Dynamic Seat Assignment With Social Distancing

Dis· count

June 13, 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 plans 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 Dynamic Seat Assignment (DSA) 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. Additionally, we will offer guidance on setting appropriate occupancy rates and group sizes. 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.

Some papers use seat assignment to express seat planning, to avoid confusion about seat planning and seat assignment, we clarify the difference between them. Seat planning involves determining the best layout and arrangement of seats in a venue or space based on factors such as the size of the room, the number of attendees, and the type of event. This can include deciding on the number of rows and columns, the spacing between seats, and any special requirements such as seats partition with social distancing. Seat assignment, on the other hand, involves assigning specific seats to attendees based on factors such as ticket type and availability. This is typically done closer to the event date and can involve a variety of methods such as manual seat selection by the attendee or an automated system that assigns seats based on predetermined criteria. For deterministic demand, seat planning equals seat assignment.

When purchasing tickets for movies or concerts, there are generally two approaches to seat assignment: seat assignment after all groups arrived and seat assignment for each group arrival.

The seat assignment after all groups arrived involves delaying seat assignment until the reservation deadline has passed. This means that the organizer does not need to immediately allocate seats to customers, but has to make the decision to accept or deny, so implementing social distancing restrictions will not affect the booking process. After the reservation deadline, the seller will inform customers of the seat layout information before admission. For instance, in venues such as singing concert halls, where there is high ticket demand and numerous seats available, organizers usually do not determine the seats during booking. Instead, they will inform customers of the seat information after the overall demands are determined. This approach allows for more flexibility in seat assignments and can accommodate changes in group sizes or preferences. However, in other venues where seating options are limited, it may be necessary to assign seats immediately upon accepting a group.

On the other hand, the seat assignment for each group arrival typically involves the cinema releasing the seating charts online, which show the available and unavailable seats, when there are no social distancing requirements. Customers can then choose their desired seats and reserve them by paying for their tickets. After successful payment, the seats are allocated to the customers. However, due to

social distancing requirements, this approach needs to be modified. Seat assignments are still arranged when groups book their tickets, but the seller will provide the seat information directly. For example, in movie theaters with relatively few seats, the demands for tickets are usually low enough to allow for free selection of seats directly online. Early seat planning can satisfy the requirement of social distancing and save costs without changing seat allocation. The seat allocation could remain for one day because the same film genre will likely attract similar groups with similar seating preferences.

Our study mainly focuses on the latter 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 and a maximum capacity percentage, 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 seating plan 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 in the post-pandemic era.

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 two-stage stochastic programming and Benders decomposition methods, we 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 seating plans.

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, we first consider 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 set of groups, each of which consists of no more than M people, to be assigned to a set of seats. There are M different group types, with group type i containing i people, where $i \in \mathcal{M} \coloneqq \{1, 2, \ldots, M\}$. (We use $\mathcal{M} = \{1, \ldots, M\}$ to denote the set of all positive integers that are no larger than M.)

These groups can be represented by a demand vector, denoted by $\mathbf{d} = [d_1, \ldots, d_M]$. Each element d_i, where $i \in \mathcal{M}$, indicates the number of group type i. For illustration, we consider a layout consisting of N rows, each containing S_j seats, where $j \in \mathcal{N}$.

In accordance with epidemic prevention requirements, customers from the same group are allowed to sit together, while different groups must maintain social distancing. Let s denote the social distancing, which can be one seat or more seats.

Specifically, each group must leave empty seat(s) to maintain social distancing from adjacent groups. Additionally, different rows do not affect each other, meaning that a person from one group can sit directly behind a person from another group.

To achieve the social distancing requirements in the seat planning process, we add s to the original size of each group to create the new size of the group. Let $n_i = i + s$ denote the new size of group type i for each $i \in \mathcal{M}$. Construct new seat layout by adding s to each row, i.e., let $L_j = S_j + s$ denote the length of row j for each $j \in \mathcal{N}$, where S_j represents the number of seats in row j.

Then we can illustrate the seat planning for one row below.

Figure 1: Problem Conversion

The social distancing here is one seat. On the left side of the diagram, the blue squares represent the empty seats required for social distancing, while the orange squares represent the seats occupied by groups. On the right side, we have added one dummy seat at the end of each row. The orange squares surrounded by the red line represent the seats taken by groups in this row, which includes two groups of 1, one group of 2, and one group of 3.

By incorporating the additional seat and designating certain seats for social distancing, we can integrate social distancing measures into the seat planning problem.

Now, we analyse the effect of introducing social distancing for each row. At first, we consider the types of pattern, which refers to the seat planning for each row. For each pattern k, we use α_k, β_k to indicate the number of groups and the left seats, respectively. Denote by $l(k) = \alpha_k + \beta_k - 1$ the loss

for pattern k. The loss represents the number of people lost compared to the situation without social distancing.

Let I_1 be the set of patterns with the minimal loss. Then we call the patterns from I_1 are the largest. Similarly, the patterns from I_2 are the second largest, so forth and so on. The patterns with zero left seat are called full patterns. Suppose there are n groups in a row, for ease of brevity, we use a descending form $P_k = (t_1, t_2, \ldots, t_n)$ to denote pattern k, where t_h is the new group size, $h = 1, \ldots, n$.

Example 1. *Suppose the social distancing is one seat and there are four types of groups, then the new sizes of groups are* $2, 3, 4, 5$*, respectively. The length of one row is* $L = 21$ *Then these patterns,* $(5, 5, 5, 5, 1), (5, 4, 4, 4, 4), (5, 5, 5, 3, 3)$*, belong to* I_1*.*

We can use the following greedy way to generate the largest pattern. Select the maximal group size, n_M, as many as possible and the left space is assigned to the group with the corresponding size. Let $L = n_M \cdot q + r, 0 \leq r < n_M$, where q is the number of times n_M selected. When $r > 0$, there are $d[0][u - r][q + 1]$ largest patterns with the same loss of q. When $r = 0$, there is only one possible largest pattern.

Use dynamic programming to solve. $d[k][i][j]$ indicates the number of assignment of using i capacity to allocate j units, k is the number of capacity allocated on the last unit. In our case, $u - r$ is the capacity need to be allocated, $q + 1$ is the number of units which corresponds to the groups. Notice that we only consider the number of combinations, so we fix the allocation in ascending order, which means the allocation in current unit should be no less than the last unit.

The number of largest patterns equals the number of different schemes that allocate $u - r$ on $q + 1$ units,i.e., $d[0][u - r][q + 1]$.

The recurrence relation is $d[k][i][j] = \sum_{t=k}^{i-k} d[t][i - k][j - 1]$. When $i < k$, $d[k][i][j] = 0$; when $i \geq k$, $d[k][i][1] = 1$.

