Seat Planning and Seat Assignment with Social Distancing

February 16, 2025

Abstract

This study tackles the challenge of seat planning and assignment with social distancing measures. Initially, we analyze seat planning with deterministic requests. Subsequently, we introduce a scenario-based stochastic programming approach to formulate seat planning with stochastic requests. We also investigate the dynamic situation where groups enter a venue and need to sit together while adhering to physical distancing criteria. Seat planning can serve as the foundation for assignment. Combined with relaxed dynamic programming, we propose a dynamic seat assignment policy for either accommodating or rejecting incoming groups. Our method outperforms traditional bid-price and booking-limit strategies. The findings furnish valuable insights for policymakers and venue managers regarding seat occupancy rates and provide a practical framework for implementing social distancing protocols while optimizing seat allocations.

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

Terminologies to use

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

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

1 Introduction

Social distancing is a proven concept for containing the spread of an infectious disease. It has been widely adopted worldwide, for example, during the most recent Covid 19 pandemic. As a general principle, social distancing measures can be specified from different dimensions. The basic requirement of social distancing is the specification of a minimum physical distance between people in public areas. For example, the World Health Organization (WHO) suggests social distancing as to "keep physical distance of at least 1 meter from others" [23]. In the US, the Center for Disease and Control (CDC) refers to social distancing as "keeping a safe space between yourself and other people who are not from your household" [6]. Note that under such a requirement, social distancing is actually applied with respect to groups of people. Similarly in Hong Kong, the government has adopted social distancing measures, in the recent Covid 19 pandemic, by limiting the size of groups in public gatherings to two, four, and six people per group over time. Moreover, the Hong Kong government has also adopted an upper limit on the total number of people in a venue; for example, restaurants can operate at 50% or 75% of their normal seating capacity.

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

The impact of enforcing social distancing measures on economic activities is also an important factor for governmental decision making. Facing an outbreak of an infectious disease, a government shall declare a social distancing policy based on a holistic analysis, considering not only the severity of the outbreak, but also the potential impact on all stakeholders. What is particularly important is the level of business loss suffered by the industries that are directly affected.

However, the requirement of social distancing is not applicable to all places. Strict physical distance requirements make some cinemas with small row spacing and seat spacing only able to accommodate fewer customers, making the implementation of this policy impractical. Therefore, we hope to develop a policy that is easy to implement and satisfies both the government and businesses to meet requirements, and evaluate the impact of these strategies, in order to provide insights for policy implementation and business operations.

We will address the above issues of social distancing in the context of seating management. Consider a venue, such as a cinema or a conference hall, which is to be used in an event. The venue is equipped with seats of multiple rows. In the event, requests for seats are in groups where each group contains a limited number of people. Any group can be accepted or rejected, and the people in an accepted group will sit consecutively in one row. Each row can accommodate multiple groups as long as any two adjacent groups in the same row are separated by one or multiple empty seats, as the requirement of the social distancing measures. The objective is to accept the number of individuals as many as possible.

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

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

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

In seat assignment, groups of seating requests arrive dynamically. The problem is to decide, upon the arrival of each group of request, whether to accept or reject the group, and assign seats for each accepted groups. Seat assignment can be used for those commercial applications where requests arrive as a stochastic process, for example, tickets selling in movie theaters.

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

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

We address the above issue from the following aspects.

1. We introduce the concept of gap point to characterize the situations in which social distancing begins to cause loss to an event. Roughly speaking, given a distribution of the group size of each request, the gap point can be specified as an upper bound of the number of requests in an event such that if an event has fewer requests than the gap point, then the event will virtually not be affected by social distancing. Our computational experiments show that the gap point depends mainly on the mean of the group size, and relatively insensitive to its exact distribution. This offers an easy way to estimate the gap point and the impact of social distancing.

2. Our models and analysis are developed for the social distancing requirement on the physical distance and group size, where we can determine a target occupancy rate for any given event in a venue, and a maximum achievable occupancy rate for all events. Sometimes the government also imposes a maximum allowable occupancy rate to tighten the social distancing requirement. This maximum allow-

able rate is effective for an event if it is lower than the target occupancy rate of the event. Furthermore, the maximum allowable rate will be redundant if it is higher than the maximum achievable rate for all events.

3. The above qualitative insights are stable with respect to different parameters in the model, such as the layout of the venue, the maximum group sizes and the minimum physical distances.

The rest of this paper is structured as follows. We review the relevant literature in Section 2. Then we introduce the major issues brought by social distancing and define the seating planning with deterministic requests in Section 3. In Section 4, we establish the stochastic model, analyze its properties and obtain the seat planning. Section 5 introduces the dynamic seat assignment problem. Section 6 demonstrates the dynamic seat assignment policy to assign the seats for incoming groups. Section 7 gives the numerical results and the insights of implementing social distancing. The conclusions are shown in Section 8.

2 Literature Review

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

2.1 Seat Planning with Social Distancing

Seating management is a practical problem that exists in many applications with different issues to handle, especially in the context of accommodating group-based seating requests. For example, in passenger rail services, work has been done on problems for maximizing capacity utilization or reducing the total capacity required [7, 9]. In another example, such as weddings or dinner events [18], the main focus is on satisfying customer preferences and enhancing the overall experience.

Including social distancing in seating management has added another dimension of consideration, forming a new stream of research. In some cases, it involves layout design, specifically, to determine the seating location within a given venue, for example, with the aim of maximizing the physical distancing between students in a classroom [5], and positioning tables in restaurants and beach umbrellas [10]. In other situations with predetermined seating layout, individuals are assigned seats while adhering to social distancing guidelines, for instance, problems in the air travel [12] and long-distance train travel [14]. Such work highlights the relevance and importance in seating management with the consideration of social distancing.

Our work belongs to seating management with social ditancing for group-based requests, which has found its applications across various areas, including airplanes [21], trains [15], sports arenas [17], and theaters [4]. Because of the diversity in applications, there are different issues to handle, for example, in [21], ...

Our work in [4] is the most related one to our research, both addressing group-based seating problem in theaters. In [4], the primarily focuses on the cases with known groups, which is referred to as seat planning with deterministic requests in this paper, we have a broader scope. We also consider group-based seat planning with stochastic requests. Additionally, we incorporate dynamic seat assignment, assuming that groups arrive with a certain probability, to provide a comprehensive solution pattern.

2.2 Dynamic Seat Assignment

In dynamic seat assignment, the decision to either reject or accept-and-assign groups is made at each stage upon their arrival. This problem can be regarded as a special case of the dynamic multiple knapsack problem. When there is one row, the related problem is dynamic knapsack problem [16]. Our model in its static form, deterministic request, can be viewed as a specific instance of the multiple knapsack problem [20]. There is little study, only one mildly related to the stochastic and dynamic multiple knapsack problem. It mentioned that **to be added**

Dynamic seat assignment has applications in the transportation industry, including airplanes, trains, and buses [2, 13]. This process involves assigning seats to passengers in a manner that maximizes the efficiency and convenience of seating arrangements, without considering the acceptance or rejection of requests.

Our work is closely related to the group-based network revenue management (RM) problem [24], which focuses on accepting or rejecting a request [11]. One of the characteristics we are studying is that the decision should be made on an all-or-none basis for each group, which is the real complication in group arrivals [22].

In hotel revenue management, group characteristics can also be observed in multi-day stays [1, 3], which differs from the concept of a group in our problem.

Another key characteristic of our study is the importance of seat assignment, which distinguishes it from traditional revenue management. The assign-to-seat feature introduced by Zhu et al. [25] further emphasizes the significance of seat assignment. This approach tackles the challenge of selling high-speed train tickets, where each request must be assigned to a specific seat for the entire journey. However, this paper focuses on individual passengers rather than groups, which sets it apart from our research.

3 Seat Planning Problem with Social Distancing

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

3.1 Concepts

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

To adhere to social distancing requirements, individuals from the same group must sit together in one specific row while maintaining a distance—measured by the number of empty seats—from individuals in different groups. Since each group occupies only one row, we assume that the physical distance between different rows is sufficient. If the social distancing requirement is more stringent, an empty row can be implemented, as practiced by the theater in Berlin [19].

