\text{s.t.} \quad \sum_{j=1}^N x_{ij} - y^+_{i\omega} + y^+_{i+1,\omega} + y^-_{i\omega} = d_{i\omega}, \quad i = 1, \ldots, M-1, \omega \in \Omega \tag{3}$$

$$\sum_{j=1}^N x_{ij} - y^+_{i\omega} + y^-_{i\omega} = d_{i\omega}, \quad i = M, \omega \in \Omega \tag{4}$$

$$\sum_{i=1}^M n_i x_{ij} \leq L_j, j \in \mathcal{N} \tag{5}$$

$$y^+_{i\omega}, y^-_{i\omega} \in \mathbb{Z}_+, \quad i \in \mathcal{M}, \omega \in \Omega \tag{6}$$

$$x_{ij} \in \mathbb{Z}_+, \quad i \in \mathcal{M}, j \in \mathcal{N}. \tag{7}$$

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

By reformulating the objective function, we have

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

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

Remark 1. *For any i, ω, at most one of $y^+_{i\omega}$ and $y^-_{i\omega}$ can be positive. 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 (3) and (4) but increase the objective value when p_{ω_0} is positive. Thus, at most one of $y^+_{i\omega}$ and $y^-_{i\omega}$ can be positive.*

Let $\mathbf{n} = (n_1, \ldots, n_M)$, $\mathbf{L} = (L_1, \ldots, L_N)$ where s_i is the size of seats taken by group type i and L_j is the length of row j as we defined above. Then the constraint (5) can be expressed as $\mathbf{nx} \leq \mathbf{L}$.

The linear constraints associated with scenarios, i.e., constraints (3) and (4), can be written in a matrix form as

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

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

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

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

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

As we can find, this deterministic equivalent form is a large-scale problem even if the number of possible scenarios 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,

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

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 $\omega \in \Omega$,

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

Problem (9) stands for the subproblem and $z_\omega(\mathbf{x})$ is the optimal value of problem (9), together with the convention $z_\omega(\mathbf{x}) = \infty$ if the problem is infeasible.

4.2 Solve SSP by Benders Decomposition

At first, we generate a closed-form solution to problem (9). Then we obtain the solution to the linear relaxation of problem (8) by the delayed constraint generation. Based on the solution, we obtain a seat planning by deterministic model. Finally, we construct an integral seat planning composed of full or largest patterns.

4.2.1 Solve The Subproblem

Consider a $\mathbf{x}$ such that $\mathbf{nx} \le \mathbf{L}$ and $\mathbf{x} \ge 0$ and suppose that this represents the seat planning. Once $\mathbf{x}$ is fixed, the optimal decisions $\mathbf{y}_\omega$ can be determined by solving problem (9) for each ω.

Notice that the feasible region of the dual of problem (9) does not depend on $\mathbf{x}$. Let $\boldsymbol{\alpha}$ be the vector of dual variable. For each ω, we can form its dual problem, which is

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

Let $\mathbb{P} = \{\boldsymbol{\alpha} \in \mathbb{R}^M | \boldsymbol{\alpha}^\intercal \mathbf{V} \ge \mathbf{f}^\intercal\}$. We assume that $\mathbb{P}$ is nonempty and has at least one extreme point. Then, either the dual problem (10) has an optimal solution and $z_\omega(\mathbf{x})$ is finite, or the primal problem (9) is infeasible and $z_\omega(\mathbf{x}) = \infty$.

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

Lemma 3. *The feasible region of problem (10), $\mathbb{P}$, is bounded. In addition, all the extreme points of $\mathbb{P}$ are integral.*

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

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

Corollary 2. *The optimal value of the problem (9), $z_\omega(x)$, is finite and will be attained at extreme points of the set P. Thus, we have $z_\omega(x) = \min_{\boldsymbol{\alpha} \in \mathcal{O}} \boldsymbol{\alpha}^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1})$.*

When we are given $\mathbf{x}^*$, the demand that can be satisfied by the 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:

$$\begin{aligned} y^-_{M\omega} &= (d_{M\omega} - d_{M0})^+ \\ y^+_{M\omega} &= (d_{M0} - d_{M\omega})^+ \\ y^-_{i\omega} &= \left(d_{i\omega} - d_{i0} - y^+_{i+1,\omega}\right)^+, i = 1, \ldots, M-1 \\ y^+_{i\omega} &= \left(d_{i0} - d_{i\omega} + y^+_{i+1,\omega}\right)^+, i = 1, \ldots, M-1 \end{aligned} \tag{11}$$

For scenario ω, the optimal value is $\mathbf{f}^\intercal y_\omega$, then we need to find the dual optimal solution.