Lemma 1. *When given the length of row,* L*, and the new size of the largest group allowed,* n_M*. Let* $L = n_M \cdot q + r$*, then the loss of the largest pattern is* $q - f(r)$*, where* $f(r) = 1$ *if* $r = 0$*;* $f(r) = 0$ *if* $r \neq 0$*.*

Lemma 2. *The seat assignment made up of the largest patterns is optimal.*

This lemma holds because we cannot find a better solution occupying more seats. When the demand can meet that the largest patterns can be generated in all rows, an optimal seat assignment can be obtained.

Proposition 1. *For a seat layout,* $\{S_1, S_2, \ldots, S_N\}$*, the total loss is* $\sum_j (\lfloor \frac{S_j+1}{u} \rfloor - f((S_j + 1) \mod u))$*. The maximal number of people assigned is* $\sum_j (S_j - \lfloor \frac{S_j+1}{u} \rfloor + f((S_j + 1) \mod u))$*.*

3.2 Dynamic Seat Assignment with Social Distancing

Now consider the scenario where groups arrive dynamically, the decision-maker must determine whether to accept or reject each group and assign them to empty seats while ensuring that the social

distancing constraint is met. Once the seats are confirmed and assigned to a group, they cannot be changed.

We use a vector $\mathbf{L} = (L_1^r, L_2^r, \ldots, L_N^r)$ to record the remaining capacity of rows, where L_j^r represents the number of remaining seats in row j. Let $V_t(\mathbf{L})$ denote the maximal expected value to go at period t with capacity of rows. There are T periods and the arrival probability of the group type i in each period is p_i. In every period, the group will be decided which row to assign. Let $u_{i,j}$ denote the decision, where $u_{i,j}(t) = 1$ if we accept a group type i in row j at period t, $u(t) = 0$ otherwise.

The dynamic programming formulation for this problem is

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

$U(\mathbf{L}) = \{u_{i,j} \in \{0,1\}, \forall i, j | \sum_{j=1}^{N} u_{i,j} \leq 1, \forall i, n_i u_{i,j} \mathbf{e}_j^\top \leq \mathbf{L}, \forall i, j\}$ and $\mathbf{e}_j$ is the unit row vector with j-th element being 1.

The compact form can be written as:

$$V_t(\mathbf{L}) = \mathbb{E}_{i \sim p} \left[\max_{\substack{j \in \mathcal{N}: \\ L_j \geqslant n_i}} \left\{ V_{t+1}\left(\mathbf{L} - n_i \mathbf{e}_j^\top\right) + i, V_{t+1}(\mathbf{L}) \right\} \right]$$

Initially, we have $\mathbf{L}_T = (L_1, L_2, \ldots, L_N)$. As we can observe, the dynamic programming algorithm has to make a decision on which row to assign group type i. 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 for assigning groups. To obtain the final seat planning firstly, we develop the scenario-based stochastic programming.

4 Scenario-based Stochastic Programming

Firstly, we give the formulation of the scenario-based stochastic programming. Then we develop the benders decomposition to obtain the optimal linear solution. Finally, we obtain a feasible seat planning.

4.1 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 $\omega \in \Omega, \Omega$ corresponds to a particular realization of the demand vector, $\mathbf{D}_\omega = (d_{1\omega}, d_{2\omega}, \ldots, d_{M,\omega})$. 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 assignment based on the scenarios then assign the groups to seats according to the realized true demand.

The seat assignment 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 $S = |\Omega|$ possible scenarios. There is a scenario-dependent decision variable, $\mathbf{y}$, to be chosen. It includes two vectors of decisions, $\mathbf{y}^+ \in \mathbb{Z}_+^{M\times S}$ and $\mathbf{y}^- \in \mathbb{Z}_+^{M\times S}$. Each component of $\mathbf{y}^+$, $y_{i\omega}^+$, represents the number of surplus seats for group type i. Similarly, $y_{i\omega}^-$ represents the number of inadequate seats for group type i. Considering that the group can take the seats assigned to the larger group type, we assume that the surplus 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 $y_{M\omega}^+ = (\sum_{j=1}^N x_{ij} - d_{i\omega})^+$, $y_{M\omega}^- = (d_{i\omega} - \sum_{j=1}^N x_{ij})^+$.

Then we have the deterministic equivalent form of the scenario-based stochastic programming:

$$
\begin{aligned}
\max \quad & E_\omega \left[\sum_{i=1}^{M-1} (n_i - s)(\sum_{j=1}^{N} x_{ij} + y_{i+1,\omega}^+ - y_{i\omega}^+) + (n_M - s)(\sum_{j=1}^{N} x_{Mj} - y_{M\omega}^+) \right] \\
\text{s.t.} \quad & \sum_{j=1}^{N} x_{ij} - y_{i\omega}^+ + y_{i+1,\omega}^+ + y_{i\omega}^- = d_{i\omega}, \quad i = 1, \ldots, M-1, \omega \in \Omega \\
& \sum_{j=1}^{N} x_{ij} - y_{i\omega}^+ + y_{i\omega}^- = d_{i\omega}, \quad i = M, \omega \in \Omega \\
& \sum_{i=1}^{M} n_i x_{ij} \le L_j, j \in \mathcal{N} \\
& y_{i\omega}^+, y_{i\omega}^- \in \mathbb{Z}_+, \quad i \in \mathcal{M}, \omega \in \Omega \\
& x_{ij} \in \mathbb{Z}_+, \quad i \in \mathcal{M}, j \in \mathcal{N}.
\end{aligned} \tag{1}
$$

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 $\sum_{j=1}^N x_{ij} + y_{i+1,\omega}^+ - y_{i\omega}^+$.

Reformulate the objective function,

$$
\begin{aligned}
& E_\omega \left[\sum_{i=1}^{M-1} (n_i - s)(\sum_{j=1}^{N} x_{ij} + y_{i+1,\omega}^+ - y_{i\omega}^+) + (n_M - s)(\sum_{j=1}^{N} x_{Mj} - y_{M\omega}^+) \right] \\
=& \sum_{j=1}^{N}\sum_{i=1}^{M}(n_i - s)x_{ij} - \sum_{\omega=1}^{S} p_\omega \left(\sum_{i=1}^{M}(n_i - s)y_{i\omega}^+ - \sum_{i=1}^{M-1}(n_i - s)y_{i+1,\omega}^+ \right) \\
=& \sum_{j=1}^{N}\sum_{i=1}^{M} i \cdot x_{ij} - \sum_{\omega=1}^{S} p_\omega \left(y_{1\omega}^+ + \sum_{i=2}^{M} y_{i\omega}^+ \right)
\end{aligned}
$$

The last equality holds because of $n_i - s = i, i \in \mathcal{M}$. 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 row length constraint can be expressed as $\mathbf{nx} \le \mathbf{L}$.

The linear constraints associated with scenarios 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 the 1-vector of size N, $\mathbf{V} = [\mathbf{W},\ \mathbf{I}]$.

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

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

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

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, let us write this problem in the form of the two-stage stochastic programming.