Let δ denote the social distancing, which could entail leaving one or more empty seats. Specifically, each group must ensure the empty seat(s) with the adjacent group(s). To model the social distancing requirements into the seat planning process, we define the size of group type i as $n_i = i + \delta$, where $i \in \mathcal{M}$. Correspondingly, the size of each row is defined as $L_j = L_j^0 + \delta$. It is a clear one-to-one mapping between the original physical seat planning and the model of seat planning. By incorporating the additional seat(s) and designating certain seat(s) for social distancing, we can integrate social distancing measures into the seat planning problem.

We introduce the term, pattern, to represent the seat planning arrangement for a single row. A specific pattern can be represented by a vector $\boldsymbol{h} = (h_1, \ldots, h_M)$, where h_i is the number of group type i in the row for $i = 1, \ldots, M$. A feasible pattern, $\boldsymbol{h}$, must satisfy the condition $\sum_{i=1}^{M} h_i n_i \leq L$ and belong to the set of non-negative integer values, denoted as $\boldsymbol{h} \in \mathbb{N}^M$. Then a seat planning with N rows can be expressed by $\boldsymbol{H} = \{\boldsymbol{h}_1; \ldots; \boldsymbol{h}_N\}$.

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

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

In the model, $L = L^0 + 1 = 11$, $n_i = i + 1$ for $i = 1, 2, 3, 4$. Then, the row can be represented by $\boldsymbol{h} = (2, 1, 1, 0)$. The maximum number of individuals that can be accommodated is $|\boldsymbol{h}| = 7$.

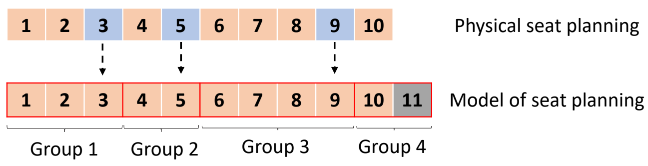


Figure 1: Illustration of Groups with Social Distancing

To formulate the deterministic model, let x_{ij} represent the number of group type i planned in row j. The seat planning problem with deterministic request is formulated below.

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

The objective is to maximize the number of individuals accommodated. Constraint (1) ensures the number of accommodated groups does not exceed the number of requests. Constraint (2) stipulates that the number of seats allocated in each row does not exceed the size of the row.

By examining the monotonic ratio between the original group sizes and the adjusted group sizes, we can establish the upper bound of the optimal value of Problem (1). This is illustrated in Proposition 1 and will be utilized in the bid-price control discussed in Section 9.

Proposition 1. *For the LP relaxation of problem (1), there exists an index $\tilde{i}$ such that the optimal solutions satisfy the following conditions: $x_{ij}^* = 0$ for all j, $i = 1, \ldots, \tilde{i}-1$; $\sum_j x_{ij}^* = d_i$ for $i = \tilde{i}+1, \ldots, M$; $\sum_j x_{ij}^* = \frac{L-\sum_{i=\tilde{i}+1}^{M} d_i n_i}{n_{\tilde{i}}}$ for $i = \tilde{i}$.*

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

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

3.2 Seat Planning Composed of Full or Largest Patterns

The seat planning obtained from problem (1) may not utilize all available seats, as it depends on the given requests. To improve a given seat planning and utilize all seats, we aim to generate a new seat planning composed of full or largest patterns while ensuring that the original group type requirements are met.

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

In other words, a full pattern is one in which the sum of the product of the number of occurrences h_i and the size n_i of each group in the pattern is equal to the size of the row L. This ensures that the pattern fully occupies the available row seats. A largest pattern is one that either has the maximum size or is equal in size to other patterns, ensuring that it can accommodate the maximum number of individuals within the given row size.

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

The size, $qM + \max\{r - \delta, 0\}$, corresponds directly to a largest pattern that includes q group type M and r seats for one group type $(r - \delta)$ when $r > \delta$. However, the form of the largest pattern is not unique; there are other largest patterns that share the same size.

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

Example 2. *Consider the given values: $\delta = 1$, $L = 21$, and $M = 4$. The size of the largest pattern can be calculated as $qM+\max\{r-\delta, 0\} = 4\times4+0 = 16$. The largest patterns are as follows: $(1, 0, 1, 3)$, $(0, 1, 2, 2)$, $(0, 0, 0, 4)$, $(0, 0, 4, 1)$, and $(0, 2, 0, 3)$. Among these, $(0, 0, 0, 4)$ is the form referenced in Proposition 2.*

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

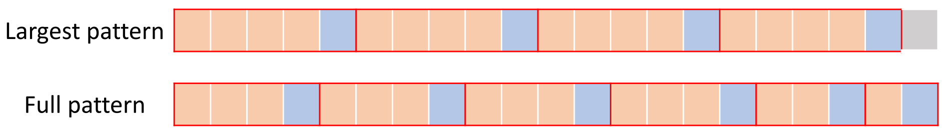


Figure 2: Largest and Full Patterns

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

To obtain a seat planning composed of the full or largest patterns, we make the following statements. Let the original seat planning be $\boldsymbol{H}$ and the desired seat planning be $\boldsymbol{H}'$. To satisfy requirements for the original group types, the total quantity of groups from type i to type M in $\boldsymbol{H}'$ must be at least equal to the total quantity from group type i to group type M in $\boldsymbol{H}$. Mathematically, we aim to find a feasible seat planning $\boldsymbol{H'}$ such that $\sum_{k=i}^{M}\sum_{j=1}^{N} H_{jk} \leq \sum_{k=i}^{M}\sum_{j=1}^{N} H_{jk}^{'}, \forall i \in \mathcal{M}$. We say $\boldsymbol{H} \subseteq \boldsymbol{H}^{'}$ if this condition is satisfied.

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

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

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

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

4 Seat Planning with Stochastic Requests

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

4.1 Scenario-based Stochastic Programming Formulation

Now suppose the demand of groups is stochastic, the stochastic information can be obtained from scenarios through historical data. Use ω to index the different scenarios, each scenario $\omega \in \Omega$. Regarding the nature of the obtained information, we assume that there are $|\Omega|$ possible scenarios. Each scenario associates with a particular realization of the request that 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 individuals accommodated over all the scenarios, we propose a scenario-based stochastic programming to obtain a seat planning.

Recall that the supply for group type i can be denoted as $\sum_{j=1}^{N} x_{ij}$. However, considering the variability across different scenarios, it is necessary to model the potential excess or shortage of supply.

To capture this characteristic, we introduce a scenario-dependent decision variable, denoted as $\mathbf{y}$. It includes two vectors of decisions, $\mathbf{y}^+ \in \mathbb{N}^{M\times|\Omega|}$ and $\mathbf{y}^- \in \mathbb{N}^{M\times|\Omega|}$. Each component of $\mathbf{y}^+$, denoted as $y^+_{i\omega}$, represents the excess of supply for group type i for each scenario ω. On the other hand, $y^-_{i\omega}$ represents the shortage of supply for group type i for each scenario ω.

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

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

where $(x)^+$ equals x if $x > 0$, 0 otherwise. Based on the above mentioned considerations, the total supply of group type i under scenario ω can be expressed as $\sum_{j=1}^{N} x_{ij} + y^+_{i+1,\omega} - y^+_{i\omega}, i = 1,\dots,M-1$. For the special case of group type M, the total supply under scenario ω is $\sum_{j=1}^{N} x_{Mj} - y^+_{M\omega}$.

Then we have the following formulation:

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

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

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

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

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

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

The objective function consists of two parts. The first part represents the number of individuals in the group type M that can be accommodated, given by $(n_M - \delta)(\sum_{j=1}^{N} x_{Mj} - y^+_{M\omega})$. The second part represents the number of individuals in group type i, excluding M, that can be accommodated, given by $(n_i - \delta)(\sum_{j=1}^{N} x_{ij} + y^+_{i+1,\omega} - y^+_{i\omega}), i = 1,\dots,M-1$. The overall objective function is subject to an expectation operator denoted by E_ω, which represents the expectation with respect to the scenario set.