Theorem 1. *The optimal solutions to problem (10) are given by*

$$\begin{aligned} &\alpha_i = 0, i \in \mathcal{M} \quad \textit{if } y^-_{i\omega} > 0, y^+_{i\omega} = 0 \\ &\alpha_i = \alpha_{i-1} + 1, i \in \mathcal{M} \quad \textit{if } y^+_{i\omega} > 0, y^-_{i\omega} = 0 \\ &\alpha_i = 0, i = 1, \ldots, M-1 \quad \textit{if } y^-_{i\omega} = y^+_{i\omega} = 0, y^+_{i+1,\omega} > 0 \\ &0 \le \alpha_i \le \alpha_{i-1} + 1, i = 1, \ldots, M-1 \quad \textit{if } y^-_{i\omega} = y^+_{i\omega} = 0, y^+_{i+1,\omega} = 0 \\ &0 \le \alpha_i \le \alpha_{i-1} + 1, i = M \quad \textit{if } y^-_{i\omega} = y^+_{i\omega} = 0 \end{aligned} \tag{12}$$

We can use the forward method, calculating from $\alpha_{1\omega}$ to $\alpha_{M\omega}$, to obtain the value of α_ω instead of solving problem (10).

4.2.2 Delayed Constraint Generation

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

According to Corollary 1 and take into account the optimality cuts, we can conclude the BMP will have the form:

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

When substituting $\mathcal{O}$ with its subset, $\mathcal{O}^t$, the problem (13) becomes the Restricted Benders Master Problem(RBMP). To determine the initial $\mathcal{O}^t$, we have the following lemma.

Lemma 4. *RBMP is always bounded with at least any one optimality cut for each scenario.*

Given the initial $\mathcal{O}^t$, we can have the solution $\mathbf{x}_0$ and $\mathbf{z}^0 = (z_1^0, \ldots, z_S^0)$. Then $c^\intercal \mathbf{x}_0 + \sum_{\omega \in \Omega} p_\omega z_\omega^0$ is an upper bound of problem (13).

When $\mathbf{x}_0$ is given, the optimal solution, $\boldsymbol{\alpha}_\omega^1$, to problem (10) can be obtained according to Theorem 1. $z_\omega^{(0)} = \boldsymbol{\alpha}_\omega^1 (d_\omega - \mathbf{x}_0 \mathbf{1})$ and $(\mathbf{x}_0, \mathbf{z}^{(0)})$ is a feasible solution to problem (13) because it satisfies all the constraints. Thus, $\mathbf{c}^\intercal \mathbf{x}_0 + \sum_{\omega \in \Omega} p_\omega z_\omega^{(0)}$ is a lower bound of problem (13).

If for every scenario ω, the optimal value of the corresponding problem (10) is larger than or equal to z_ω^0, all contraints are satisfied, we have an optimal solution, $(\mathbf{x}_0, \mathbf{z}_\omega^0)$, to the BMP. Otherwise, add one new constraint, $(\boldsymbol{\alpha}_\omega^1)^\intercal (\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega$, to RBMP.

The steps of the algorithm are described as below,

Algorithm 1 The benders decomposition algorithm

Step 1. Solve IP (18) with all $\alpha_\omega^0 = \mathbf{0}$ for each scenario. Then, obtain the solution $(\mathbf{x}_0, \mathbf{z}^0)$.

Step 2. Set the upper bound $UB = c^\intercal \mathbf{x}_0 + \sum_{\omega \in \Omega} p_\omega z_\omega^0$.

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

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

Step 5. Solve the updated RBMP, obtain a new solution (x_1, z^1) and update UB.

Step 6. Repeat step 3 until $UB - LB < \epsilon$.(In our case, UB converges.)

From the Lemma 4, we can set $\boldsymbol{\alpha}_\omega^0 = \mathbf{0}$ initially in **Step 1**. Notice that only contraints are added in each iteration, thus LB and UB are both monotone. Then we can use $UB - LB < \epsilon$ to terminate the algorithm in **Step 6**.

After the algorithm terminates, we obtain the optimal $\mathbf{x}^*$. The demand that can be satisfied by the arrangement is $\mathbf{x}^*\mathbf{1} = \mathbf{d}_0 = (d_{1,0}, \ldots, d_{M,0})$. Solving problem (18) directly can be computationally challenging in some cases, so practically we first obtain the optimal solution to the LP relaxation of problem (8). From this solution, we generate an integral seat planning.

4.3 Obtain The Seat Planning Composed of Full or Largest Patterns

As we mentioned above, seat planning with full or largest patterns can accommodate more groups. Thus, we need to obtain the seat planning composed of full or largest patterns. Before that, we will discuss the deterministic model that can help to achieve the goal.