Let $\mathbf{c}'\mathbf{x} = \sum_{j=1}^{N} \sum_{i=1}^{M} i \cdot x_{ij}$, $\mathbf{f}'\mathbf{y}_\omega = -\sum_{i=1}^{M} y_{i\omega}^+$. Then the formulation (1) can be expressed as below,

$$\begin{aligned} \max \quad & c'\mathbf{x} + z(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{n}\mathbf{x} \leq \mathbf{L} \\ & \mathbf{x} \in \mathbb{Z}_+^{M \times N}, \end{aligned} \tag{2}$$

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

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

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

$$\begin{aligned} z_\omega(\mathbf{x}) := \max \quad & \mathbf{f}'\mathbf{y}_\omega \\ \text{s.t.} \quad & \mathbf{x1} + \mathbf{V}\mathbf{y}_\omega = \mathbf{d}_\omega \\ & \mathbf{y}_\omega \geq 0. \end{aligned} \tag{3}$$

Here E is the expectation with respect to the scenario set. Problem (3) stands for the second-stage problem and $z_\omega(\mathbf{x})$ is the optimal value of problem (3), together with the convention $z_\omega(\mathbf{x}) = \infty$ if the problem is infeasible.

It is difficult to solve the above problem directly, we can relax problem (2) to stochastic linear programming firstly. In section 4.2, we obtain an optimal linear solution by decomposition approach and

generate a feasible seat planning.

4.2 Solve the Scenario-based Two-stage Problem

At first, we generate a closed-form solution to the second-stage problem in section 4.2.1. Then we obtain the solution to the linear relaxation of problem (2) by the delayed constraint generation. Finally, we obtain a feasible seat planning from the linear solution.

4.2.1 Solve the Second Stage Problem

Consider a $\mathbf{x}$ such that $\mathbf{nx} \leq \mathbf{L}$ and $\mathbf{x} \geq 0$ and suppose that this represents our seat planning for the first stage decisions. Once $\mathbf{x}$ is fixed, the optimal second stage decisions $\mathbf{y}_\omega$ can be determined by solving problem (3) for each ω.

To solve this problem, we should only consider that $\mathbf{x}$ for which $z_\omega(\mathbf{x})$ are all finite. Notice that the feasible region of the dual of problem (3) does not depend on $\mathbf{x}$. We can form its dual problem, which is

$$\begin{aligned} \min \quad & \alpha'_\omega(\mathbf{d}_\omega - \mathbf{x1}) \\ \text{s.t.} \quad & \alpha'_\omega \mathbf{V} \geq \mathbf{f}' \end{aligned} \tag{4}$$

Let $P = \{\alpha | \alpha' V \geq \mathbf{f}'\}$. We assume that P is nonempty and has at least one extreme point. Then, either the dual problem (4) has an optimal solution and $z_\omega(\mathbf{x})$ is finite, or the primal problem (3) is infeasible and $z_\omega(\mathbf{x}) = \infty$.

Let $\mathcal{O}$ be the set of all extreme points of P and $\mathcal{F}$ be the set of all extreme rays of P. Then $z_\omega > -\infty$ if and only if $(\alpha^k)'(\mathbf{d}_\omega - \mathbf{x1}) \geq 0, \alpha^k \in \mathcal{F}$, which stands for the feasibility cut.

Lemma 3. *The feasible region of problem (4), P, is bounded. In addition, all the extreme points of P are integral.*

(Proof of lemma 3). *Notice that $V = [W,\ I]$, W is a totally unimodular matrix. Then, we have $\alpha' W \geq -\bar{n}, \alpha' I \geq 0$. Thus, the feasible region is bounded. Furthermore, $\bar{n}_{i+1} = n_{i+1} - n_i = 1, i = 1, \ldots, M-1$ are integral, so the extreme points are all 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 $(\alpha^k)'(\mathbf{d}_\omega - \mathbf{x1}) \geq z_\omega, \alpha^k \in \mathcal{O}$, which is the optimality cut.

Corollary 1. *Only the optimality cuts, $\alpha'(\mathbf{d}_\omega - \mathbf{x1}) \geq z_\omega$, will be included in the decomposition approach.*

Corollary 2. *When $n_i = i+1$, $f' = [-\mathbf{1},\ \mathbf{0}], V = [W,\ I]$, we have $\alpha' W \geq -1, \alpha' I \geq 0$. Thus, it is easy to find that the feasible region is bounded, i.e., P does not contain any extreme rays. Furthermore, let $\alpha_0 = 0$, then we have $0 \leq \alpha_i \leq \alpha_{i-1} + 1$, $i \in \mathcal{M}$.*

Corollary 3. *The optimal value of the problem (3), $z_\omega(x)$, is finite and will be attained at extreme points of the set P. Thus, we have $z_\omega(x) = \min_{\alpha^j \in \mathcal{O}} (\alpha^k)'(\mathbf{d}_\omega - \mathbf{x1})$.*

When we are given x^*, the demand that can be satisfied by the assignment is $\mathbf{x}^*\mathbf{1} = \mathbf{d}_0 = (d_{1,0}, \ldots, d_{M,0})$. Then plug them in the subproblem (3), we can obtain the value of $y_{i\omega}$ recursively:

$$\begin{aligned} y^-_{M\omega} &= (d_{M\omega} - d_{M0})^+ \\ y^+_{M\omega} &= (d_{M0} - d_{M\omega})^+ \\ y^-_{i\omega} &= \left(d_{i\omega} - d_{i0} - y^+_{i+1,\omega}\right)^+, i = 1, \ldots, M-1 \\ y^+_{i\omega} &= \left(d_{i0} - d_{i\omega} + y^+_{i+1,\omega}\right)^+, i = 1, \ldots, M-1 \end{aligned} \tag{5}$$

The optimal value for scenario ω can be obtained by $f' y_\omega$, then we need to find the dual optimal solution.

Theorem 1. *The optimal solutions to problem (4) are given by*

$$\begin{aligned} &\alpha_{i\omega} = 0, i \in \mathcal{M} \quad \text{if } y^-_{i\omega} > 0 \\ &\alpha_{i\omega} = \alpha_{i-1,\omega} + 1, i \in \mathcal{M} \quad \text{if } y^+_{i\omega} > 0 \end{aligned} \tag{6}$$

For some i, when $y^+_{i\omega} = 0$ and $y^-_{i\omega} = 0$, $\left(d_{i0} - d_{i\omega} + y^+_{i+1,\omega}\right) = 0$, $d_{i\omega} - d_{i0} = y^+_{i+1,\omega} \geq 0$. If $y^+_{i+1,\omega} > 0$, $\alpha_{i\omega} = 0$; if $y^+_{i+1,\omega} = 0$, $0 \leq \alpha_{i\omega} \leq \alpha_{i-1,\omega} + 1$.