This implies that the objective function is evaluated by considering the average values of the decision variables and constraints over the different scenarios.

By reformulating the objective function, we have

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

Here, $\sum_{j=1}^{N} \sum_{i=1}^{M} i \cdot x_{ij}$ indicates the maximum number of individuals that can be accommodated in the seat planning $\{x_{ij}\}$. The second part, $\sum_{\omega \in \Omega} p_\omega \sum_{i=1}^{M} y^{+}_{i\omega}$ indicates the expected excess of supply for group type i over scenarios.

In the optimal solution, at most one of $y^{+}_{i\omega}$ and $y^{-}_{i\omega}$ can be positive for any i, ω. Suppose there exist i_0 and ω_0 such that $y^{+}_{i_0\omega_0}$ and $y^{-}_{i_0\omega_0}$ are positive. Substracting $\min\{y^{+}_{i_0,\omega_0}, y^{-}_{i_0,\omega_0}\}$ from these two values will still satisfy constraints (6) and (7) but increase the objective value when p_{ω_0} is positive. Thus, in the optimal solution, at most one of $y^{+}_{i\omega}$ and $y^{-}_{i\omega}$ can be positive.

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

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

When there is only one scenario, the SSP reduces to the deterministic model. This aligns with Section 3.2, which outlines the generation of seat planning consisting of full or largest patterns.

Solving SSP directly is computationally prohibitive when there are numerous scenarios, instead, we apply Benders decomposition to simplify the solving process in Section 4.2, then obtain the seat planning composed of full or largest patterns, as stated in Section 4.3.

4.2 Solve SSP by Benders Decomposition

We reformulate problem (9) into a master problem and a subproblem (10). The iterative process of solving the master problem and subproblem is known as Benders decomposition. The solution obtained from the master problem provides inputs for the subproblem, and the subproblem solutions help update the master problem by adding constraints, iteratively improving the overall solution until convergence is achieved. Firstly, we generate a closed-form solution to problem (10), then we obtain the solution to the LP relaxation of problem (9) by the constraint generation.

4.2.1 Reformulation

Then, we reformulate SSP$(\mathbf{L}, \Omega)$ in a matrix form to apply the Benders decomposition technique. Let $\mathbf{n} = (n_1, \dots, n_M)^\intercal$ represent the vector of seat sizes for each group type, where n_i denotes the size of seats taken by group type i. Let $\mathbf{L} = (L_1, \dots, L_N)^\intercal$ represent the vector of row sizes, where L_j denotes the size of row j as defined previously. The constraint (8) can be expressed as $\mathbf{x}^\intercal \mathbf{n} \leq \mathbf{L}$. This constraint ensures that the total size of seats occupied by each group type, represented by $\mathbf{x}^\intercal \mathbf{n}$, does not exceed the available row sizes $\mathbf{L}$. We can use the product $\mathbf{x1}$ to indicate the supply of group types, where $\mathbf{1}$ is a column vector of size N with all elements equal to 1.

The linear constraints associated with scenarios, denoted by constraints (6) and (7), can be expressed in matrix form as:

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

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

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

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

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

As we can find, this deterministic equivalent form is a large-scale problem even if the number of possible scenarios Ω is moderate. However, the structured constraints allow us to simplify the problem by applying Benders decomposition approach. Before using this approach, we could reformulate this problem as the following form. Let $\mathbf{c}^\intercal \mathbf{x} = \sum_{j=1}^{N} \sum_{i=1}^{M} i \cdot x_{ij}$, $\mathbf{f}^\intercal \mathbf{y}_\omega = -\sum_{i=1}^{M} y_{i\omega}^+$. Then the SSP formulation can be expressed as below,

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

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

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

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

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

We can solve problem (9) quickly if we can efficiently solve problem (10). Next, we will mention how to solve problem (10).

4.2.2 Solve The Subproblem

Notice that the feasible region of the dual of problem (10) remains unaffected by $\mathbf{x}$. This observation provides insight into the properties of this problem. Let $\boldsymbol{\alpha}_\omega = (\alpha_{1\omega}, \alpha_{2\omega}, \dots, \alpha_{M,\omega})^\intercal$ denote the vector of dual variables. For each ω, we can form its dual problem, which is

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

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

Let $\mathbb{P}$ indicate the feasible region of problem (11). According to Lemma 1, the optimal value of the problem (10), $z_\omega(\mathbf{x})$, is finite and can be achieved at extreme points of $\mathbb{P}$. Let $\mathcal{O}$ be the set of all extreme points of $\mathbb{P}$. Then, we have $z_\omega(\mathbf{x}) = \min_{\boldsymbol{\alpha}_\omega \in \mathcal{O}} \boldsymbol{\alpha}_\omega^\intercal (\mathbf{d}_\omega - \mathbf{x}\mathbf{1})$.

Alternatively, $z_\omega(\mathbf{x})$ is the largest number z_ω such that $\boldsymbol{\alpha}_\omega^\intercal (\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_w, \forall \boldsymbol{\alpha}_\omega \in \mathcal{O}$. We use this characterization of $z_w(\mathbf{x})$ in problem (9) and conclude that problem (9) can thus be put in the form by setting z_w as the variable:

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

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

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

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

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

4.2.3 Constraint Generation

Due to the computational infeasibility of solving problem (12) with an exponentially large number of constraints, it is a common practice to use a subset, denoted as $\mathcal{O}^t$, to replace $\mathcal{O}$ in problem (12). This results in a modified problem known as the Restricted Benders Master Problem (RBMP). To find the optimal solution to problem (12), we employ the technique of constraint generation. It involves iteratively solving the RBMP and incrementally adding more constraints until the optimal solution to problem (12) is obtained.

We can conclude that the RBMP will have the form:

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

Given the initial $\mathcal{O}^t$, we can have the solution $\mathbf{x}^*$ and $\mathbf{z}^* = (z_1^*, \ldots, z_{|\Omega|}^*)$. Then $c^{\intercal}\mathbf{x}^* + \sum_{\omega\in\Omega} p_{\omega} z_{\omega}^*$ is an upper bound of problem (14). When $\mathbf{x}^*$ is given, the optimal solution, $\tilde{\boldsymbol{\alpha}}_{\omega}$, to problem (11) can be obtained according to Proposition 5. Let $\tilde{z}_{\omega} = \tilde{\boldsymbol{\alpha}}_{\omega}(d_{\omega} - \mathbf{x}^*\mathbf{1})$, then $(\mathbf{x}^*, \tilde{\mathbf{z}})$ is a feasible solution to problem (14) because it satisfies all the constraints. Thus, $\mathbf{c}^{\intercal}\mathbf{x}^* + \sum_{\omega\in\Omega} p_{\omega}\tilde{z}_{\omega}$ is a lower bound of problem (12).

If for every scenario ω, the optimal value of the corresponding problem (11) is larger than or equal to z_{ω}^*, which means all contraints are satisfied, then we have an optimal solution, $(\mathbf{x}^*, \mathbf{z}^*)$, to problem (12). However, if there exists at least one scenario ω for which the optimal value of problem (11) is less than z_{ω}^*, indicating that the constraints are not fully satisfied, we need to add a new constraint $(\tilde{\boldsymbol{\alpha}}_{\omega})^{\intercal}(\mathbf{d}_{\omega} - \mathbf{x}\mathbf{1}) \geq z_{\omega}$ to RBMP.

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

Proposition 6. *RBMP is bounded when there is at least one constraint for each scenario.*

From Proposition 6, we can set $\boldsymbol{\alpha}_{\omega} = \mathbf{0}$ initially. Notice that only contraints are added in each iteration, thus UB is decreasing monotone over iterations. Then we can use $UB - LB < \epsilon$ to terminate the algorithm.

However, solving problem (14) even with the simplified constraints directly can be computationally challenging in some cases, so practically we first obtain the optimal solution to the LP relaxation of problem (9). Then, we generate an integral seat planning from this solution.