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

$$\begin{aligned} \max \quad & \sum_{i=1}^{M}\sum_{j=1}^{N}(n_i - \delta)x_{ij} - \sum_{i=1}^{M} y_i^+ \\ \text{s.t.} \quad & \sum_{j=1}^{N} x_{ij} - y_i^+ + y_{i+1}^+ + y_i^- = d_i, \quad i = 1, \ldots, M-1, \\ & \sum_{j=1}^{N} x_{ij} - y_i^+ + y_i^- = d_i, \quad i = M, \\ & \sum_{i=1}^{M} n_i x_{ij} \le L_j, j \in \mathcal{N} \\ & y_i^+, y_i^- \in \mathbb{Z}_+, \quad i \in \mathcal{M} \\ & x_{ij} \in \mathbb{Z}_+, \quad i \in \mathcal{M}, j \in \mathcal{N}. \end{aligned} \tag{14}$$

To maximize the objective function, we can take $y_i^+ = 0$. Notice that $y_i^- \ge 0$, thus the constraints $\sum_{j=1}^{N} x_{ij} + y_i^- = d_i, i \in \mathcal{M}$ can be rewritten as $\sum_{j=1}^{N} x_{ij} \le d_i, i \in \mathcal{M}$. That is to say, problem 14 is equivalent to the deterministic model.

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

$$\{\max \sum_{j=1}^{N}\sum_{i=1}^{M}(n_i - \delta)x_{ij} : \sum_{i=1}^{M} n_i x_{ij} \le L_j, j \in \mathcal{N}; \sum_{j=1}^{N} x_{ij} \le s_i^0, i \in \mathcal{M}; x_{ij} \in Z^+\} \tag{15}$$

then solve the resulting problem (15) to obtain the optimal solution, $\mathbf{x}^1$, which represents a feasible seat planning. Aggregate $\mathbf{x}^1$ to the number of each group type, $s_i^1 = \sum_j x_{ij}^1, i \in \mathbf{M}$, which represents the supply for each group type.

To fully utilize the seats, we should set the supply $\mathbf{s}^1$ as the lower bound, then re-solve a seat planning problem. We substitute the constraint $\sum_{j=1}^{N} x_{ij} \le s_i^0, i \in \mathcal{M}$ in problem (15) with the new constraint $\sum_{j=1}^{N} x_{ij} \ge s_i^1, i \in \mathcal{M}$.

$$\{\max \sum_{j=1}^{N}\sum_{i=1}^{M}(n_i-\delta)x_{ij} : \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N}; \sum_{j=1}^{N} x_{ij} \geq s_i^1, i \in \mathcal{M}; x_{ij} \in Z^+\} \tag{16}$$

Notice that the number of unoccupied seats in the seat planning obtained from problem (16) is at most δ for each row, given any feasible supply, $\mathbf{s}^1$. To maximize the utilization of seats, we should assign full or largest patterns to each row. This procedure can be described in **Step 4** of the following algorithm.

Algorithm 2 Feasible seat planning algorithm

Step 1. Obtain the solution, $\mathbf{x}^*$, from stochastic linear programming by benders decomposition. Aggregate $\mathbf{x}^*$ to the number of each group type, $s_i^0 = \sum_j x_{ij}^*, i \in \mathbf{M}$.

Step 2. Solve problem (15) to obtain the optimal solution, $\mathbf{x}^1$. Aggregate $\mathbf{x}^1$ to the number of each group type, $s_i^1 = \sum_j x_{ij}^1, i \in \mathbf{M}$.

Step 3. Obtain the optimal solution, $\mathbf{x}^2$, from problem (16) with supply $\mathbf{s}^1$. Aggregate $\mathbf{x}^2$ to the number of each group type, $s_i^2 = \sum_j x_{ij}^2, i \in \mathbf{M}$.

Step 4. Check if row j is full for all j. When row j^0 is not full, i.e., $\sum_i n_i x_{ij} < L_{j^0}$, let $\beta = L_{j^0} - \sum_i n_i x_{ij}$. Find the smallest group size in row j^0 and mark it as i^0. If the smallest group is exactly the largest, then the row corresponds to the largest pattern and check next row. Otherwise, reduce the number of group type i^0 by one and increase the number of group type $\min\{(i^0+\beta), M\}$ by one. Continue this procedure until this row is full.

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

Remark 2. *For the integral seat planning, we provide a full or largest pattern for each row. The sequence of groups within each pattern can be arranged arbitrarily, allowing for a flexible seat planning that can accommodate realistic operational constraints. Therefore, any fixed sequence of groups within each pattern can be used to construct a seat planning that meets practical needs.*

5 Dynamic Seat Assignment Based on Stochastic Planning Policy

In this section, we discuss dynamic seat assignment based on stochastic planning policy(SPP). Recall that we need to make decisions not only on whether to accept or reject an arrival but also on how to assign seats to each group if we decide to accept it. First, we will present the dynamic seat assignment based on stochastic planning policy, which incorporates group-type control to optimize resource utilization.