(Proof of Theorem 1). *According to the complementary relaxation property, when $d_{i\omega} > d_{i0} \Rightarrow y^-_{i\omega} > 0$, then $\alpha_{i\omega} = 0$ for all i; when $d_{i\omega} < d_{i0} \Rightarrow y^+_{i\omega} > 0$, then $\alpha_{i\omega} = \alpha_{i-1,\omega} + 1, i \in \mathcal{M}$.*

When $d_{i\omega} = d_{i0}$, we can find that $\alpha_{i\omega} = \alpha_{i-1,\omega} + 1$ will minimize the objective function.

Let $\Delta d = d_\omega - d_0$, then the elements in Δd will be a negative integer, positive integer and zero. Only the negative element will affect the objective function. The larger the value of α associated with a negative integer is, the smaller the objective function will be. Thus, let $\alpha_{i\omega} = \alpha_{i-1,\omega} + 1$ when $d_{i\omega} = d_{i0}$ can obtain the minimized objective function. □

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 the original large-scale linear programming.

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.

Use the characterization of $z_\omega(x)$ in the problem (2) and take into account the optimality cut, we can conclude the BMP will have the form:

$$
\begin{aligned}
\max \quad & c'x + \sum_{\omega \in \Omega} p_\omega z_\omega \\
\text{s.t.} \quad & \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} \\
& (\alpha^k)'(\mathbf{d}_\omega - \mathbf{x1}) \geq z_\omega, \alpha^k \in \mathcal{O}, \forall \omega \\
& \mathbf{x} \geq 0
\end{aligned} \tag{7}
$$

When substituting $\mathcal{O}$ with its subset, $\mathcal{O}^t$, the problem (7) 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.*

(Proof of lemma 4)**.** *Suppose we have one extreme point α^ω for each scenario. Then we have the following problem.*

$$
\begin{aligned}
\max \quad & c'x + \sum_{\omega \in \Omega} p_\omega z_\omega \\
s.t. \quad & \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} \\
& (\alpha^\omega)'\mathbf{d}_\omega \geq (\alpha^\omega)'\mathbf{x1} + z_\omega, \forall \omega \\
& \mathbf{x} \geq 0
\end{aligned} \tag{8}
$$

Problem (8) reaches its maximum when $(\alpha^\omega)'\mathbf{d}_\omega = (\alpha^\omega)'\mathbf{x1} + z_\omega, \forall \omega$. Substitute z_ω with these equations, we have

$$
\begin{aligned}
\max \quad & c'x - \sum_{\omega} p_\omega (\alpha^\omega)'\mathbf{x1} + \sum_{\omega} p_\omega (\alpha^\omega)'\mathbf{d}_\omega \\
s.t. \quad & \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} \\
& \mathbf{x} \geq 0
\end{aligned} \tag{9}
$$

Notice that $\mathbf{x}$ is bounded by $\mathbf{L}$, then the problem (8) is bounded. Adding more optimality cuts will not make the optimal value larger. Thus, RBMP is bounded. □

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 $c'\mathbf{x}_0 + \sum_{\omega \in \Omega} p_\omega z_\omega^0$ is an upper bound of problem (7).

When $\mathbf{x}_0$ is given, the optimal solution, α_ω^1, to problem (4) can be obtained according to Theorem 1. $z_\omega^{(0)} = \alpha_\omega^1(d_\omega - \mathbf{x}_0\mathbf{1})$ and $(\mathbf{x}_0, \mathbf{z}_\omega^{(0)})$ is a feasible solution to problem (7) because it satisfies all the constraints. Thus, $c'\mathbf{x}_0 + \sum_{\omega \in \Omega} p_\omega \mathbf{z}_\omega^{(0)}$ is a lower bound of problem (7).

If for every scenario, the optimal value of the corresponding problem (4) is larger than or equal to z_ω^0, all contraints are satisfied, we have an optimal solution, (x_0, z_ω^0), to the BMP. Otherwise, add one new constraint, $(\alpha_\omega^1)'(\mathbf{d}_\omega - \mathbf{x1}) \geq z_\omega$, to RBMP.

The steps of the algorithm are described as below,

Algorithm 1 The benders decomposition algorithm

Step 1. Solve LP (8) 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 $UB = c'\mathbf{x}_0 + \sum_{\omega\in\Omega} p_\omega z_\omega^0$.

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

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

Remark 1. *From the Lemma 4, we can set $\alpha_\omega^0 = \mathbf{0}$ initially in Step 1.*

Remark 2. *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 $\mathbf{x}^*\mathbf{1} = d_0 = (d_{1,0}, \ldots, d_{M,0})$. Then we can obtain the value of $y_{i\omega}$ from equation (5).

We show the results of Benders and IP in the section 6.1.

4.3 Obtain the Feasible Seat Planning

The decomposition method only gives a fractional solution and the stochastic model does not provide an appropriate seat planning when the number of people in scenario demands is way smaller than the number of the seats. Thus, we change the linear solution from the decomposition method to obtain a feasible seat planning. Before that, we will discuss the deterministic model that can help achieve the goal.

When $|\Omega| = 1$ in problem (1), the stochastic programming will be

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{M}\sum_{j=1}^{N}(n_i - s)x_{ij} - \sum_{i=1}^{M} y_i^+ \\
\text{s.t.} \quad & \sum_{j=1}^{N} x_{ij} - y_i^+ + y_{i+1}^+ + y_i^- = d_i, \quad i = 1, \ldots, M-1, \\
& \sum_{j=1}^{N} x_{ij} - y_i^+ + y_i^- = d_i, \quad i = M, \\
& \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} \\
& y_i^+, y_i^- \in \mathbb{Z}_+, \quad i \in \mathcal{M} \\
& x_{ij} \in \mathbb{Z}_+, \quad i \in \mathcal{M}, j \in \mathcal{N}.
\end{aligned} \tag{10}
$$

To maximize the objective function, we can take $y_i^+ = 0$. Notice that $y_i^- \geq 0$, thus the constraints $\sum_{j=1}^{N} x_{ij} + y_i^- = d_i, i \in \mathcal{M}$ can be rewritten as $\sum_{j=1}^{N} x_{ij} \leq d_i, i \in \mathcal{M}$, then we have

$$\begin{aligned} \max \quad & \sum_{i=1}^{M}\sum_{j=1}^{N}(n_i - s)x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{N} x_{ij} \leq d_i, \quad i \in \mathcal{M}, \\ & \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} \\ & x_{ij} \in \mathbb{Z}_+, \quad i \in \mathcal{M}, j \in \mathcal{N}. \end{aligned} \tag{11}$$

Problem (11) represents the deterministic model. Demand, $d_i, i \in \mathcal{M}$ is known in advance, our goal is to accommodate as many as people possible in the fixed rows.

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), which is strongly NP-hard as we all known.

Consider the solution to the linear relaxation of (11). 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 $u_{\text{MKP}} = \sum_{j=1}^{b-1} p_j + \left(L - \sum_{j=1}^{b-1} w_j\right) \frac{p_b}{w_b}$.

Let $\sum_{j=1}^{N} x_{ij}$ indicate the supply for group type i. Denote by $\mathbf{X} = (\sum_{j=1}^{N} x_{1j}, \ldots, \sum_{j=1}^{N} x_{Mj})$ the aggregate solution to the linear relaxation of problem (11).

