Dynamic Seat Assignment With Social Distancing

December 4, 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 found that our approach performs better than bid-price policy and booking-limit policy, even well compared with the offline optimal solution. 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.

This paper focuses on addressing the dynamic seating assignment problem with a given set of seats in the context of social distancing. 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 must 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 seat planning using scenario-based stochastic programming. We then make a decision for each incoming group based on our seat assignment 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 seat planning. Section 5 demonstrates the dynamic seat assignment during booking period and after booking period. Section 6 gives the results and the insights. 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 L_j^0 seats, where $j \in \mathcal{N} \coloneqq \{1, 2, \ldots, N\}$. The seating arrangement is used to accommodate various groups, where each group consists of no more than M individuals. There are M distinct group types, denoted by group type i, where each group type consists of i people. 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 represents the number of group type i.

In order to comply with the social distancing requirements, individuals from the same group must 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 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 sizes, resulting in the new size of group type i being denoted as $n_i = i + \delta$, where $i \in \mathcal{M}$. Accordingly, the length of each row is also adjusted to accommodate the adjusted group sizes. Consequently, $L_j = L_j^0 + \delta$ represents the length 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.

Let x_{ij} represent the number of group type i planned in row j. 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 regarded as a special case of the multiple knapsack problem. In this context, we define X as the aggregate solution, where $X = (\sum_{j=1}^{N} x_{1j}, \ldots, \sum_{j=1}^{N} x_{Mj})$. Each element of X, $\sum_{j=1}^{N} x_{ij}$, represents the available supply of seats for group type i. In other words, X captures the total number of seats that can be allocated to each group type 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 supply corresponding to the optimal solution of the LP relaxation of

Problem (1), as demonstrated in Proposition 2.

Although the problem size is small and the optimal solution can be easily obtained using a solver, it is still important to analyze the problem further to gain additional insights and understanding. We introduce the term pattern to refer to the seat planning arrangement for a single row. A specific pattern can be represented by a vector $\boldsymbol{t} = (t_1, \ldots, t_M)$, where t_i represents the number of group type i in the row for $i = 1, \ldots, M$. This vector $\boldsymbol{t}$ must satisfy the condition $\sum_{i=1}^{M} t_i n_i \leq L$ and belong to the set of non-negative integer values, denoted as $\boldsymbol{t} \in \mathbb{Z}_+^M$.

In order to quantify the impact of each pattern, we introduce the concept of loss, which is the number of unoccupied seats. Mathematically, the loss is defined as $L - \delta - \sum_{i=1}^{M} i t_i$, where L denotes the length of the row.

The loss provides a measure of the number of seats which cannot be taken 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. *Consider the length of the row is L, the social distancing is $\delta = 1$ and a pattern $(t_1, \ldots, t_M)$. We call the pattern $\boldsymbol{t}$ a full pattern if it satisfies the condition $\sum_{i=1}^{M} t_i n_i = L$. We call the pattern $\boldsymbol{t}$ a largest pattern if it belongs to* $\arg\min_{\boldsymbol{t}}\{L - \delta - \sum_{i=1}^{M} i t_i\}$.

In many cases, the optimal solution for the seat planning problem tends to involve rows with either full patterns or the largest patterns. Distinguishing these patterns from other configurations can provide valuable insights into effective seat planning strategies that prioritize accommodating as many people as possible while adhering to social distancing guidelines.

When there is high demand for seats, it is advantageous to prioritize the largest patterns. These patterns allow for the accommodation of the largest number of individuals due to social distancing requirements. On the other hand, in scenarios with moderate demand, adopting the full pattern becomes more feasible. The full pattern maximizes seating capacity by utilizing all available seats, except those needed for social distancing measures.

By considering both the largest and full patterns, we can optimize seat planning configurations to efficiently accommodate a significant number of individuals while maintaining adherence to social distancing guidelines.