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

5.1 Group-type Control

Seat planning represents the supply for each group type. We can use supply control to determine whether to accept a group. Specifically, if there is a supply available for an arriving group, we will accept the group. However, if there is no corresponding supply for the arriving group, we need to decide whether to use a larger group supply to meet the group's needs. When a group is accepted to occupy larger-size seats, the remaining empty seat(s) can be reserved for future demand.

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

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

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

Let $d(i, j)$ be the difference of expected number of accepted people between acceptance and rejection on group i occupying $(j + \delta)$-size seats. If $j \geq i + \delta$, $d(i, j)$ equals $i + (j - i - \delta)P(D_{j-i-\delta} \geq x_{j-i-\delta} + 1; T_r) - jP(D_j \geq x_j; T_r)$, otherwise, $d(i, j)$ equals $i - jP(D_j \geq x_j)$. One intuitive decision is to choose the largest difference. For all $j > i$, find the largest $d(i, j)$, denoted as $d(i, j^*)$. If $d(i, j^*) > 0$, we will place the group of i in $(j^* + \delta)$-size seats. Otherwise, reject the group.

This control is based on the current seat planning. We still need further comparison of values of accepting or rejecting an arrival to decide whether to accept it. That is, the group-type control is a necessary condition to accept a group.

5.2 Value of Stochastic Programming

After we decide to assign a small group to a larger one, we need to compare the objective values of accepting or rejecting an arrival to make a decision. To determine this objective value, we need to consider the potential outcomes that could result from accepting the current arrival, i.e., the Value of Acceptance(VoA), as well as the potential outcomes that could result from rejecting it, i.e., the Value of Rejection(VoR).

The VoA considers the scenarios that could arise if we accept the current arrival, while the VoR considers the same scenarios if we reject it. By comparing the VoA and the VoR, we can make an informed decision about whether to accept or reject the arrival based on which option has the higher objective value. This approach takes into account the uncertain nature of the decision-making environment and allows for a more optimal decision to be made.

If the VoA is larger than the VoR, it indicates that accepting the arrival would result in a higher objective value. In such cases, we refer to the corresponding planning group row in the group-type control, where we determine which group to break in order to accommodate the incoming group. If the VoA is less than the VoR, we will reject the incoming group.

In essence, this decision-making approach weighs the potential benefits and costs associated with accepting or rejecting an arrival, allowing us to select the option that maximizes the objective value and aligns with our overall goals.

5.2.1 Algorithm Based on Stochastic Planning Policy

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

The algorithm is shown below:

Algorithm 3 Stochastic planning policy algorithm

Step 1. Obtain the set of patterns, $\mathbf{P} = \{P_1, \ldots, P_N\}$, from the feasible seat planning algorithm. The corresponding aggregate supply is $\mathbf{X} = [x_1, \ldots, x_M]$.

Step 2. For the arrival group type i at period T', If $\exists k \in \mathcal{N}$ such that $i \in P_k$, accept the group, update $P_k = P_k/(i)$ and $x_i = x_i - 1$. Go to step 4. Otherwise, go to step 3.

Step 3. Calculate $d(i, j^*)$. If $d(i, j^*) > 0$, find the first $k \in \mathcal{N}$ such that $j^* \in P_k$. If value of acceptance is larger than value of rejection, accept group type i and update $P_k = P_k/(j^*)$, $x_{j^*} = x_{j^*} - 1$. Then update $x_{j^*-i-\delta} = x_{j^*-i-\delta} + 1$ and $P_k = P_k \cup (j^* - i - \delta)$ when $j^* - i - \delta > 0$. If $d(i, j^*) \leq 0$, reject group type i.

Step 4. If $T' \leq T$, move to next period, set $T' = T' + 1$, go to step 2. Otherwise, terminate this algorithm.

5.2.2 Break Tie for Stochastic Planning Policy

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

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

6 Results

We carried out several experiments, including comparing the running time of decomposition and integer programming, comparing the number of people served using the seat planning and integer programming methods, analyzing different policies, evaluating the impact of implementing social distancing.

6.1 Running time of Benders Decomposition and IP

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

Table 1: Running time of Decomposion and IP

# of scenarios	demands	# of rows	# of groups	# of seats	running time of IP(s)	Benders (s)
1000	(150, 350)	30	8	(21, 50)	5.1	0.13
5000		30	8		28.73	0.47
10000		30	8		66.81	0.91
50000		30	8		925.17	4.3
1000	(1000, 2000)	200	8	(21, 50)	5.88	0.29
5000		200	8		30.0	0.62
10000		200	8		64.41	1.09
50000		200	8		365.57	4.56
1000	(150, 250)	30	16	(41, 60)	17.15	0.18
5000		30	16		105.2	0.67
10000		30	16		260.88	1.28
50000		30	16		3873.16	6.18

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

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

6.2 Seat Planning Composed of Full or Largest Patterns versus IP Solution

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

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

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

Use $D(T, M)$ to denote the number of scenarios, which equals the number of different solutions to $x_1 + \ldots + x_M = T, \mathbf{x} \geq 0$. Then, we know the recurrence relation $D(T, M) = \sum_{i=0}^{T} D(i, M-1)$ and boundary condition, $D(i, 1) = 1$. So we have $D(T, 2) = T + 1$, $D(T, 3) = \frac{(T+2)(T+1)}{2}, D(T, M) = O(T^{M-1})$. The number of scenarios is too large to enumerate all possible cases. Thus, we choose to sample some sequences from the multinomial distribution.