Lemma 5. *Let e_i denote the unit size of the i-th element of $\mathbf{X}$. Suppose item b is in type h, then the aggregate solution is $xe_h + \sum_{i=h+1}^{M} d_i e_i$, where $x = (L - \sum_{i=h+1}^{M} d_i n_i)/n_h$.*

Suppose we obtain the optimal linear solution x_{ij}^* from the stochastic model, set the supply $\mathbf{s}^0 = \sum_j x_{ij}^*$ as the upper bound of demand in problem (11). We can get a feasible integer solution by solving this problem, denote by $\mathbf{s}^1$ the corresponding supply. As we mentioned above, this solution does not utilize the empty seats when the scenario demands are smaller than supply. Thus, we should set the supply $\mathbf{s}^1$ as the lower bound of demand, then re-solve a seat assignment problem. We substitute the constraint $\sum_{j=1}^{N} x_{ij} \leq d_i, i \in \mathcal{M}$ with the new constraint $\sum_{j=1}^{N} x_{ij} \geq s_i^1, i \in \mathcal{M}$ in problem (11), s_i^1 represents the number of group type i we must allocate seats.

$$\{\max \sum_{j=1}^{N}\sum_{i=1}^{M}(n_i - s)x_{ij} : \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N}; \sum_{j=1}^{N} x_{ij} \geq s_i^1, i \in \mathcal{M}; x_{ij} \in Z^+\} \tag{12}$$

The optimal solution to this problem with the lower bound will give a better seat assignment. The numerical results show that this seat assignment has good performances under any stochastic demands, and also shows good results when dealing with the dynamic demands.

Algorithm 2 Feasible seat planning algorithm

Step 1. Obtain the solution, $\mathbf{x}^*$, from stochatic linear programming by benders decomposition.

Step 2. Aggregate the solution to the supply, $s_i^0 = \sum_j x_{ij}^*$.

Step 3. Obtain the optimal solution, $\mathbf{x}^1$, from problem (11) by setting the supply $\mathbf{s}^0$ as the upper bound.

Step 4. Aggregate the solution to the supply, $s_i^1 = \sum_j x_{ij}^1$.

Step 5. Obtain the optimal solution, $\mathbf{x}^2$, from problem (12) by setting the supply $\mathbf{s}^1$ as the lower bound.

Step 6. Aggregate the solution to the supply, $s_i^2 = \sum_j x_{ij}^2$, which is the feasible seat planning.

Remark 3. *Step 3 can give a feasible integer supply. In Step 5, problem (11) with this supply as the lower bound can always give an integer solution. Thus, we can obtain the near-optimal seat assignment by solving stochastic programming once and deterministic programming twice.*

Remark 4. *For a feasible seat planning, we provide a 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(DSA)

This section discusses policies for assigning seats under conditions of stochastic information. We first present the dynamic seat assignment policy for each group arrival, which incorporates group-type control and loss control to optimize resource utilization. We also compare this approach with the bid-price policy. Next, we discuss the dynamic seat assignment policy for all group arrivals together. Finally, we establish the FCFS policy as the benchmark.

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. Suppose there are T independent periods, one group will arrive in each period. There are still M different group types. Let $\mathbf{y}$ be a discrete random variable indicating the number of people in the group. Let $\mathbf{p}$ be the vector probability, where $p(y = i) = p_i$, $i \in \mathcal{M}$ and $\sum_i p_i = 1$.

5.1 Dynamic Seat Assignment for Each Group Arrival

In this situation, we not only need to decide whether to accept or reject an arrival, but also assign seats to each group if we accept it.

5.1.1 Group-type(Supply) Control

For a feasible seat planning, we must follow some basic rules to assign seats.

- When the supply of one arriving group is enough, we will accept the group directly.
- When the supply of one arriving group is 0, the demand can be satisfied by only one larger-size supply.
- When one group is accepted to occupy the larger-size seats, the rest empty seat(s) can be reserved for future demand.

We can assign the seats to the corresponding-size group. But when a group comes while the corresponding supply is 0, should we give this group to the larger-size seats? Now we demonstrate the split larger group rule for this problem.

Suppose we accept a group of i to take over $(j+s)$-size seats(of group type j). In that case, the expected served people is $i + (j - i - s)P(D_{j-i-s} \geq x_{j-i-s} + 1)$, where $i + s \leq j$, $P(D_i \geq x_i)$ is the probability of that the expected demand of group type i in the following periods is no less than x_i, the remaining supply of group type i.

When a group of i occupies $(j+s)$-size seats, $(j-i-s)$ seats can be provided for one group of $j-i-s$ with s seats of social distancing. Thus, the term, $P(D_{j-i-s} \geq x_{j-i-s}+1)$, indicates the probability that the demand of group type $(j-i-s)$ in the future is no less than its current remaining supply plus 1. If $j - i - s \leq 0$, this term equals 0.

Similarly, when the expected demand of a group of j in the future is no less than its remaining supply currently, we would reject a group of i, the expected served people is $jP(D_j \geq x_j)$.

Let $d(i, j)$ be the difference of expected served people between acceptance and rejection on group i occupying $(j+s)$-size seats. If $j \geq i+s$, $d(i, j)$ equals $i+(j-i-s)P(D_{j-i-s} \geq x_{j-i-s}+1)-jP(D_j \geq x_j)$, otherwise, $d(i, j)$ equals $i - jP(D_j \geq x_j)$.

One intuitive decision is to choose the largest difference. We can obtain $d(i, j) = jP(D_j \leq x_j - 1) - (j-i-s)P(D_{j-i-s} \leq x_{j-i-s}) - 1$ after reformulating. Let $F_j(x; T_r)$ be the cumulative distribution function of the number of arrival groups D_j in T_r periods. Then $F_j(x; T_r) = P(D_j \leq x)$, and D_j follows a binomial distribution $B(T_r, p_j)$, where T_r is the numebr of remaining periods.

Thus, $d(i, j) = jF_j(x_j - 1; T_r) - (j - i - s)F_{j-i-s}(x_{j-i-s}; T_r) - 1$. 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 i in $(j^* + s)$-size seats. Otherwise, reject the group.

When the number of remaining period is large, the group-type control tends to accept smaller groups by breaking the largest group. But accepting one smaller group will result in permanent loss of seats as social distance, we need to reduce the number of accepting groups and try to accept the larger-size group as many as possible. Building on that idea, we have developed a loss control strategy. This approach enables us to minimize losses and optimize resource utilization under dynamic arrival conditions.

5.1.2 Loss Control

To implement this policy effectively, we need to incorporate the loss control. Specifically, when there is a large number of periods remaining, we employ both group-type control and loss control strate-

gies. However, when there are only a few periods remaining, we rely solely on group-type control to accommodate as many people as possible within the limited periods available.