4.3 Obtain The Feasible Seat Planning

We may obtain a fractional optimal solution when we solve the LP relaxation of problem (9). This solution represents the optimal allocations of groups to seats but may involve fractional values, indicating

Algorithm 1: Benders Decomposition

Input: Initial problem (14) with $\boldsymbol{\alpha}_\omega = 0, \forall \omega$, $LB = 0$, $UB = \infty$, ϵ.
Output: $\mathbf{x}^*$

while $UB - LB > \epsilon$ **do**
  Obtain $(\mathbf{x}^*, \mathbf{z}^*)$ from problem (14);
  $UB \leftarrow c^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega z_\omega^*$;
  for $\omega = 1, \ldots, |\Omega|$ **do**
    Obtain $\tilde{\boldsymbol{\alpha}}_\omega$ from Proposition 5;
    $\tilde{z}_\omega = (\tilde{\boldsymbol{\alpha}}_\omega)^\intercal (\mathbf{d}_\omega - \mathbf{x}^* \mathbf{1})$;
    if $\tilde{z}_\omega < z_\omega^*$ **then**
      Add one new constraint, $(\tilde{\boldsymbol{\alpha}}_\omega)^\intercal (\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega$, to problem (14);
    end
  end
  $LB \leftarrow c^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega \tilde{z}_\omega$;
end

partial assignments. Based on the fractional solution obtained, we use the deterministic model to generate a feasible seat planning. The objective of this model is to allocate groups to seats in a way that satisfies the supply requirements for each group without exceeding the corresponding supply values obtained from the fractional solution. To accommodate more groups and optimize seat utilization, we aim to construct a seat planning composed of full or largest patterns based on the feasible seat planning obtained in the last step.

Let the optimal solution to the LP relaxation of problem (14) be $\mathbf{x}^*$. Aggregate $\mathbf{x}^*$ to the number of each group type, $\tilde{X}_i = \sum_j x_{ij}^*, \forall i \in \mathbf{M}$. Solve the SPP($\{\tilde{X}_i\}$) to obtain the optimal solution, $\tilde{\mathbf{x}}$, and the corresponding pattern, $\boldsymbol{H}$, then generate the seat planning by problem (3) with $\boldsymbol{H}$.

Algorithm 2: Seat Planning Construction

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

5 Dynamic Seat Assignment with Social Distancing

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

To model this problem, we adopt a discrete-time framework. Time is divided into T periods, indexed forward from 1 to T. We assume that in each period, at most one request arrives and the probability of an arrival for a group type i is denoted as p_i, where i 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 each period. To simplify the analysis, we assume that the arrivals of different group types are independent and the arrival probabilities remain constant over time. This assumption can be extended to consider dependent arrival probabilities over time if necessary.

At time t, the state of remaining capacity in each row is represented by a vector $\mathbf{L}^t = (l_1^t, l_2^t, \dots, l_N^t)$, where l_j^t denotes the number of remaining seats in row j at time t. Upon the arrival of a group type i at time t, the seller needs to make a decision denoted by $u_{i,j}^t$, where $u_{i,j}^t = 1$ indicates acceptance of group type i in row j during period t, while $u_{i,j}^t = 0$ signifies rejection of that group type in row j. The feasible decision set is defined as

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

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

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

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

with the boundary conditions $V^{T+1}(\mathbf{L}) = 0, \forall \mathbf{L}$ which implies that the revenue at the last period is 0 under any capacity.

Initially, we have the current remaining capacity vector denoted as $\mathbf{L} = (L_1, L_2, \dots, L_N)$. Our objective is to make group assignments that maximize the total expected revenue during the horizon from period 1 to T which is represented by $V^1(\mathbf{L})$.

Solving the dynamic programming problem described in equation (15) can be challenging due to the curse of dimensionality, which arises when the problem involves a large number of variables or states. To mitigate this complexity, we aim to develop a heuristic method for assigning arriving groups. In our approach, we begin by generating a seat planning, as outlined in section 4. This seat planning acts as a foundation for the seat assignment. In section 6, building upon the generated seat planning, we further develop a dynamic seat assignment policy which guides the allocation of seats to the incoming requests sequentially.

6 Seat Assignment with Dynamic Demand

In this section, we propose our policy for assigning arriving requests in a dynamic context. First, we employ relaxed dynamic programming to determine whether to prepare a request for assignment or to reject it. Then, we make seating allocation decisions based on the seat planning strategy outlined in Section 4.

6.1 DP-based Heuristic

To simplify the complexity of the original DP (15), we consider a simplified version by relaxing all rows to a single row with the same total capacity, denoted as $\tilde{L} = \sum_{j=1}^{N} L_j$. Using the relaxed dynamic programming approach, we can determine the seat assignment decisions for each group arrival. Let u denote the decision, where $u_i^t = 1$ if we accept a request of type i in period t, $u_i^t = 0$ otherwise. Similarly to the DP in Section 5, the DP with one row can be expressed as:

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

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

After accepting a group, assign it to some row arbitrarily when the capacity of that row allows.

Algorithm 3: DP-based Heuristic Algorithm

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

Here, we encounter some straightforward scenarios. If the size of an arriving group exceeds the maximum remaining length of any row, we reject it. Conversely, if the size of the arriving group exactly matches the remaining length of a particular row, we accept it.

Since this policy does not guide specific assignment methods, we proceed with the assignment based on the planning strategy.

6.2 Assignment Based on Planning

In this section, we assign groups based on the planning that includes full or largest patterns. When a group type i is ready to be assigned by the DP approach, if the corresponding supply in the planning $X_i > 0$, we allocate seats according to the tie-breaking rule. If $X_i = 0$, we implement the group-type control policy to decide whether to assign the group to a specific row. We will also discuss the tie-breaking rule for assigning specific rows. Finally, we will address the conditions for regenerating the seat planning.

In the following part, we will refer to this policy as Dynamic Seat Assignment (DSA).

6.2.1 Group-type Control

The group-type control aims to determine the group type to assign the arriving group, thereby narrowing down the options for row selection during seat assignment. This policy evaluates whether to utilize the supply of larger group seats to accommodate the arriving group, given the current seat planning.

When a group is accepted and assigned to larger-size seats, the remaining empty seat(s) can be reserved for future demand without affecting the rest of the seat planning. To determine whether to use larger seats to accommodate the incoming group, we compare the expected number of acceptable individuals of accepting the group in the larger seats and rejecting the group based on the current seat planning. Then we identify the possible rows where the incoming group can be assigned based on the group types and seat availability.

Specifically, suppose the supply is $[X_1, \ldots, X_M]$ at period t, the number of remaining periods is $(T-t)$. For the arriving group type i when $X_i = 0$, we demonstrate how to decide whether to accept the group to occupy larger-size seats. For any $\hat{\imath} = i+1, \ldots, M$, we can use one supply of group type $\hat{\imath}$ to accept a group type i. In that case, when $\hat{\imath} = i+1, \ldots, i+\delta$, the expected number of accepted individuals is i and the remaining seats beyond the accepted group, which is $\hat{\imath} - i$, will be wasted. When $\hat{\imath} = i+\delta+1, \ldots, M$, the rest $(\hat{\imath} - i - \delta)$ seats can be provided for one group type $(\hat{\imath} - i - \delta)$ with δ seats of social distancing. Let $D_{\hat{\imath}}^t$ be the random variable that indicates the number of group type $\hat{\imath}$ in t periods. The expected number of accepted people is $i + (\hat{\imath} - i - \delta) P(D_{\hat{\imath}-i-\delta}^{T-t} \geq X_{\hat{\imath}-i-\delta} + 1)$, where $P(D_i^{T-t} \geq X_i)$ is the probability that the demand of group type i' in $(T-t)$ periods is no less than X_i, the remaining supply of group type i. Thus, the term, $P(D_{\hat{\imath}-i-\delta}^{T-t} \geq X_{\hat{\imath}-i-\delta} + 1)$, indicates the probability that the demand of group type $(\hat{\imath} - i - \delta)$ in $(T-t)$ periods is no less than its current remaining supply plus 1.