Proposition 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 $r \leq \delta$.*

Example 1. *Consider the given values: $\delta = 1$, $L = 21$, and $M = 4$. In this case, we have $n_i = i + 1$ for $i \in 1, 2, 3, 4$. The loss of the largest pattern can be calculated as $\lfloor \frac{21}{5} \rfloor - 1 + g(1) = 5$. The largest patterns are the following: $(1, 0, 1, 3)$, $(0, 1, 2, 2)$, $(0, 0, 0, 4)$, $(0, 0, 4, 1)$, and $(0, 2, 0, 3)$.*

Through this example, we observe that the largest pattern does not exclusively consist of large groups but can also include smaller groups. This highlights the importance of considering the various

group sizes when generating the largest pattern. Another observation relates to the relationship between the largest patterns and full patterns. It is apparent that a full pattern may not necessarily be the largest pattern. For instance, consider the pattern $(1, 1, 4, 0)$, which is a full pattern as it utilizes all available seats. However, its loss value is 6, indicating that it is not the largest pattern. Conversely, a largest pattern may also not necessarily be a full pattern. Take the pattern $(0, 0, 0, 4)$ as an example. It is a largest pattern as it can accommodate the maximum number of individuals. However, it does not satisfy the requirement of fully utilizing all available seats since $4 \times 5 \neq 21$.

Although the optimal solution to the seat planning problem is complex, the LP relaxation of problem (1) has a nice property.

Proposition 2. *In the LP relaxation of problem (1), there exists an index h such that the optimal solutions satisfy the following conditions:*

- *For $i = 1, \ldots, h-1$, $x^*_{ij} = 0$ for all rows, indicating that no group type i are assigned to any rows before index h.*
- *For $i = h+1, \ldots, M$, the optimal solution assigns $\sum_j x^*_{ij} = d_i$ group type i to meet the demand for group type i.*
- *For $i = h$, the optimal solution assigns $\sum_j x^*_{ij} = \frac{L - \sum_{i=h+1}^{M} d_i n_i}{n_h}$ group type h 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 $h+1$ to M, divided by the size of group type h, denoted as n_h.*

Hence, the corresponding supply values can be summarized as follows: $X_h = \frac{L - \sum_{i=h+1}^{M} d_i n_i}{n_h}$, $X_i = d_i$ for $i = h+1, \ldots, M$, and $X_i = 0$ for $i = 1, \ldots, h-1$. These supply values represent the allocation of seats to each group type.

3.2 Dynamic Seat Assignment with Social Distancing

In a more realistic scenario, 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; partial acceptance is not permitted. 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.

The state of remaining capacity in each row is represented by a vector $\mathbf{L} = (l_1, l_2, \ldots, l_N)$, where l_j denotes the number of remaining seats in row j. Upon the arrival of a group type i in period t, the seller needs to make a decision denoted by $u_{i,j}$, 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 at that period. The feasible decision set is defined as

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

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 $U(\mathbf{L})$ consists of all possible combinations of acceptance and rejection decisions for each group type in each row, subject to the constraints that at most one group of each type can be accepted in any row, and the number of seats occupied by each accepted group must not exceed the remaining capacity of the row.

Let $V_t(\mathbf{L})$ denote the maximum expected revenue earned by the best decisions regarding group seat assignments in period t, given remaining capacity $\mathbf{L}$. Then, the dynamic programming formula for this problem can be expressed as:

$$V_t(\mathbf{L}) = \max_{u_{i,j} \in U(\mathbf{L})} \left\{ \sum_{i=1}^{M} p_i (\sum_{j=1}^{N} i u_{i,j} + V_{t+1}(\mathbf{L} - \sum_{j=1}^{N} n_i u_{i,j} \mathbf{e}_j)) + p_0 V_{t+1}(\mathbf{L}) \right\} \tag{2}$$

with the boundary conditions $V_{T+1}(\mathbf{L}) = 0, \forall \mathbf{L}, V_t(\mathbf{0}) = 0, \forall t$. Implying that

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 selling horizon from period 1 to T which is represented by $V_1(\mathbf{L})$.