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

Table 2: Feasible seat planning versus IP solution

# samples	T	probabilities	# rows	people served by decomposition	people served by IP
1000	45	[0.4,0.4,0.1,0.1]	8	85.30	85.3
1000	50	[0.4,0.4,0.1,0.1]	8	97.32	97.32
1000	55	[0.4,0.4,0.1,0.1]	8	102.40	102.40
1000	60	[0.4,0.4,0.1,0.1]	8	106.70	NA
1000	65	[0.4,0.4,0.1,0.1]	8	108.84	108.84
1000	35	[0.25,0.25,0.25,0.25]	8	87.16	87.08
1000	40	[0.25,0.25,0.25,0.25]	8	101.32	101.24
1000	45	[0.25,0.25,0.25,0.25]	8	110.62	110.52
1000	50	[0.25,0.25,0.25,0.25]	8	115.46	NA
1000	55	[0.25,0.25,0.25,0.25]	8	117.06	117.26
5000	300	[0.25,0.25,0.25,0.25]	30	749.76	749.76
5000	350	[0.25,0.25,0.25,0.25]	30	866.02	866.42
5000	400	[0.25,0.25,0.25,0.25]	30	889.02	889.44
5000	450	[0.25,0.25,0.25,0.25]	30	916.16	916.66

Each entry of people served is the average of 50 instances. IP will spend more than 2 hours in some instances, as 'NA' showed in the table. The number of seats is 20 when the number of rows is 8, the number of seats is 40 when the number of rows is 30.

6.3 Performances of Different Policies

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

6.3.1 Bid-price Control

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

The dual problem of LP relaxation of problem (1) is:

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

In (17), β_j can be interpreted as the bid-price for a seat in row j. A request is only accepted if the revenue it generates is 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 5. *The optimal solution to probelm (17) is given by $z_1, \ldots, z_h = 0$, $z_i = \frac{\delta(n_i - n_h)}{n_h}$ for $i = h+1, \ldots, M$ and $\beta_j = \frac{n_h - \delta}{n_h}$ for all j.*

The bid-price decision can be expressed as $i - \beta_j n_i = i - \frac{n_h - \delta}{n_h} n_i = \frac{\delta(i-h)}{n_h}$. When $i < h$, $i - \beta_j n_i < 0$. When $i \geq h$, $i - \beta_j n_i \geq 0$. This means that group type i with i greater than or equal to h 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 4 Bid-price control algorithm

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

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

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

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

6.3.2 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 5 Booking limit control algorithm

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

Step 2. Solve problem (1) with $d_i^t = (T-t)\cdot p_i$ and $\mathbf{L}^t$, obtain the optimal solution, x_{ij}^* and the aggregate optimal solution, $\mathbf{X}$.

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

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

6.3.3 Dynamic Programming Base-heuristic

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

Relax all rows to one row with the same capacity by $L = \sum_{j=1}^{N} L_j$. Let u denote the decision, where $u(t) = 1$ if we accept a request in period t, $u(t) = 0$ otherwise. Similar to the DP in section 3.2, the DP with one row can be expressed as:

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

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

6.3.4 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.

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

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

Table 3: Performances of Different Policies

T	probabilities	SSP(%)	DP1(%)	Bid-price(%)	Booking	FCFS(%)
60	[0.25, 0.25, 0.25, 0.25]	99.12	98.42	98.38	96.74	98.17
70	[0.25, 0.25, 0.25, 0.25]	98.34	96.87	96.24	97.18	94.75
80	[0.25, 0.25, 0.25, 0.25]	98.61	95.69	96.02	98.00	93.18
90	[0.25, 0.25, 0.25, 0.25]	99.10	96.05	96.41	98.31	92.48
100	[0.25, 0.25, 0.25, 0.25]	99.58	95.09	96.88	98.70	92.54
60	[0.25, 0.35, 0.05, 0.35]	98.94	98.26	98.25	96.74	98.62
70	[0.25, 0.35, 0.05, 0.35]	98.05	96.62	96.06	96.90	93.96
80	[0.25, 0.35, 0.05, 0.35]	98.37	96.01	95.89	97.75	92.88
90	[0.25, 0.35, 0.05, 0.35]	99.01	96.77	96.62	98.42	92.46
100	[0.25, 0.35, 0.05, 0.35]	99.23	97.04	97.14	98.67	92.00
60	[0.15, 0.25, 0.55, 0.05]	99.14	98.72	98.74	96.61	98.07
70	[0.15, 0.25, 0.55, 0.05]	99.30	96.38	96.90	97.88	96.25
80	[0.15, 0.25, 0.55, 0.05]	99.59	97.75	97.87	98.55	95.81
90	[0.15, 0.25, 0.55, 0.05]	99.53	98.45	98.69	98.81	95.50
100	[0.15, 0.25, 0.55, 0.05]	99.47	98.62	98.94	98.90	95.25