Recall that the loss we defined above is the number of people lost. Denote by l_t the loss of up to current period t. When the social distance is s, accepting one group will incur s losses. Every time we break one group into smaller one, we will create another s losses. According to the initial seat planning, we can obtain the ideal loss l_i. This loss is close to the optimal loss, we try to get a closer loss when we reach the final seat assignment. Thus, we can develop a loss-control. The target ratio is $\frac{l_i}{T}$. The ratio at period t is $\frac{l_t}{t}$. At period $t+1$, if we accept one group according to seat planning, $l_{t+1} = l_t + s$; if we accept one group by breaking a larger group, $l_{t+1} = l_t + 2s$; if we reject the arrival, $l_{t+1} = l_t$. The decision rule is based on comparing the current ratio to the target ratio. If the current ratio is lower than the target ratio, we will accept the current arrival. However, if the current ratio is equal to or greater than the target ratio, we will reject the arrival.

It's difficult to achieve the ideal loss with dynamic arrivals. To accommodate more people, we only use group-type control when there are fewer remaining periods. This approach allows us to make the most of the limited periods we have and serve as many people as possible.

5.1.3 Algorithm

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

The algorithm is shown below:

Algorithm 3 Dynamic seat assignment 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', find the first $k \in \mathcal{N}$ 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.

Step 3. If $i \notin P_k, \forall k \in \mathcal{N}$, find $d(i, j^*)$. If $d(i, j^*) > 0$, find the first $k \in \mathcal{N}$ such that $j^* \in P_k$. Accept group type i and update $P_k = P_k/(j^*)$, $x_{j^*} = x_{j^*} - 1$. Then update $x_{j-i-s} = x_{j-i-s} + 1$ and $P_k = P_k \cup (j^* - i - s)$ when $j^* - i - s > 0$. 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 Bid-price

The dual problem of LP relaxation of (11) is:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{M} d_i z_i + \sum_{j=1}^{N} L_j \beta_j \\
\text{s.t.} \quad & z_i + \beta_j n_i \geq (n_i - s), \quad i \in \mathcal{M}, j \in \mathcal{N} \\
& z_i \geq 0, i \in \mathcal{M}, \beta_j \geq 0, j \in \mathcal{N}.
\end{aligned} \tag{13}
$$

When a group type i arrives, we can calculate $i - \beta_j n_i$ for all j and choose $\arg\max_j \{i - \beta_j n_i\}$ as the row to allocate that group. The bid-price control policy based on the static model is stated below.

Algorithm 4 Bid-price algorithm

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

Step 2. Solve (13) with $d_i^t = (T - t) \cdot p_i$ and $\mathbf{L}^t$, obtain an optimal solution β_j^t.

Step 3. Set $k = \arg\max_j \{i - \beta_j^t n_i | n_i \mathbf{e}_k^\top \leq \mathbf{L}\}$.

Step 4. If $i - \beta_k^t n_i \geq 0$, then assign the group to row k, update $\mathbf{L}^{t+1} = \mathbf{L}^t - n_i \mathbf{e}_k^\top$; otherwise, reject the group, let $\mathbf{L}^{t+1} = \mathbf{L}^t$.

Step 5. If $t \leq T$, move to next period, set $t = t + 1$, go to step 2. Otherwise, terminate this algorithm.

5.3 Dynamic Seat Assignment after All Group Arrivals

In this seat assignment situation, we first decide whether to accept or reject each group arrival based on the following DP. After all group arrivals have been considered, we then assign seats to each group.

Relax all rows to one row with the same capacity by $L = \sum_{j=1}^{N} L_j$. The deterministic problem is

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{M} (n_i - s) x_i \\
\text{s.t.} \quad & x_i \leq d_i, \quad i \in \mathcal{M}, \\
& \sum_{i=1}^{M} n_i x_i \leq L \\
& x_i \in \mathbb{Z}_+, \quad i \in \mathcal{M}.
\end{aligned} \tag{14}
$$

Lemma 6. *Let* $\mathbf{X}$ *be the solution of linear relaxation of problem* (14). $\mathbf{X}$ *is the same as the aggregate solution of linear relaxation of problem* (11).

Let v_r^* and v^* denote the optimal values of problem (14) and (11), respectively. After relaxation, assume that each row has a capacity of L seats, which can be filled. For each seperate row, the maximum number of empty seats in each row is s. Then, the total number of empty seats in N rows is given by Ns. Therefore, the biggest difference between v_r^* and v^* is the number of people accommodated in the Ns empty seats. The difference between them is zero only when the groups corresponding to the optimal solution of problem (14) can be accommodated in N rows. The numerical results indicate that this is

the case for most scenarios, which suggests that a DP-based approach can be used to solve the dynamic seat assignment problem after all groups have arrived.

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:

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

$i \geq V_{t+1}(L) - V_{t+1}(L - n_i) \approx \nabla V_{t+1}^{\top}(L) n_i$

After accepting each group, it is necessary to ensure that the accepted groups can be feasibly assigned seats. Conducting inspections at each stage can be cumbersome. To streamline the process, a threshold capacity can be developed. When the accepted groups fall within the prescribed threshold, they can be accepted directly using DP. However, when the accepted groups exceed the prescribed threshold, feasibility must be checked and the group should be rejected if a feasible seat assignment cannot be found.

The threshold capacity can be obtained by the following analysis. When the remaining capacity is no smaller than n_M for any row, we can always assign any group to this row. Thus, the largest number of taken seats for row j is $L_j - n_M + 1$ to ensure the next group can be placed. For N rows, the threshold capacity can be $\sum_{j=1}^{N}(L_j - n_M + 1)$.

5.4 Break tie

To determine which row to place accepted groups in when there are multiple options, follow these steps:

1. Number each row in ascending order.
2. Prioritize placing groups in rows from smallest to largest.
3. Check if the remaining capacity of the current row is greater than the maximum group size or equal to the current group size. If it is, accept the current arrival and place the group in that row.
4. Otherwise, consider the next row with the next higher number.
5. Repeat steps 3 and 4 until a row is found that can accommodate the current group size.

By following this approach, it is possible to efficiently find rows to place accepted groups while maximizing capacity utilization. This is achieved by filling as many rows as possible, starting from the row with the smallest number and only moving to the row with the larger number if necessary.

5.5 Benchmark

5.5.1 FCFS

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 the capacity of all rows is fully utilized.

5.5.2 FCFS-based

For dynamic seat assignment after all group arrivals, we can continue to use the first-come, first-served approach for seat assignment. Relax all rows to one row with the total number of seats. For each arrival, we need to check the feasibility of constructing a seat assignment in N rows. If the seat assignment is feasible, we accept the request; otherwise, we reject it. The threshold capacity is $(L-u+1)$.

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 feasible seat planning and Integer programming methods, analyzing different policies under the two booking situations, evaluating the results under varying demands, assessing the results for different numbers of people in each period and finally investigating the impact of seat layout on the number of served people.

6.1 Running time of Benders Decomposition and IP

The running times of solving IP directly and using Benders decomposition are shown in Table 1.