It is difficult to solve the DP (2) due to the curse of dimensionality. To avoid this complexity, we seek to develop the heuristic method to assign arriving groups. It consists of generating the seat planning in Section 4 and stochastic assignment policy in Section 5.

4 Seat Planning Composed of Full or Largest Patterns

This section focuses on obtaining the 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. Based on that, we construct a seat planning composed of full or largest patterns to fully utilize all seats.

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 $\omega \in \Omega$. 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 $\mathbf{x} \in \mathbb{Z}+^{M\times N}$. Here, x_{ij} denotes the number of group type i assigned to row j. The total supply for group type i can be calculated as the sum of x_{ij} over all rows j, i.e., $\sum_{j=1}^{N} x_{ij}$. 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 $\mathbf{y}^+$, denoted as $y^+_{i\omega}$, represents the surplus 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.

Considering that the group can occupy the seats planned for the larger group type when the corresponding supply is not enough, 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, $i \le M-1$, $y^+_{i\omega} = \left(\sum_{j=1}^{N} x_{ij} - d_{i\omega} + y^+_{i+1,\omega}\right)^+$ 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 $y^+_{M\omega} = (\sum_{j=1}^{N} x_{Mj} - d_{M\omega})^+$, $y^-_{M\omega} = (d_{M\omega} - \sum_{j=1}^{N} x_{Mj})^+$.

Then we have the formulation of SSP:

$$\max \quad E_\omega \left[(n_M - \delta)(\sum_{j=1}^{N} x_{Mj} - y^+_{M\omega}) + \sum_{i=1}^{M-1} (n_i - \delta)(\sum_{j=1}^{N} x_{ij} + y^+_{i+1,\omega} - y^+_{i\omega}) \right] \tag{3}$$

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

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

$$\sum_{i=1}^{M} n_i x_{ij} \le L_j, j \in \mathcal{N} \tag{6}$$

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

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

The objective function consists of two parts. The first part represents the number of the largest group type that can be accommodated, given by $\sum_{j=1}^{N} x_{Mj} - y^+_{M\omega}$. Here, $\sum_{j=1}^{N} x_{Mj}$ represents the total supply for group type M and $y^+_{M\omega}$ represents the number of surplus supply for group type M in scenario ω. The second part represents the number of group type i, excluding M, that can be accommodated. It is given by $\sum_{j=1}^{N} x_{ij} + y^+_{i+1,\omega} - y^+_{i\omega}$. Similarly, $\sum_{j=1}^{N} x_{ij}$ represents the total supply for group type i, $y^+_{i\omega}$ represents the number of surplus supply for group type i in scenario ω. 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 in the 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 stochastic programming has the following properties.

For any i, ω, at most one of $y^+_{i\omega}$ and $y^-_{i\omega}$ can be positive of the optimal solution. 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 (4) and (5) but increase the objective value when p_{ω_0} is positive. Thus, in the optimal solution, at most one of $y^+_{i\omega}$ and $y^-_{i\omega}$ can be positive.

Proposition 3. *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.*

(Proof of Proposition 3)**.** *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 overall solution. Next, we explain the allocation scheme that ensures each row becomes full or largest. Let* $\beta = L_j - \sum_i n_i x_{ij}$*. If row* j *is not full or largest, then* $\beta > 0$*. To allocate the remaining unoccupied seats in row* j*, we can follow the following steps:*

- *If* $\beta \geq n_M$*, we can assign* n_M *seats to a new group type* M*. We repeat this procedure until* $\beta < n_M$*.*
- *If* $n_1 \leq \beta < n_M$*, we can assign* β *seats to a new group type* $\beta - n_1 + 1$*.*
- *If* $0 < \beta < n_1$*, we need to assign the seats to the existing seat planning. Find the smallest group type denoted as* i^0*. If* $i^0 = M$*, it means that this row represents a largest pattern, we already change this row to a largest pattern. If* $i^0 \neq M$*, we reduce the number of group type* i^0 *by one and increase the number of group type* $\min(i^0 + \beta), M$ *by one. We repeat this procedure until* $\beta = 0$ *or until this pattern is changed to a largest one.*

By repeating these steps for each row, we can ensure that each row in the seat planning becomes either full or largest. This allocation scheme maintains the optimality of the solution while maximizing the utilization of available seats. □

Suppose there exists an optimal solution where at least one corresponding pattern is not full or largest. In such a case, we can utilize the following algorithm to construct another optimal solution in which all patterns are both full and the largest.