6.4 Impact of Implementing Social Distancing in SPP

In this section, our focus is to analyze the influence of social distancing on the number of accepted individuals. Intuitively, when demand is small, we will accept all arrivals, thus there is no difference whether we implement the social distancing. What is interesting for us is when the difference occurs. Our primary objective is to determine the first time period at which, on average, the number of people accepted without social distancing is not less than the number accepted with social distancing plus one. This critical point, referred to as the gap point, is of interest to us. Additionally, we will examine the corresponding occupancy rate at this gap point. It should be noted that the difference at a specific time period may vary depending on the total number of periods considered. Therefore, when evaluating the difference at a particular time period, we assume that there are a total of such periods under consideration.

It is evident that as the demand increases, the effect of social distancing becomes more pronounced. We aim to determine the specific time period where the absence of social distancing results in a higher number of accepted individuals compared to when social distancing measures are in place. Additionally, we will calculate the corresponding occupancy rate during this period.

By analyzing and comparing the data, we can gain insights into the relationship between demand, social distancing, the number of accepted individuals, and occupancy rates. This information is valuable for understanding the impact of social distancing policies on overall capacity utilization and making informed decisions regarding resource allocation and operational strategies.

6.4.1 Estimation of Gap Point

Based on our findings, we observed that the seat allocation derived from the optimal solution consistently satisfies the formation of either the largest pattern or the full pattern, regardless of different probability combinations. However, certain counterexamples arise when the requirements associated with specific probability combinations are unable to form a full pattern, resulting in gaps in the seating arrangement. The occurrence of these counterexamples is closely tied to the seat layout itself. The ratio of the number of largest patterns to the number of full patterns in the final seat allocation is influenced by the expected number of people in each period.

We can leverage the expected number of people in each period to estimate the gap point when utilizing the SPP. This approach allows us to approximate the period at which the number of people accepted without social distancing surpasses the number accepted with social distancing, based on the performance characteristics observed in the SPP.

Let γ represent the average number of people who arrive in each period, and let L represent the total number of seats available. Assuming that we accept all incoming groups within T' periods, filling all the available seats, the total number of people, taking social distancing into account, would be equal to the number of all seats. This can be expressed as $\gamma T' + \delta T' = L$, where $\gamma T'$ is the expected number of people. Thus, the expected period for reaching this point is given by $T' = \frac{L}{\gamma+\delta}$. The corresponding occupancy rate at this period can be calculated as $\frac{\gamma T'}{(\gamma+\delta)T'-N\delta} = \frac{\gamma}{\gamma+\delta}\frac{L}{L-N\delta}$. However, it is important to note that the actual first period will be smaller than the ideal one because it is impossible to accept groups to fill all seats exactly. To estimate the gap point when applying SPP, we can use $y_1 = c_1\frac{L}{\gamma+\delta}$, where c_1 is a discount rate compared to the ideal situation. Similarly, we can estimate the corresponding occupancy rate as $y_2 = c_2\frac{\gamma}{\gamma+\delta}\frac{L}{L-N\delta}$, where c_2 is a discount rate for the occupancy rate compared to the ideal scenario.

We consider the scenario where the number of group types is limited to 4. In this case, 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 people, respectively. We assume that p_4 always has a positive value. Additionally, the social distancing requirement is set to one seat.

To analyze the relationship between the increment of γ and the gap point, we define each combination (p_1, p_2, p_3, p_4) satisfying $p_1 + p_2 + p_3 + p_4 = 1$ as a probability combination. We conducted an analysis using a sample of 200 probability combinations. 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 $y_1 = \frac{c_1}{\gamma+1}$(blue line in the figure) and $y_2 = c_2\frac{L}{L-N}\frac{\gamma}{\gamma+1}$(orange line in the figure), which are fitted to the data. These equations capture the relationship between the gap point and the increment of γ, allowing us to approximate the values. We utilized an Ordinary Least Squares (OLS) model to fit the data and obtain the parameter values. For the first function, we found that $c_1 = 200.0208$, with a standard error of 0.203. For the second function, we obtained $c_2 = 90.9284$, with a standard error of 0.099. The R-square values for both models are 1.000, indicating a perfect fit between the data and the models.

By examining the relationship between the gap point and the increment of γ, we can find that γ can be used to estimate gap point.