Similarly, when we retain the supply of group type $\hat{\imath}$ by rejecting the group type i, the expected number of accepted people is $\hat{\imath} P(D_{\hat{\imath}}^{T-t} \geq X_{\hat{\imath}})$. The term, $P(D_{\hat{\imath}}^{T-t} \geq X_{\hat{\imath}})$, indicates the probability that the demand of group type $\hat{\imath}$ in $(T-t)$ periods is no less than its current remaining supply.

Let $d^t(i, \hat{\imath})$ be the difference of the expected number of accepted people between acceptance and rejection in the group type i that occupies seats of $(\hat{\imath} + \delta)$ size in period t. Then we have

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

One intuitive decision is to choose $\hat{\imath}$ with the largest difference. For all $\hat{\imath} = i+1, \ldots, M$, find the largest $d^t(i, \hat{\imath})$, denoted as $d^t(i, \hat{\imath}^*)$. If $d^t(i, \hat{\imath}^*) > 0$, we will plan to assign the group type i in $(\hat{\imath}^* + \delta)$-size seats. Otherwise, reject the group.

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

6.2.2 Tie-breaking Rule for Determining A Specific Row

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

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

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

In the above example, the group type 1 can be assigned to any row with the available seats. The group-type control can help us find the larger group type that can be used to place the arriving group while maximizing the expected values. When there are multiple rows containing the larger group type, we choose the row containing the larger group type according to the tie-breaking rule.

By combining the group-type control strategy with the evaluation of relaxed DP values, we obtain a comprehensive decision-making process within a single period. This integrated approach enables us to make informed decisions regarding the acceptance or rejection of incoming requests, as well as determine the appropriate row for the assignment when acceptance is made.

6.2.3 Regenerate The Seat Planning

To optimize computational efficiency, it is not necessary to regenerate the seat planning for each request. Instead, we can employ a more streamlined approach. Considering that the seats planned for the largest group type can meet the needs of all smaller group types, thus, if the supply for the largest group type diminishes from one to zero, it becomes necessary to regenerate the seat planning.

This avoids rejecting the largest group due to infrequent regenerations. Another situation that requires seat planning regeneration is after we determine whether to assign the arriving group seats planned for the larger group. By regenerating the seat planning in these situations, we can achieve real-time seat assignment while reducing the frequency of planning updates.

The algorithm is shown below.

Algorithm 4: Dynamic Seat Assignment

Obtain $\boldsymbol{X} = [X_1, \ldots, X_M]$, calculate $V^t(l)$, $\forall t = 2, \ldots, T; \forall l = 1, \ldots, L$;
for $t = 1, \ldots, T$ **do**
  Observe group type i;
  if $V^{t+1}(l^t) \leq V^{t+1}(l^t - n_i) + i$ **then**
    if $X_i > 0$ **then**
      Find row k such that $H_{ki} > 0$ according to the tie-breaking rule;
      Accept group type i in row k, update L_k, H_{ki}, X_i;
      if $i = M$ *and* $X_M = 0$ **then**
        Generate seat planning $\boldsymbol{H}$ from Algorithm 2, update the corresponding $\boldsymbol{X}$;
      end
    else
      Calculate $d^t(i, \hat{i}^*)$;
      if $d^t(i, \hat{i}^*) \geq 0$ **then**
        Find row k such that $H_{k\hat{i}^*} > 0$ according to the tie-breaking rule;
        Accept group type i in row k;
        Generate seat planning $\boldsymbol{H}$ from Algorithm 2, update the corresponding $\boldsymbol{X}$;
      else
        Reject group type i;
      end
    end
  else
    Reject group type i;
  end
end

7 Computational Experiment

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

The default setting in the experiments is as follows, $\delta = 1$ and $M = 4$. The number of scenarios in SSP is $|\Omega| = 1000$. The default seat layout consists of 10 rows, each with the same size of 21. Different realistic layouts, group sizes and social distances are also explored. We simulate the arrival of exactly one group in each period, i.e., $p_0 = 0$. The average number of individuals per period, denoted as γ, can be expressed as $\gamma = \sum_{i=1}^{M} ip_i$.

7.1 Performances of Different Policies

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

Parameters Description

We consider three sets of probability distributions: D1: [0.18, 0.7, 0.06, 0.06], D2: [0, 0.5, 0, 0.5], D3: [0.2, 0.8, 0, 0], which are experimented in [4]. We also consider another probability distribution D4: [0.25, 0.3, 0.25, 0.2] for the extra experiments. To assess the effectiveness of different policies across varying demand levels, we conducted experiments spanning a range of 60 to 100 periods.

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

Table 1: Performances of Different Policies

Distribution	T	DSA (%)	DP (%)	Bid (%)	Booking (%)	FCFS (%)
D1	60	100.00	100.00	100.00	88.56	100.00
	70	99.53	99.01	98.98	92.69	98.82
	80	99.38	98.91	98.84	97.06	96.06
	90	99.52	99.23	99.10	98.24	95.37
	100	99.58	99.27	98.95	98.46	94.98
D2	60	99.45	97.23	96.58	99.09	96.49
	70	99.82	98.28	97.46	99.81	95.76
	80	99.92	98.60	98.86	100.00	95.66
	90	99.99	99.10	99.70	100.00	95.66
	100	100.00	98.74	99.99	100.00	95.66
D3	60	100.00	100.00	100.00	93.68	100.00
	70	100.00	100.00	100.00	92.88	100.00
	80	99.54	97.89	97.21	98.98	96.19
	90	99.90	99.73	99.44	99.61	94.53
	100	100.00	100.00	100.00	99.89	94.32
D4	60	99.61	99.68	99.63	94.01	99.61
	70	99.05	98.95	98.49	97.10	96.03
	80	98.96	98.84	98.71	97.99	93.80
	90	99.38	99.17	98.70	98.36	92.87
	100	99.62	99.45	99.13	98.56	92.35

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

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

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

7.2 Impact of Social Distancing

We investigate the impact of social distancing when implementing DSA under varying levels of demand. As an illustrative example, we adopt $D4$ as the probability distribution. The demand levels are varied by adjusting the parameter T from 30 to 90 in increments of 1.

The figure below displays the number of assigned individuals over demand under two different conditions: with social distancing measure and without social distancing measure. For the former case, we employ DSA to obtain the results. In this case, we consider the constraints of social distancing and optimize the seat allocation accordingly. For the latter case, we adopt the optimal solution assuming that all arrivals are known and that there are no social distancing constraints. The figures depicting the results are presented below. The difference between these two figures is the x-axis, the left one is period, while the right one is the percentage of expected demand relative to total seats.

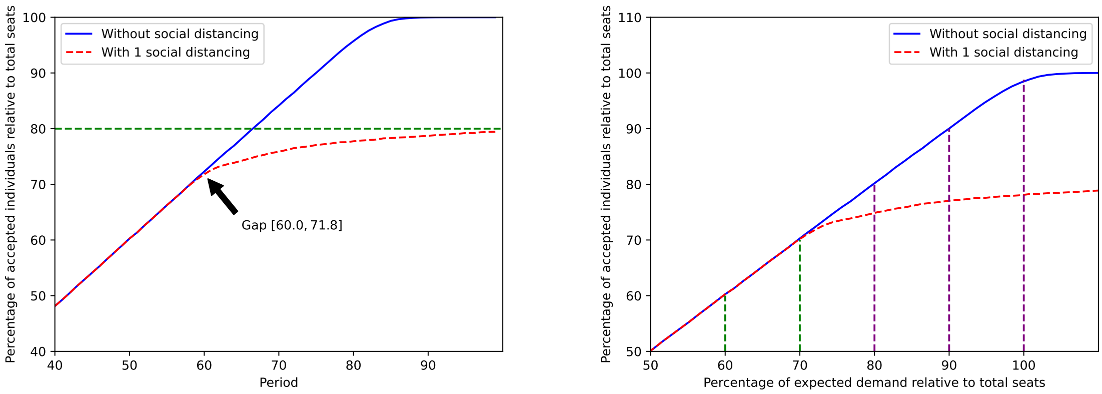
There are three key points in the figure, the gap point, the target occupancy rate and the maximum achievable occupancy rate.