Algorithm 1 Construct The Full or Largest Pattern Algorithm

Step 1 Given the solution $\{x_{ij}\}$. Let X represent the supply of the given solution. Check if every pattern is full or largest. Start from $j = 1$.

Step 2 Let $\beta = L_j - \sum_i n_i x_{ij}$.

Step 2.1 If $\beta = 0$, let $j = j + 1$. Continue step 2.

Step 2.2 If $\beta \geq n_M$, let $\beta = \beta - n_M$, $X_M = X_M + 1$. Continue this step until $\beta < n_M$. Go to step 2.3 or 2.4.

Step 2.3 If $n_1 \leq \beta < n_M$, let $\beta = 0$, $X_{\beta - n_1 + 1} = X_{\beta - n_1 + 1} + 1$, $j = j + 1$. Go to step 2.

Step 2.4 If $n_1 > \beta > 0$, find the smallest group type denoted as i^0 in row j.

Step 3 If $i^0 = M$, then the row represents a largest pattern and let $j = j + 1$, go to Step 2. If $i^0 \neq M$, reduce the number of group type i^0 by one and increase the number of group type $\min\{(i^0 + \beta), M\}$ by one. Go to Step 2.

Notice that this procedure does not impact the optimality of the solution. Instead, it maximizes the utilization of available seats.

Let $\mathbf{n} = (n_1, \ldots, n_M)$, $\mathbf{L} = (L_1, \ldots, L_N)$ where n_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 (6) can be expressed as $\mathbf{nx} \leq \mathbf{L}$.

The linear constraints associated with scenarios, i.e., constraints (4) and (5), 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, $\mathbf{V} = [\mathbf{W},\ \mathbf{I}]$.

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

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,

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal\mathbf{x} + z(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{n}\mathbf{x} \leq \mathbf{L} \\ & \mathbf{x} \in \mathbb{Z}_{+}^{M\times N}, \end{aligned} \tag{7}$$

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 $\omega \in \Omega$,

$$\begin{aligned} z_\omega(\mathbf{x}) := \max \quad & \mathbf{f}^\intercal\mathbf{y} \\ \text{s.t.} \quad & \mathbf{x}\mathbf{1} + \mathbf{V}\mathbf{y} = \mathbf{d}_\omega \\ & \mathbf{y} \geq 0. \end{aligned} \tag{8}$$

Problem (8) maintains the same mathematical form across different scenarios. Therefore, if we can efficiently solve problem (8), it implies that we can also solve problem (7) quickly.

4.2 Solve SSP by Benders Decomposition

In this section, we employ Benders decomposition to solve SSP, after transforming problem (8) into a master problem and a subproblem (8). Firstly, we generate a closed-form solution to problem (8), then we obtain the solution to the linear relaxation of problem (7) 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

Notice that the feasible region of the dual problem (8) remains unaffected by the variable $\mathbf{x}$. 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

$$\begin{aligned} \min \quad & \boldsymbol{\alpha}^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \\ \text{s.t.} \quad & \boldsymbol{\alpha}^\intercal\mathbf{V} \geq \mathbf{f}^\intercal \end{aligned} \tag{9}$$

Proposition 4. *Let $\mathbb{P} = \{\boldsymbol{\alpha} \in \mathbb{R}^M | \boldsymbol{\alpha}^\intercal\mathbf{V} \geq \mathbf{f}^\intercal\}$. The feasible region of problem (9), $\mathbb{P}$, is not empty and bounded. Furthermore, all the extreme points of $\mathbb{P}$ are integral.*

Therefore, the optimal value of the problem (8), $z_\omega(\mathbf{x})$, is finite and can be achieved at extreme points of the set P. Let $\mathcal{O}$ be the set of all extreme points of $\mathbb{P}$. That is, we have $z_\omega(\mathbf{x}) = \min_{\boldsymbol{\alpha}\in\mathcal{O}} \boldsymbol{\alpha}^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1})$.