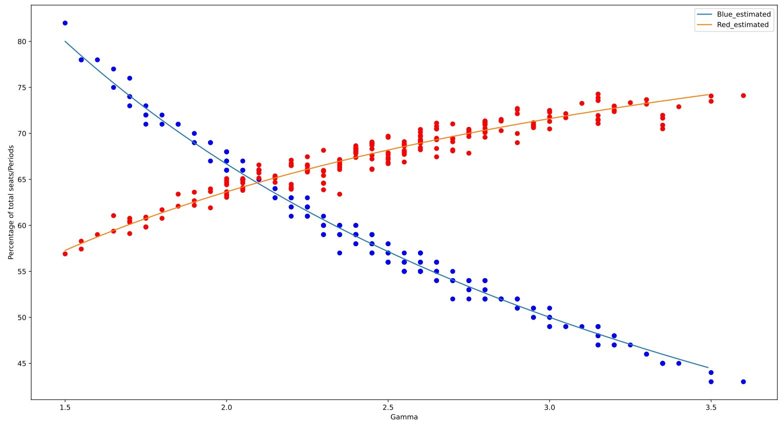


Figure 1: Gap points and their estimation under 200 probabilities

6.4.2 Impact of Social Distancing as Demands Increase

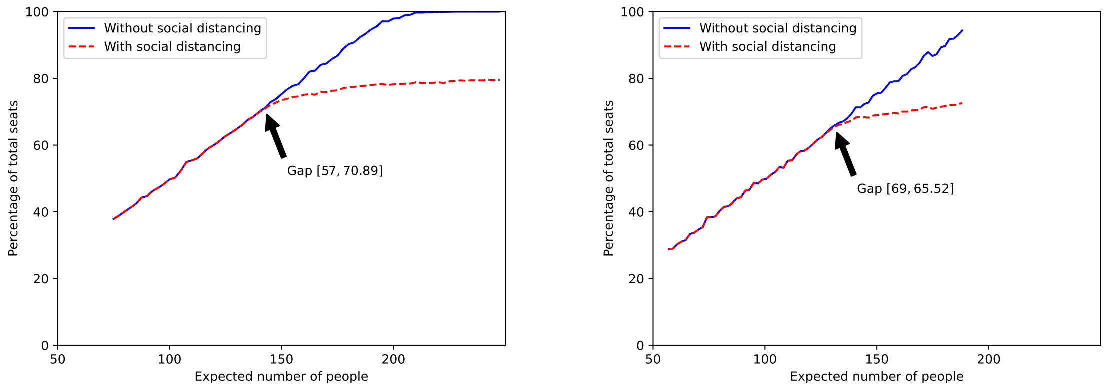
Now, we consider impact of social distance as demands increase by changing T. Specifically, we consider two situations: $\gamma = 2.5$ and $\gamma = 1.9$. We set the parameters as follows: T varies from 30 to 120, the step size is 1. The seat layout consists 10 rows and the number of seats per row is 21. The social distancing is 1 seat.

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 SPP 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 a different approach. We simply accept all incoming groups as long as the capacity allows, without considering the constraints of groups needing to sit together. This means that we prioritize filling the available seats without enforcing any specific seating arrangements or social distancing requirements. Since the various probabilities with the same γ exhibit similar patterns as shown in the figure, we present only one case of probabilities to illustrate the detailed figure.

The analysis comprises three stages. In the first stage, where the capacity is sufficient, social distancing measures have no impact on the outcome. In the second stage, the gap between the outcomes with and without social distancing measures widens as T increases. Finally, in the third stage, as T continues to increase, the gap between the outcomes with and without social distancing measures converges when both situations accept the maximum number of people.

The table below presents the gap points and the percentage gaps for different demand levels (130, 150, 170, 190, 210).

According to Lemma 1, 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. The maximum number of people that can be accepted is $200 * \frac{16}{20} = 160$, which is the upper bound on the number of people that can be accepted.



(a) When $\gamma = 2.5$ (b) When $\gamma = 1.9$

Figure 2: The occupancy rate over the number of arriving people

Table 4: Gap points of different gammas and percentage differences under different demands

γ	gap point	difference under different demands				
		130	150	170	190	210
1.9	[69, 65.52]	0.25	5.82	12.82	20.38	24.56
2.1	[64, 67.74]	0.05	4.11	11.51	18.77	21.87
2.3	[61, 69.79]	0	2.29	10.21	17.36	21.16
2.5	[57, 70.89]	0	1.45	9.30	15.78	19.80
2.7	[53, 71.28]	0	1.38	7.39	14.91	19.14

7 Conclusion

Since the outbreak of the pandemic, social distancing has been widely recognized as a crucial measure for containing the spread of the virus. It has been implemented in seating areas to ensure safety. While static seating arrangements can be addressed through integer programming by defining specific social distancing constraints, dealing with the dynamic situations is challenging.