Table 1: Running time of Decomposion and IP

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

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 Feasible Seat Planning versus IP Solution

A 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 count 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 the feasible seat assignment has a close performance with IP when considering nested policy.

Table 2: Results of Feasible Seat Planning and 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.

We can find that the people served by Benders decomposition and IP under nested policy are close. But obtaining the near-optimal seat assignment will be faster.

6.3 Results of Different Policies

We compare the performance of different policies to the optimal value. Specifically, we consider four policies for seat assignment for each group arrival: DSA, DSA(mean), bid-price and FCFS. In

addition, we evaluate two policies for seat assignment after all group arrivals: one is based on dynamic programming (DP) and the other is based on first-come, first-served (FCFS) scheduling.

The seat layout consists of 10 rows, each containing 21 seats.

Table 3: Results of Policies

T	probabilities	DSA(%)	DSA(mean)	Bid-price	FCFS(%)	DP-based(%)	FCFS-based(%)
60	[0.25, 0.25, 0.25, 0.25]	98.92	99.10	98.19	98.48	99.48	99.31
70	[0.25, 0.25, 0.25, 0.25]	98.22	98.00	96.27	94.70	99.23	95.99
80	[0.25, 0.25, 0.25, 0.25]	98.04	98.42	96.00	93.08	99.39	94.08
60	[0.15, 0.25, 0.55, 0.05]	99.13	98.71	98.27	98.45	99.72	99.53
70	[0.15, 0.25, 0.55, 0.05]	99.19	99.12	97.16	96.24	99.30	97.65
80	[0.15, 0.25, 0.55, 0.05]	99.60	99.39	97.85	95.87	99.21	97.11
60	[0.25, 0.35, 0.05, 0.35]	98.98	98.64	98.14	98.19	99.62	99.33
70	[0.25, 0.35, 0.05, 0.35]	98.06	97.81	96.18	94.34	99.14	95.89
80	[0.25, 0.35, 0.05, 0.35]	97.97	98.33	95.81	92.82	99.31	94.16

6.4 Result of Different Demands

In this subsection, we discuss the effect of the number of periods on the results of our experiments. Assuming that groups of up to four people can sit together, we calculate the expected number of demands as $E(D) = (p_1 * 1 + p_2 * 2 + p_3 * 3 + p_4 * 4)T$, where p_1, p_2, p_3, and p_4 represent the probabilities of groups with one, two, three, and four people, respectively. Let $c = p_1 * 1 + p_2 * 2 + p_3 * 3 + p_4 * 4$ denote the number of people each period. Specifically, we consider two situations: $c = 2.5$ and $c = 1.9$.

When $c = 2.5$, we set the parameters as follows: T = 10-100, step size = 1, expected number of periods = 60, expected number of demand (people) = 150, number of rows = 10, and number of seats per row = 21.

When $c = 1.9$, we set the parameters as follows: T = 30-120, step size = 1, expected number of periods = 72, expected number of demand (people) = 137, number of rows = 10, and number of seats per row = 21.

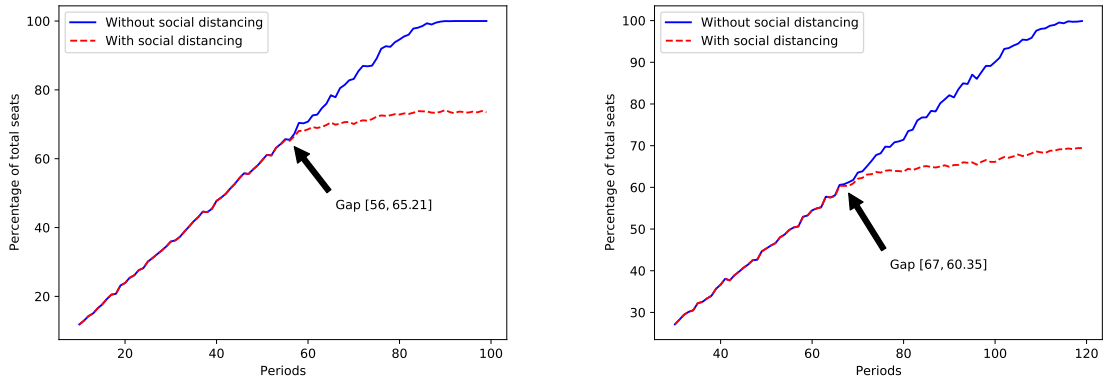
For each c, we give several probabilities in the table. The gap point represents the first period where the number of people without social distancing is larger than that with social distancing and the gap percentage is the corresponding percentage of total seats.

Table 4: Gap points of different probabilities

c	probabilities	gap point	gap percentage
2.5	[0.25,0.25,0.25,0.25]	56	65.21
2.5	[0.1,0.4,0.4,0.1]	55	65.59
2.5	[0.1, 0.5, 0.2, 0.2]	55	65.45
2.5	[0.2, 0.3, 0.3, 0.2]	54	64.56
2.5	[0.3, 0.2, 0.2, 0.3]	55	65.51
2.5	[0.2, 0.4, 0.1, 0.3]	55	65.41
1.9	[0.4, 0.4, 0.1, 0.1]	67	60.35
1.9	[0.5, 0.2, 0.2, 0.1]	67	58.9
1.9	[0.3, 0.5, 0.2, 0]	68	61.7
1.9	[0.6, 0.1, 0.1, 0.2]	66	58.31

We provide the results of our experiments for people accepted when applying social distance and

not applying social distance over different periods. The different probabilities with the same c share the similar pattern of the figure, so we only provide one case to show the detailed figure.



(a) When $c = 2.5$

(b) When $c = 1.9$

Figure 2: The number of people served versus periods

There are three stages, the first stage is when the capacity is sufficient. The measure of social distancing will not cause any effect. The gap is becoming larger as T increases at the second stage. At the third stage, as T continues to increase, the gap will converge when the capacity is limited.

We can estimate the attendance rate from $\frac{c}{c+1}*$ the number of seats. The number of periods will be $E(D)/c$. Thus, the government can make the policy on how much attendance rate can be established.

6.5 Results of the Number of Arriving People per Period

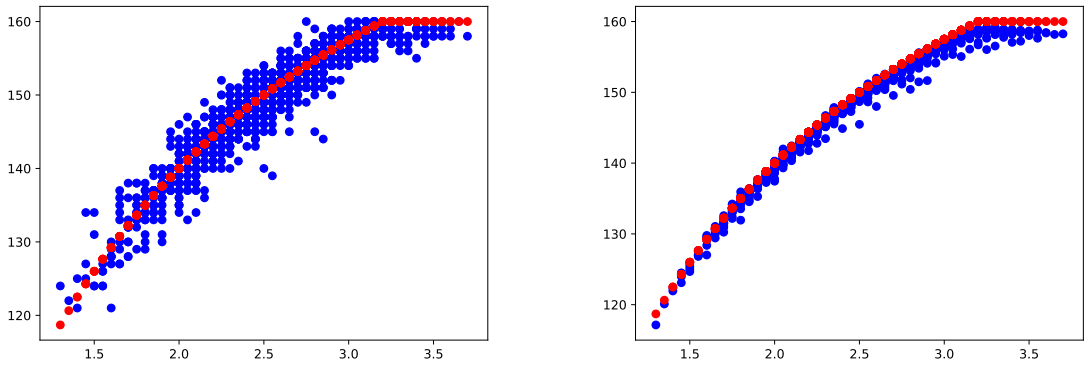
In this subsection, we discuss the effect of different probabilities on our experimental results. We choose $T(E(D)/c)$ such that the supply is near the demand. We then compare the number of people served under different values of c.