Alternatively, $z_\omega(\mathbf{x})$ is the largest number z_ω such that $\boldsymbol{\alpha}^\intercal(\mathbf{d}_\omega - \mathbf{x1}) \geq z_w, \forall \boldsymbol{\alpha} \in \mathcal{O}$. We use this characterization of $z_w(\mathbf{x})$ in problem (7) and conclude that problem (7) can thus be put in the form:

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} p_\omega z_\omega \\ \text{s.t.} \quad & \mathbf{nx} \leq \mathbf{L} \\ & \boldsymbol{\alpha}^\intercal(\mathbf{d}_\omega - \mathbf{x1}) \geq z_\omega, \boldsymbol{\alpha} \in \mathcal{O}, \forall \omega \\ & \mathbf{x} \in \mathbb{Z}_+, z_\omega \text{ is free} \end{aligned} \tag{10}$$

Before applying Benders decomposition to solve Problem (10), it is important to address the efficient computation of the optimal solution to Problem (9). 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 (8), 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 $\mathbf{f}^\intercal y_\omega$. The dual optimal solution can be obtained by the following proposition.

Proposition 5. *The optimal solutions to problem (9) are given by*

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

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 $\alpha_{M\omega}$.

4.2.2 Delayed Constraint Generation

Due to the computational infeasibility of solving Problem (10) 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 (10). This results in a modified problem known as the Restricted Benders Master Problem(RBMP). To find the optimal solution of Problem (10), we employ a technique called delayed constraint generation. It involves iteratively solving the RBMP and incrementally adding more constraints until the optimal solution to Problem (10) is obtained.

We can conclude that the RBMP will have the form:

$$
\begin{aligned}
\max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} p_\omega z_\omega \\
\text{s.t.} \quad & \mathbf{nx} \leq \mathbf{L} \\
& \boldsymbol{\alpha}^\intercal (\mathbf{d}_\omega - \mathbf{x1}) \geq z_\omega, \boldsymbol{\alpha} \in \mathcal{O}^t, \forall \omega \\
& \mathbf{x} \in \mathbb{Z}_+, z_\omega \text{ is free}
\end{aligned} \tag{13}
$$

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

Proposition 6. *RBMP is always bounded with at least any one feasible constraint 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 (9) can be obtained according to Proposition 5. Let $z_\omega^{(0)} = \boldsymbol{\alpha}_\omega^1 (d_\omega - \mathbf{x}_0 \mathbf{1})$ 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 (10).

If for every scenario ω, the optimal value of the corresponding problem (9) is larger than or equal to z_ω^0, which means all contraints are satisfied, then we have an optimal solution, $(\mathbf{x}_0, \mathbf{z}_\omega^0)$, to the BMP.

However, if there exists at least one scenario ω for which the optimal value of problem (9) is less than z_ω^0, indicating that the constraints are not fully satisfied, we need to add a new constraint $(\boldsymbol{\alpha}_\omega^1)^\intercal (\mathbf{d}_\omega - \mathbf{x1}) \geq z_\omega$ to RBMP.

The steps of the algorithm are described as below,

Algorithm 2 The benders decomposition algorithm

Step 1. Solve problem (13) 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, $(\alpha_\omega^1)^\intercal (\mathbf{d}_\omega - \mathbf{x1}) \geq z_\omega$, 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 $UB - LB < \epsilon$.

From the Lemma 6, 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 (13) directly can be computationally challenging in some cases, so practically we first obtain the optimal solution to the LP relaxation of problem (7). Then, we generate an integral seat planning from this solution.

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 wish to obtain the seat planning composed of full or largest patterns.