The gap point $\tilde{T}$ is defined as the largest value of T for which the inequality $E(T; DSA) + 1 \geq E(T; OPT)$ holds, where $E(T; DSA)$ denotes the expected number of accepted individuals across 100 instances by DSA with one seat as social distancing, $E(T; OPT)$ denotes the expected number of accepted individuals across 100 instances by the optimal solution when there is no social distancing.

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

The maximum achievable occupancy rate is defined as $\frac{\sum_j \phi(M,L_j)}{\sum_j L_j - N\delta}$, where $\phi(M, L)$ indicate the size of any largest pattern under M and L. According to the definition of the largest pattern in Proposition 2, $\phi(M, L)$ does not decrease in M. Since any largest pattern $\boldsymbol{h}$ under M is a feasible pattern under $(M + 1)$, then, $\phi(M, L) \leq \phi(M + 1, L)$. Therefore, when M increases and L is unchanged, the maximum achievable occupancy rate increases.

Sometimes, the policy also requires a maximum allowable occupancy rate to tighten the social distancing requirement. The maximum allowable rate will be redundant if it is higher than the maximum achievable rate for all events. Only the requirement on occupancy rate is effective and the requirement on social distancing is not effective for an event if it is lower than the target occupancy rate of the event. Furthermore, when the maximum allowable rate is between the target occupancy rate and the maximum achievable rate, the requirements on seat occupancy and social distancing interact to determine the seat assignment.



(a) (b)

Figure 3: The occupancy rate over demand

In the first figure, the gap point is 60, the target occupancy rate is 71.8%. As the expected demand continues to increase, both situations reach their maximum capacity acceptance. For the social distancing situation, when the largest pattern is realized in each row, the maximum achievable occupancy rate is given by $\frac{16}{20} = 80\%$. The second figure is plotted to show that when the expected demand is less than 71.8%, the social distancing measures will not have an impact; when the expected demand is larger than 71.8%, the difference between the number of accepted individuals with and without social distancing measures becomes more pronounced.

7.3 Estimation of Gap Points

To estimate the gap point, we aim to find the maximal period such that all requests can be assigned into the seats during these periods, i.e., for each group type i, we have $\boldsymbol{X}_i = \sum_j x_{ij} \geq d_i$. Meanwhile, we have the capacity constraint $\sum_i n_i x_{ij} \leq L_j$, thus, $\sum_i n_i d_i \leq \sum_i n_i \sum_j x_{ij} \leq \sum_j L_j$. Notice that $E(d_i) = p_i T$, we have $\sum_i n_i p_i T \leq \sum_j L_j$ by taking the expectation. Recall that $\tilde{L} = \sum_j L_j$ denotes the total number of seats, and γ represents the average number of individuals in each period. Then, we can derive the inequality $T \leq \frac{\tilde{L}}{\gamma+\delta}$. Therefore, the upper bound for the expected maximal period is given by $T' = \frac{\tilde{L}}{\gamma+\delta}$.

Assuming that all arrivals within T' periods are accepted and fill all the available seats, the target occupancy rate can be calculated as $\frac{\gamma T'}{(\gamma+\delta)T' - N\delta} = \frac{\gamma}{\gamma+\delta} \frac{\tilde{L}}{\tilde{L} - N\delta}$. However, it is important to note that the actual maximal period will be smaller than T' because it is nearly impossible to accept groups to fill all seats exactly. To estimate the gap point when applying DSA, we can use $y_1 = c_1 \frac{\tilde{L}}{\gamma+\delta}$, where c_1 is a discount factor compared to the ideal assumption. Similarly, we can estimate the target occupancy rate as $y_2 = c_2 \frac{\gamma}{\gamma+\delta} \frac{\tilde{L}}{\tilde{L}-N\delta}$, where c_2 is a discount factor for the occupancy rate compared to the ideal scenario.

To analyze the relation between γ and the gap point, we conducted an analysis using a sample of 200 probability distributions. The figure below illustrates the gap point as a function of γ, along with the corresponding estimations. Additionally, the target occupancy rate is represented by red points. We applied an Ordinary Least Squares (OLS) model to fit the data and estimate the parameter values. The resulting fitted equations, $y_1 = \frac{c_1 \tilde{L}}{\gamma+\delta}$ (represented by the blue line in the figure) and $y_2 = c_2 \frac{\gamma}{\gamma+\delta} \frac{\tilde{L}}{\tilde{L}-N\delta}$ (represented by the orange line in the figure), are displayed in the figure. The goodness of fit is evaluated using R-squared values, which are 1.000 for both models, indicating a perfect fit between the data and the fitted equations. The estimated discount factor values are $c_1 = 0.9578$ and $c_2 = 0.9576$.

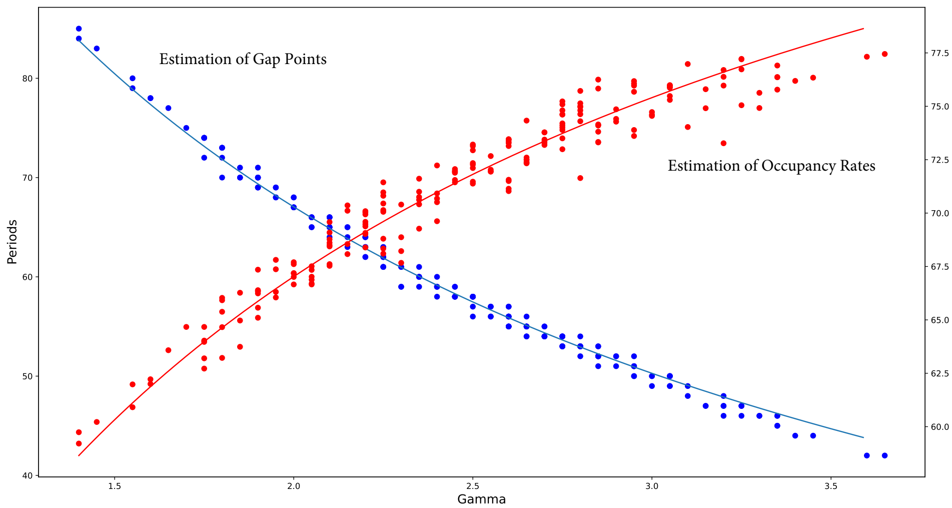


Figure 4: Gap points and their estimation under 200 probabilities

Based on the above analysis, we also explore the results of different layouts, different group sizes and different social distances. Since the figure about the occupancy rate over demand is similar to Figure 3, we only use three metrics to show the results: the gap point and the target occupancy rate, maximal achievable occupancy rate.

Different Layouts

We experiment with several realistic seat layouts selected from the theater seat plan website, https://www.lcsd.gov.hk/en/ticket/seat.html. We select five places, Hong Kong Film Archive Cinema, Kwai Tsing Theatre Transverse Stage, Sai Wan Ho Civic Centre, Sheung Wan Civic Centre, Ngau Chi Wan Civic Centre, represented as HKFAC, KTTTS, SWHCC, SWCC, NCWCC respectively. HKFAC, SWCC, NCWCC, are approximately rectangular layouts, SWHCC is a standard rectangular layout. While KTTTS is an irregular layout.

In these layouts, wheelchair seats and management seats are excluded, while seats with sufficient space for an aisle are treated as new rows.

The occupancy rate over demand follows the typical pattern of Figure 3. The gap point, the target occupancy rate and the maximum achievable occupancy rate are also given in the following table. The maximum achievable occupancy rate can be calculated from Proposition 2.

Table 2: Gap points and target occupancy rates of the layouts

Seat Layout	Gap point	Target occupancy rate
HKFAC	37	70.63 %
KTTTS	40	75.64 %
SWHCC	33	71.31 %
SWCC	45	73.57 %
NCWCC	105	72.05 %

Generally speaking, the length of each row impacts the occupancy rate, as a full pattern can maximize seat utilization, leading to a higher occupancy rate. However, in rectangular layouts, achieving a full pattern in each row can be challenging, resulting in a relatively low occupancy rate, as seen in SWHCC. Although these layouts are all approximately rectangular, the varying lengths of each row lead to different occupancy rates, as demonstrated by HKFAC, SWCC, and NCWCC.