We sample p_1, p_2, and p_3 from 0.05 to 0.95 with an increment of 0.05. We call each realization (p_1, p_2, p_3, p_4) the probability combination when $p_1 + p_2 + p_3 + p_4 = 1$. The number of all sampled probability combinations is n_p. The number of rows is set at 10, and the number of seats per row is 21 (including one dummy seat). We simulate 200 periods, and the number of people served with respect to the value of c is shown in the figure below. For each probability combination, the blue point represents the average number of people served from 50 instances and the red point represents the expected number of people served.

Suppose we accept D_a people with T arrivals, where $D_a = c * T$, and the sum of D_a and T is equal to the total number of seats. In this case, the estimation of the occupancy rate is $\frac{c}{c+1}$. The number of people served is near $\frac{c}{c+1} * 210$ on average (as shown by the red points in the figure).

If the largest pattern is assigned for each row, the occupancy rate is $\frac{16}{21}$. The maximum number of people that can be served is $210 * \frac{16}{21} = 160$, which is the upper bound of the number of people served.

We also observe that some blue points are far from the red points in the figure. For example, when the probability is $[0.05, 0.05, 0.85, 0.05](c = 2.9)$, the demands can be $[4, 1, 45, 2]$ or $[2, 2, 47, 1]$. It is not



(a) One instance for each probability combination (b) Average of 50 instances for each probability combination

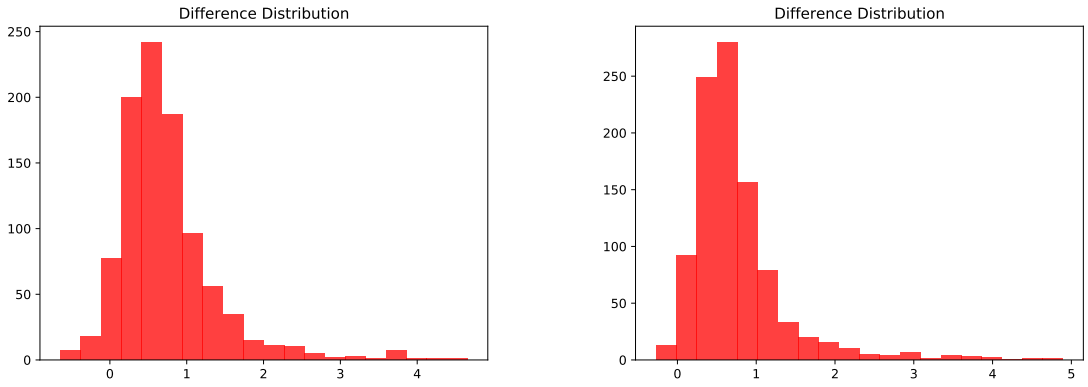
Figure 3: The number of people served versus c

possible to construct all full patterns for every row with these demands, violating our assumption and resulting in a large gap between the blue and red points in this case.

We can give the absolute difference between the blue point and red point for each probability combination as below.

Table 5: Difference Distribution

# of instances	abs_diff ≥ 1	abs_diff ≥ 2	abs_diff ≥ 3	abs_diff ≥ 4
20	32.92 %	5.13 %	1.74%	0.51 %
50	22.46 %	4.31 %	1.54 %	0.31 %
100	20.00 %	4.21 %	1.54 %	0.31 %



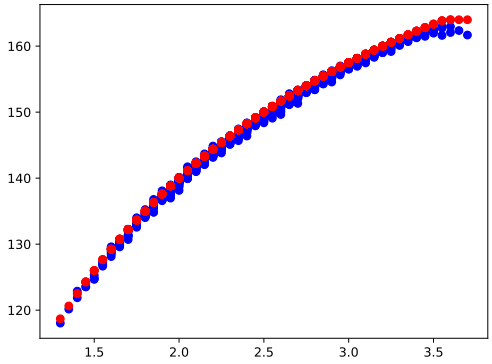
(a) Average of 50 instances for each probability combination (b) Average of 100 instances for each probability combination

Figure 4: The difference distribution

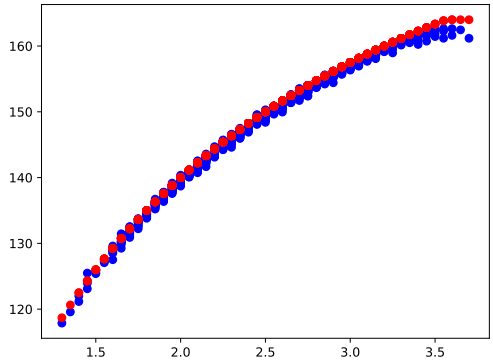
The results show that we can estimate attendance rate based on c for most probability combinations.

6.6 Results of Different Seat Layouts

We compare two seat layouts with the same total number of seats. The first layout has rows with 17, 18, 19, 20, 21, 21, 22, 23, 24, and 25 seats, while the second layout has rows with 19, 20, 21, 21, 23, 24, 26, 17, 19, and 20 seats. Both layouts can accommodate a maximum of 164 people when each row corresponds to a largest pattern.



(a) Average of 50 instances for step-size seat layout



(b) Average of 50 instances for random seat layout

Figure 5: The number of people served versus c

We can see that random seat layout will lead to a more accurate estimation, which means a random seat layout is likely to construct a full pattern so that to accept more people when facing the uncertain demands.

7 Conclusion

In conclusion, this paper addresses the problem of dynamic seat assignment with social distancing in the context of a pandemic. We propose a practical algorithm that balances seat utilization rates and the associated risk of infection to obtain a final seating plan that satisfies social distancing constraints when groups arrive. Our approach provides a comprehensive solution for optimizing seat assignments while ensuring the safety of customers. Our contributions include establishing a deterministic model to analyze the effects of social distancing when demand is known, using two-stage stochastic programming and Benders decomposition methods to obtain the optimal linear solution for stochastic demand situations, and developing a feasible seating plan using scenario-based stochastic programming for the dynamic scenario problem. Our results demonstrate significant improvements over baseline strategies and provide guidance for developing attendance policies. 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 in the post-pandemic era. Our study demonstrates the efficiency of obtaining the final seating plan using our proposed algorithm. The results indicate that our policy yields a seating plan that is very close to the optimal result. 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.