Let the optimal solution to the relaxation of SSP be $\mathbf{x}^*$. 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 $\mathbf{s}^0$ in the deterministic model, we have the following problem,

$$\{\max \sum_{j=1}^{N} \sum_{i=1}^{M} (n_i - \delta) x_{ij} : \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N}; \sum_{j=1}^{N} x_{ij} \leq s_i^0, i \in \mathcal{M}; x_{ij} \in Z^+\} \tag{14}$$

then solve the resulting problem (14) 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 maximize the utilization of seats, we should assign full or largest patterns to each row. We will check if every row is full. When row j is not full, i.e., $\sum_i n_i x_{ij} < L_j$, let $\beta = L_j - \sum_i n_i x_{ij}$. If β is no less than n_M, then let $\beta = \beta - n_M$, $s_M = s_M + 1$ until β is less than n_M. If row j is still not full, then find the smallest group size in row j and mark it as i^0. If the smallest group is exactly the largest, then the row represents a 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.

This procedure can be described in **Step 4** of the following algorithm.

Algorithm 3 Seat Planning Construction Algorithm

Step 1. Obtain the solution, $\mathbf{x}^*$, from stochatic 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 (14) to obtain the optimal solution, $\mathbf{x}^1$.

Step 3. Construct the full or largest patterns mentioned in algorithm 1.

For Algorithm 3, **Step 2** can give a feasible seat planning, **Step 4** can give the full or largest patterns 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.

5 Dynamic Seat Assignment Based on Stochastic Assignment Policy

In this section, we discuss dynamic seat assignment based on stochastic assignment policy (SAP), which includes the group-type control and value of stochastic programming control to assign the seats to groups.

Recall that our decision-making process involves not only determining whether to accept or reject an arrival but also selecting the specific row for seat assignment if we decide to accept the group. 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 part, 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 j with 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.

Group-type policy can only tell us which group type will be provided for the smaller group based on the current planning, we still need to further compare the values of stocahstic programming when accepting or rejecting an group on the specific row.

5.2 Value of Stochastic Programming Control

After we determine the larger group to assign the arriving group, we need to compare the objective values of stochastic programming when accepting or rejecting this group to make a decision. For the objective values of stochastic programming, 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 decision about whether to accept or reject the arrival based on which option has the higher expected 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 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 Assignment Policy

We do not need to regenerate the seat planning in every period. Instead, when the available supply is sufficient for the arriving group, we can directly assign the group and keep the remaining supply. Only when comparing the VoA and VoR, we need to update the seat planning that is generated during the stochastic programming calculation. There is one exception to regenerating the seat planning, which occurs when the arriving group is the largest and the corresponding supply is only 1. In this case, after accepting the largest group, the supply becomes 0, necessitating the generation of a new seat planning.

The algorithm is shown below:

Algorithm 4 Stochastic assignment policy algorithm

Step 1. Observe the arrival group type i at period $T' = 1$.

Step 2. Obtain the set of patterns, $\mathbf{P} = \{P_1, \ldots, P_N\}$, from algorithm 3. The corresponding aggregate supply is $\mathbf{X} = (x_1, \ldots, x_M)$.

Step 3. If $\exists k \in \mathcal{N}$ such that $P_{ki} > 0$, accept the group, update $P_{ki} = P_{ki} - 1$ and $x_i = x_i - 1$. When $i = M$ and $x_M = 0$, obtain the new pattern $\mathbf{P}$ from algorithm 3. Then go to step 5. If there does not exist $k \in \mathcal{N}$ such that $P_{ki} > 0$, go to step 4.

Step 4. Calculate $d(i, j^*)$. If $d(i, j^*) > 0$, find the first $k \in \mathcal{N}$ such that $P_{kj^*} > 0$. If VoA is larger than VoR, accept group type i; otherwise, reject group type i. Obtain the new pattern $\mathbf{P}$ from algorithm 3. If $d(i, j^*) \leq 0$, reject group type i.

Step 5. If $T' \leq T$, move to next period, set $T' = T' + 1$, go to step 3. Otherwise, terminate this algorithm.

5.2.2 Break Tie for Stochastic Assignment 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 this 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 analyzing different policies, evaluating the impact of implementing social distancing.