Different Group Sizes

For group sizes of 3, 4 and 5, we present the gap point, the target occupancy rate and the maximum achievable occupancy rate in the table below. The probability distributions for these group sizes all share the same value of $\gamma = 2.4$.

Table 3: Gap points and target occupancy rates of the group sizes

Group Size	Probability distribution	Gap point	Target occupancy rate
3	[0.2, 0.2, 0.6]	60.0	71.83 %
4	[0.25, 0.3, 0.25, 0.2]	60.0	71.79 %
5	[0.3, 0.3, 0.2, 0.1, 0.1]	58.0	70.00 %

Different Social Distances

The following figure illustrates the number of accepted individuals over time with social distancing set at 0, 1, and 2 seats, respectively.

At times, the government may establish a maximum allowable occupancy rate. This rate is effective for an event only if it is lower than the expected occupancy rate. Additionally, the maximum allowed rate becomes redundant if it exceeds the maximum achievable rate for all events.

8 Conclusion

We study the seat planning and seat assignment problem under social distancing requirement. Specifically, we first consider seat planning with deterministic requests. To utilize all seats, we introduce

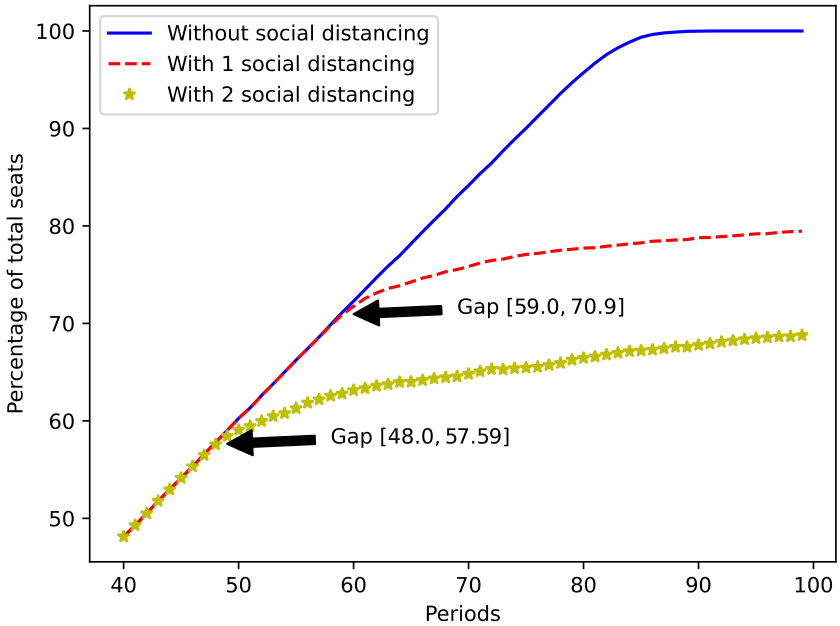


Figure 5: The occupancy rate over demand for different social distancings

the full and largest patterns. Subsequently, we investigate seat planning with stochastic requests. To tackle this problem, we propose a scenario-based stochastic programming model. Then, we utilize the Benders decomposition method to efficiently obtain seat planning, which serves as a reference for dynamic seat assignment. Last but not least, we introduce an approach to address the problem of dynamic seat assignment by integrating relaxed dynamic programming and a group-type control policy.

We conduct several numerical experiments to investigate various aspects of our approach. First, we analyze different policies for dynamic seat assignment. In terms of dynamic seat assignment policies, we consider the classical bid-price control, booking limit control in revenue management, dynamic programming-based heuristics, and the first-come-first-served policy. Comparatively, our proposed policy exhibited superior and consistent performance.

Building upon our policies, we further evaluate the impact of implementing social distancing. By defining the gap point to characterize the situations under which social distancing begins to cause loss to an event, the experiments show that the gap point depends mainly on the mean of the group size. This lead us to estimate the gap point by the mean of the group size.

Our models and analysis are developed for the social distancing requirement on the physical distance and group size, where we can determine an expected occupancy rate for any given event in a venue, and a maximum achievable occupancy rate for all events. Sometimes the government also imposes a maximum allowed occupancy rate to tighten the social distancing requirement. This maximum allowed rate is effective for an event if it is lower than the expected occupancy rate of the event. Furthermore, the maximum allowed rate will be redundant if it is higher than the maximum achievable rate for all events. The above qualitative insights are stable with respect to different parameters in the model, such as the minimum physical distances, the maximum group sizes, and the layout of the venue.

Future research can be pursued in several ways. First, when the seating requests are established, a scattered seat assignment can be examined to maximize the distance between adjacent groups when sufficient seating is available. Second, more flexible scenarios where individuals can select their own seats may be considered. Third, within our framework, individuals arrive at different times to participate in the same event. Research could also examine shared areas where people can come and go at random.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[21] 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.

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

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

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

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

9 Policies for Dynamic Situations

Bid-price Control

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

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

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

In (16), β_j can be interpreted as the bid-price for a seat in row j. A request is only accepted if the revenue it generates is above the sum of the bid prices of the seats it uses. Thus, if its revenue is more than its opportunity costs, i.e., $i - \beta_j n_i \geq 0$, we will accept the group type i. And choose $j^* = \arg\max_j \{i - \beta_j n_i\}$ as the row to allocate that group.

Lemma 2. *The optimal solution to problem* (16) *is given by* $z_1, \ldots, z_{\tilde{i}} = 0$, $z_i = \frac{\delta(n_i - n_{\tilde{i}})}{n_{\tilde{i}}}$ *for* $i = \tilde{i}+1, \ldots, M$ *and* $\beta_j = \frac{n_{\tilde{i}} - \delta}{n_{\tilde{i}}}$ *for all* j.

The bid-price decision can be expressed as $i - \beta_j n_i = i - \frac{n_{\tilde{i}} - \delta}{n_{\tilde{i}}} n_i = \frac{\delta(i - \tilde{i})}{n_{\tilde{i}}}$. When $i < \tilde{i}$, $i - \beta_j n_i < 0$. When $i \geq \tilde{i}$, $i - \beta_j n_i \geq 0$. This means that group type i greater than or equal to $\tilde{i}$ will be accepted if the capacity allows. However, it should be noted that β_j does not vary with j, which means the bid-price control cannot determine the specific row to assign the group to. In practice, groups are often assigned arbitrarily based on availability when the capacity allows, which can result in a large number of empty seats.

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

Algorithm 5: Bid-price Control Algorithm

```
for t = 1, ..., T do
    Observe group type i;
    Solve the LP relaxation of problem (1) with d_i^t = (T - t) · p_i and L^t;
    Obtain ĩ such that the aggregate optimal solution is x e_ĩ + Σ_{i=ĩ+1}^{M} d_i e_i;
    if i ≥ ĩ and max_j L_j^t ≥ n_i then
        Accept the group and assign the group to row k such that L_k^t ≥ n_i;
    else
        Reject the group;
    end
end
```

Booking Limit Control

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

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

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

Algorithm 6: Booking Limit Control Algorithm

for $t = 1, \dots, T$ **do**
  Observe group type i;
  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, $\mathbf{X}$;
  if $X_i > 0$ **then**
    Accept the group and assign the group to row k such that $x_{ik} > 0$;
  else
    Reject the group;
  end
end

First Come First Served (FCFS) Policy

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

Algorithm 7: FCFS Policy Algorithm

for $t = 1, \dots, T$ **do**
  Observe group type i;
  if $\exists k$ *such that* $L_k^t \geq n_i$ **then**
    Accept the group and assign the group to row k;
  else
    Reject the group;
  end
end