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 a stochastic planning 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 included comparing the running time of the benders decomposition method and integer programming, analyzing different policies for dynamic seat assignment, and evaluating the impact of implementing social distancing. The results of our experiments demonstrated that the benders decomposition method efficiently solves our model. In terms of dynamic seat assignment policies, we considered 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 highlights the importance of considering the operational significance behind social distancing and provides a new perspective for the government to adopt mechanisms for setting seat assignments to protect people during the pandemic. Moreover, our analysis provides managerial guidance on how to set the occupancy rate and largest size of one group under the background of pandemic.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Proof

(Proof of Lemma 1). *We can employ a greedy approach to generate a pattern by following these steps. First, we select the maximum group size, denoted as n_M, as many times as possible, filling up the available space. The remaining seats are then allocated 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 loss of the pattern is $q\delta - \delta + \nVdash(r)$. We can prove that it is the smallest loss by contradiction. Suppose the loss is not the smallest, there exists one pattern with a loss of $p < q\delta - \delta + \nVdash(r)$. Then the largest number of seats occupied in this pattern is $\lfloor \frac{p}{\delta} \rfloor \cdot n_M$, which is always less than L due to the inequality $\frac{\nVdash}{\delta} \cdot n_M < r + n_M$. Consequently, the loss of p cannot exist as it would contradict the maximum number of seats taken.* $\square$

(Proof of Lemma 3). *Notice that $\mathbf{f}^\intercal = [-\mathbf{1},\ \mathbf{0}], V = [W,\ I]$, W is a totally unimodular matrix. Then, we have $\boldsymbol{\alpha}^\intercal W \geq -\mathbf{1}, \boldsymbol{\alpha}^\intercal I \geq \mathbf{0}$. Thus, the feasible region is bounded. Furthermore, let $\alpha_0 = 0$, then we have $0 \leq \alpha_i \leq \alpha_{i-1} + 1$, $i \in \mathcal{M}$, so the extreme points are all integral.* $\square$

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

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

When $y^-_{i\omega} > 0$, we have $\alpha_i = 0$; when $y^+_{i\omega} > 0$, we have $\alpha_i = \alpha_{i-1} + 1$. Let $\Delta d = d_\omega - d_0$, then the elements of 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 $0 \leq \alpha_i \leq \alpha_{i-1} + 1$. $\square$

(Proof of lemma 4). *Suppose we have one extreme point $\boldsymbol{\alpha}^0_\omega$ for each scenario. Then we have the following problem.*

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

Problem (18) reaches its maximum when $(\boldsymbol{\alpha}^0_\omega)^\intercal \mathbf{d}_\omega = (\boldsymbol{\alpha}^0_\omega)^\intercal \mathbf{x}\mathbf{1} + z_\omega, \forall \omega$. Substitute z_ω with these equa-

tions, we have

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

Notice that $\mathbf{x}$ *is bounded by* $\mathbf{L}$*, then the problem (18) is bounded. Adding more optimality cuts will not make the optimal value larger. Thus, RBMP is bounded.* □

(Proof of Lemma 2)**.** *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* $b = \min\{j : \sum_{k=1}^{j} w_k \geq L\}$*, where* $L = \sum_{j=1}^{N} L_j$ *is the total size of all knapsacks. Then the Dantzig upper bound [8] becomes* $u_{\text{MKP}} = \sum_{j=1}^{b-1} p_j + \left(L - \sum_{j=1}^{b-1} w_j\right) \frac{p_b}{w_b}$*. The corresponding optimal solution is to accept the whole items from* 1 *to* $b-1$ *and fractional* $(L - \sum_{j=1}^{b-1} w_j)$ *item* b*. Suppose the item* b *belong to type* h*, then for* $i < h$*,* $x_{ij}^* = 0$*; for* $i > h$*,* $x_{ij}^* = d_i$*; for* $i = h$*,* $\sum_j x_{ij}^* = (L - \sum_{i=h+1}^{M} d_i n_i)/n_h$*.* □

(Proof of Lemma 5)**.** *According to the Lemma 2, the aggregate optimal solution to relaxation of problem (1) takes the form* $xe_h + \sum_{i=h+1}^{M} d_i e_i$*, then according to the complementary slackness property, we know that* $z_1, \ldots, z_h = 0$*. This implies that* $\beta_j \geq \frac{n_i - \delta}{n_i}$ *for* $i = 1, \ldots, h$*. Since* $\frac{n_i - \delta}{n_i}$ *increases with* i*, we have* $\beta_j \geq \frac{n_h - \delta}{n_h}$*. Consequently, we obtain* $z_i \geq n_i - \delta - n_i \frac{n_h - \delta}{n_h} = \frac{\delta(n_i - n_h)}{n_h}$ *for* $i = h+1, \ldots, M$*.*

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