6.1 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.

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:

$$\begin{aligned} \min \quad & \sum_{i=1}^{M} d_i z_i + \sum_{j=1}^{N} L_j \beta_j \\ \text{s.t.} \quad & z_i + \beta_j n_i \geq (n_i - \delta), \quad i \in \mathcal{M}, j \in \mathcal{N} \\ & z_i \geq 0, i \in \mathcal{M}, \beta_j \geq 0, j \in \mathcal{N}. \end{aligned} \tag{15}$$

In (15), 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 1. *The optimal solution to probelm (15) 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 5 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 $xe_h + \sum_{i=h+1}^{M} d_i e_i$.

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.

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

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

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.

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:

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

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.

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.

Table 1: Performances of Different Policies

T	probabilities	SAP (%)	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		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
60	[0.25, 0.35, 0.05, 0.35]	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
60	[0.15, 0.25, 0.55, 0.05]	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

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.

6.2 Impact of Implementing Social Distancing in SAP

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 relation 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.

6.2.1 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 SAP. 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 SAP.

To find such a first period, we aim to find the maximum period such that we could assign all the groups during these periods into the seats, 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. Let $L = \sum_j L_j$, representing the total number of seats, $\gamma = \sum_i i p_i$, representing the average number of people who arrive in each period, we can obtain $T \leq \frac{L}{\gamma+\delta}$, then the upper bound of the expected maximum period is $T' = \frac{L}{\gamma+\delta}$.

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 $\frac{\gamma T'}{(\gamma+\delta)T'-N\delta} = \frac{\gamma}{\gamma+\delta}\frac{L}{L-N\delta}$. 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 SAP, we can use $y_1 = c_1\frac{L}{\gamma+\delta}$, where c_1 is a discount rate compared to the ideal assumption. Similarly, we can estimate the corresponding occupancy rate as $y_2 = c_2\frac{\gamma}{\gamma+\delta}\frac{L}{L-N\delta}$, 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 relation 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 $y_1 = \frac{c_1 L}{\gamma+1}$(blue line in the figure) and $y_2 =$

$c_2 \frac{\gamma}{\gamma+\delta} \frac{L}{L-N\delta}$(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.

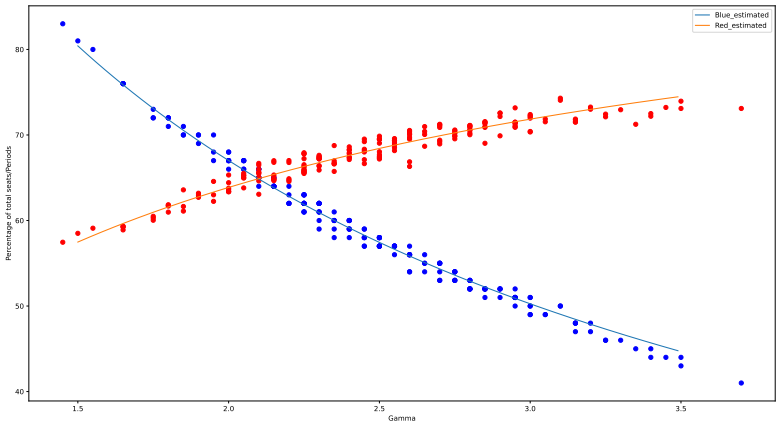


Figure 1: Gap points and their estimation under 200 probabilities

The estimation 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 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:

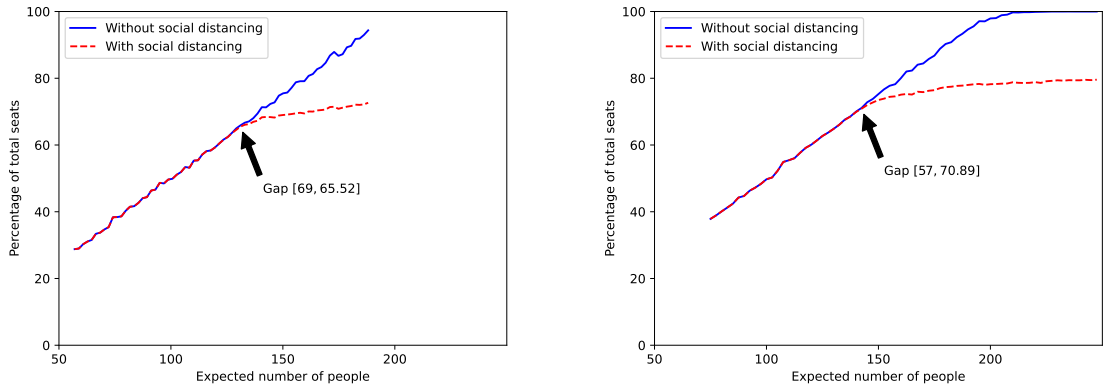
Table 2: Estimation of c_1 and c_2

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

6.2.2 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: $\gamma = 1.9$ and $\gamma = 2.5$. 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 SAP 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. We conduct an analysis using a sample of 100 probability combinations associated with the same γ. The occupancy rate at different demands is calculated as the mean of these 100 samples. The figures depicting the results are presented below.



(a) When $\gamma = 1.9$

(b) When $\gamma = 2.5$

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

The analysis consists of three stages. In the first stage, when the capacity is sufficient, the outcome remains unaffected by the implementation of social distancing measures. In the second stage, as the value of T increases, the difference between the outcomes with and without social distancing measures becomes more pronounced. Finally, in the third stage, as T continues to increase, both scenarios 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 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 below table presents the percentage differences for different demand levels (130, 150, 170, 190, 210).

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

γ	gap point	percentage differences 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.

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 Proposition 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 $q\delta - \delta + g(r)$. 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 $p < q\delta - \delta + g(r)$. 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 $\frac{g(r)}{\delta} \cdot n_M < r + n_M$. Consequently, the loss of p cannot exist as it would contradict the maximum number of seats taken.* □

(Proof of Proposition 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 Proposition 4). *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.* □

(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.

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} p_\omega z_\omega \\ s.t. \quad & \mathbf{nx} \leq \mathbf{L} \\ & (\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{d}_\omega \geq (\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{x1} + z_\omega, \forall \omega \\ & \mathbf{x} \in \mathbb{Z}_+ \end{aligned} \tag{16}$$

Problem (16) reaches its maximum when $(\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{d}_\omega = (\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{x1} + z_\omega, \forall \omega$. *Substitute* z_ω *with these equations, we have*

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal \mathbf{x} - \sum_{\omega} p_\omega (\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{x1} + \sum_{\omega} p_\omega (\boldsymbol{\alpha}_\omega^0)^\intercal \mathbf{d}_\omega \\ s.t. \quad & \mathbf{nx} \leq \mathbf{L} \\ & \mathbf{x} \in \mathbb{Z}_+ \end{aligned} \tag{17}$$

Notice that $\mathbf{x}$ *is bounded by* $\mathbf{L}$, *then the problem (16) is bounded. Adding more constraints will not make the optimal value larger. Thus, RBMP is bounded.* □

(Proof of Lemma 1)**.** *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* $i = h+1, \ldots, M$. □

8 Other Results

8.1 Running times of solving SSP and relaxation of SSP with Benders Decomposition

The running times of solving SSP directly and solving the LP relaxation of SSP with Benders decomposition are shown in Table 4.

Table 4: Running times of solving SSP and relaxation of SSP with Benders Decomposition

# 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 SSP, 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.

It is evident that the utilization of Benders decomposition can enhance computational efficiency.

8.2 Seat Planning From SSP Relaxation versus Seat Planning From SSP

In this section, we compare the accpected people of seat plannings from SSP relaxation and SSP under group-type control policy.

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,

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

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 these two seat plannings have a close performance when considering group-type control policy.

Table 5: Seat planning from SSP relaxation versus seat planning from SSP

# samples	T	probabilities	# rows	people served by SSP relaxation	people served by SSP
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.