Tie-Breaking Rule

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

Proof of Proposition 1

First, we regard this problem as a special case of the Multiple Knapsack Problem (MKP), then we consider the LP relaxation of this problem. Treat the groups as the items, the rows as the knapsacks. There are M types of items, the total number of which is $K = \sum_i d_i$, each item k has a profit p_k and weight w_k. Sort these items according to profit-to-weight ratios $\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \ldots \geq \frac{p_K}{w_K}$. Let the break item b be given by $b = \min\{j : \sum_{k=1}^{j} w_k \geq \tilde{L}\}$, where $\tilde{L} = \sum_{j=1}^{N} L_j$ is the total size of all knapsacks. For the LP relaxation of (1), the Dantzig upper bound [8] is given by $u_{\mathrm{MKP}} = \sum_{j=1}^{b-1} p_j + \left(\tilde{L} - \sum_{j=1}^{b-1} w_j\right) \frac{p_b}{w_b}$. The corresponding optimal solution is to accept the whole items from 1 to $b-1$ and fractional $(\tilde{L} - \sum_{j=1}^{b-1} w_j)$ item b. Suppose the item b belong to type $\tilde{i}$, then for $i < \tilde{i}$, $x_{ij}^* = 0$; for $i > \tilde{i}$, $x_{ij}^* = d_i$; for $i = \tilde{i}$, $\sum_j x_{ij}^* = (\tilde{L} - \sum_{i=\tilde{i}+1}^{M} d_i n_i)/n_{\tilde{i}}$. ■

Proof of Proposition 2

First, we construct a feasible pattern with the size of $qM + \max\{r - \delta, 0\}$, then we prove this pattern is largest. Let $L = n_M \cdot q + r$, where q represents the number of times n_M is selected (the quotient), and r represents the remainder, indicating the number of remaining seats. It holds that $0 \leq r < n_M$. The number of people accommodated in the pattern $\boldsymbol{h}_g$ is given by $|\boldsymbol{h}_g| = qM + \max\{r - \delta, 0\}$. To establish the optimality of $|\boldsymbol{h}_g|$ as the largest number of people accommodated given the constraints of L, δ, and M, we can employ a proof by contradiction.

Assuming the existence of a pattern $\boldsymbol{h}$ such that $|\boldsymbol{h}| > |\boldsymbol{h}_g|$, we can derive the following inequalities:

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

(i) When $r > \delta$, the inequality becomes $q + 1 > \sum_i h_i$. It should be noted that h_i represents the number of group type i in the pattern. Since $\sum_i h_i \leq q$, the maximum number of people that can be accommodated is $qM < qM + r - \delta$.

(ii) When $r \leq \delta$, we have the inequality $q\delta + \delta \geq q\delta + r > \sum_i \delta h_i$. Similarly, we obtain $q + 1 > \sum_i h_i$. Thus, the maximum number of people that can be accommodated is qM, which is not greater than $|\boldsymbol{h}_g|$.

Therefore, $\boldsymbol{h}$ cannot exist. The maximum number of people that can be accommodated in the largest pattern is $qM + \max\{r - \delta, 0\}$. ■

Proof of Proposition 3

First of all, we demonstrate the feasibility of problem (3). Given the feasible seat planning $\boldsymbol{H}$ and $\tilde{d}_i = \sum_{j=1}^{N} H_{ji}$, let $\hat{x}_{ij} = H_{ji}, i \in \mathcal{M}, j \in \mathcal{N}$, then $\{\hat{x}_{ij}\}$ satisfies the first set of constraints. Because $\boldsymbol{H}$ is feasible, $\{\hat{x}_{ij}\}$ satisfies the second set of constraints and integer constraints. Thus, problem (3) always has a feasible solution.

Suppose there exists at least one pattern $\boldsymbol{h}$ is neither full nor largest in the optimal seat planning obtained from problem (3). Let $\beta = L - \sum_i n_i h_i$, and denote the smallest group type in pattern $\boldsymbol{h}$ by k. If $\beta \geq n_1$, we can assign at least n_1 seats to a new group to increase the objective value. Thus, we consider the situation when $\beta < n_1$. If $k = M$, then this pattern is largest. When $k < M$, let $h_k^1 = h_k - 1$ and $h_j^1 = h_j + 1$, where $j = \min\{M, \beta + k\}$. In this way, the constraints will still be satisfied but the objective value will increase when the pattern $\boldsymbol{h}$ changes. Therefore, by contradiction, problem (3) always generate a seat planning composed of full or largest patterns. ■

Proof of Proposition 4

Suppose that H is the seat planning associated with the optimal solution to SSP, but there exists a pattern that is neither full nor the largest. The corresponding excess of supply is $\mathbf{y}^{+}$. According to Proposition 3, H' can be obtained from H. The seat planning, H', is composed of full or largest patterns and satisfies all constraints of SSP. The corresponding excess of supply is $\mathbf{y}^{'+}$.

Then we will demonstrate that for each scenario ω, the objective function of SSP, given by $\sum_{j=1}^{N} \sum_{i=1}^{M} i \cdot x_{ij} - \sum_{i=1}^{M} y_{i\omega}^{+}$, does not decrease when transitioning from H to H'.

Let $\Delta y_{M\omega}^{+} = y_{M\omega}^{'+} - y_{M\omega}^{+}$, $\Delta \sum_{j=1}^{N} x_{Mj} = \sum_{j=1}^{N} x_{Mj}^{'} - \sum_{j=1}^{N} x_{Mj}$. According to (4), when i changes from M to 1, we obtain the following inequalities.

$$
\begin{aligned}
\Delta y_{M\omega}^{+} \geq & \Delta \sum_{j=1}^{N} x_{Mj} \\
\Delta y_{M-1,\omega}^{+} \geq & \Delta y_{M\omega}^{+} + \Delta \sum_{j=1}^{N} x_{M-1,j} \geq \Delta \sum_{j=1}^{N} (x_{Mj} + x_{M-1,j}) \\
\vdots \cdots \geq & \cdots \vdots \\
\Delta y_{1,\omega}^{+} \geq & \Delta \sum_{j=1}^{N} \sum_{i=1}^{M} x_{i,j}
\end{aligned}
$$

Since the objective function does not decrease, $H^{'}$ represents the optimal solution to SSP and is composed of full or largest patterns. ■

Proof of Lemma 1

Note that $\mathbf{f}^{\intercal} = [-\mathbf{1},\ \mathbf{0}]$ and $\mathbf{V} = [\mathbf{W},\ \mathbf{I}]$. Based on this, we can derive the following inequalities: $\boldsymbol{\alpha}^{\intercal}\mathbf{W} \geq -\mathbf{1}$ and $\boldsymbol{\alpha}^{\intercal}\mathbf{I} \geq \mathbf{0}$. According to the expression of $\mathbf{W}$ and $\mathbf{I}$, we can deduce that $0 \leq \alpha_i \leq \alpha_{i-1} + 1$ for $i \in \mathcal{M}$ by letting $\alpha_0 = 0$. These inequalities indicate that the feasible region is nonempty and bounded. For $i \in \mathcal{M}$, α_i is only bounded by $\alpha_{i-1} + 1$ and 0, thus, all extreme points within the feasible region are integral. ■

Proof of Proposition 5

According to the complementary slackness property, we can obtain the following equations

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

When $y^-_{i\omega} > 0$, we have $\alpha_i = 0$. When $y^+_{i\omega} > 0$, we have $\alpha_i = \alpha_{i-1} + 1$. When $y^+_{i\omega} = y^-_{i\omega} = 0$, let $\Delta d = d_\omega - d_0$,

- if $i = M$, $\Delta d_M = 0$, the value of objective function associated with α_M is always 0, thus we have $0 \leq \alpha_M \leq \alpha_{M-1} + 1$;
- if $i < M$, we have $y^+_{i+1,\omega} = \Delta d_i \geq 0$.
  - If $y^+_{i+1,\omega} > 0$, the objective function associated with α_i is $\alpha_i \Delta d_i = \alpha_i y^+_{i+1,\omega}$, thus to minimize the objective value, we have $\alpha_i = 0$.
  - If $y^+_{i+1,\omega} = 0$, we have $0 \leq \alpha_i \leq \alpha_{i-1} + 1$.

■

Proof of Proposition 6

Suppose we have one extreme point $\boldsymbol{\alpha}^0_\omega$ for each scenario. Then we have the following problem.

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

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

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

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

Proof of Lemma 2

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

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