Seat Planning and Seat Assignment with Social Distancing

by

Zikang LI

A Thesis Submitted to

The Hong Kong University of Science and Technology

in Partial Fulfillment of the Requirements for

the Degree of Doctorof Philosophy

in Industrial Engineering & Decision Analytics

July 2024, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Zikang LI

3 July 2024

Seat Planning and Seat Assignment with Social Distancing

by

Zikang LI

This is to certify that I have examined the above PhD thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the Doctorqualifying examination committee have been made.

Xiangtong QI, Thesis Supervisor

Qian LIU, Thesis Co-Supervisor

Jiheng ZHANG, Head of Department

Department of Industrial Engineering & Decision Analytics

3 July 2024

Acknowledgments

I would like to express my sincere appreciation to my research supervisors, Professor Xiangtong Qi and Professor Qian Liu. Throughout my entire research journey at HKUST, they have offered their utmost effort in providing insightful suggestions and unwavering support for my research. Their valuable assistance and thoughtful guidance were essential to the progress of my research and my overall PhD career.

I would also like to acknowledge the support and companionship provided by my colleagues in room 3208 for their caring, helpful and accompanying presence. Additionally, I would like to acknowledge the help provided by the professors and staff in the IEDA department for their meticulous assistance.

Furthermore, I would like to express my gratitude to my university, HKUST, for providing the beautiful scenery, wonderful facilities and valuable resources during my studies.

Finally, I would like to offer my deepest gratitude to my family and the friends I met during my research for your companionship and support.

Table of Contents

List of Figures

List of Tables

Seat Planning and Seat Assignment with Social Distancing

by Zikang LI
Department of Industrial Engineering & Decision Analytics
The Hong Kong University of Science and Technology

Abstract

The pandemic (COVID-19) has necessitated the widespread implementation of social distancing measures to mitigate the spread of the virus. This thesis addresses the complex problem of seat planning and assignment while adhering to social distancing requirements.

Although the seat planning problem has been studied extensively by many researchers, the specific operational details regarding the dynamic seat assignment have not been fully explored or reached. We first discuss the seat planning problem incorporated with social distancing constraints, whereby groups must be seated together while maintaining minimum physical distances.

To handle stochastic demand, we develop a scenario-based stochastic programming approach for generating robust seat planning. We use Benders decomposition to help generate the seat planning efficiently. The obtained seat planning can then be used as the basis when making decisions under the dynamic situation.

For the dynamic seat assignment problem, we consider two scenarios: fixed seat planning, where the seating layout is predetermined, and flexible seat planning, which is the focus of our work. In the flexible case, we propose a dynamic seat assignment policy to handle arriving groups, accepting or denying them immediately based on available seats and distancing requirements.

Our dynamic seat assignment approach outperforms traditional policies, offering a practical tool for implementing social distancing measures while optimizing seat utilization. We provide a detailed analysis of our method's performance and offer insights for policymakers and venue managers on balancing social distancing and seat capacity constraints.

Chapter 1

Introduction

Governments worldwide have been faced with the challenge of reducing the spread of Covid-19 while minimizing the economic impact. Social distancing has been widely implemented as the most effective non-pharmaceutical treatment to reduce the health effects of the virus. This website records a timeline of Covid-19 and the relevant epidemic prevention measures [18]. For instance, in March 2020, the Hong Kong government implemented restrictive measures such as banning indoor and outdoor gatherings of more than four people, requiring restaurants to operate at half capacity. As the epidemic worsened, the government tightened measures by limiting public gatherings to two people per group in July 2020. As the epidemic subsided, the Hong Kong government gradually relaxed social distancing restrictions, allowing public group gatherings of up to four people in September 2020. In October 2020, pubs were allowed to serve up to four people per table, and restaurants could serve up to six people per table. Specifically, the Hong Kong government also implemented different measures in different venues [14]. For example, the catering businesses will have different social distancing requirements depending on their mode of operation for dine-in services. They can operate at 50%, 75%, or 100% of their normal seating capacity at any one time, with a maximum of 2, 2, or 4 people per table, respectively. Bars and pubs may open with a maximum of 6 persons per table and a total number of patrons capped at 75% of their capacity. The restrictions on the number of persons allowed in premises such as cinemas, performance venues, museums, event premises, and religious premises will remain at 85% of their capacity.

The measures implemented by the Hong Kong government primarily concentrate on restricting the size of groups and the occupancy rate of seats. However, implementing these policies in practice can pose challenges, particularly for venues with fixed seating layouts. In order to adhere to social distancing guidelines, it is important to understand the process of generating seat planning based on known groups and how to assign seats to incoming groups. Additionally, it is of interest to explore how the social distancing constraints impact the sellers and the specific policies formulated by the government to address social distancing concerns.

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

To avoid confusion, we clarify the distinction between 'seat planning' and 'seat assignment' which will be used in the following parts. In our context, the seat planning means the seat partition in the planning. It includes two forms, fixed seat planning and flexible seat planning. The former one is that some seats are unavailable, they may be dismantled or disabled by staff beforehand. The latter one represents the current seat planning, but the planning can be altered later when the planned seats don't match with the size of a coming group or when the seat planning is disrupted after assigning a coming group. In the seat assignment, for the coming group, when accepting it, we assign the seats to the group, and the seats will not be used by others in the future.

The following figures illustrate the seat planning and seat assignment.

In order to adhere to social distancing guidelines, it is important to understand the process of generating seat planning based on known groups and how to assign seats to incoming groups. Additionally, it is of interest to explore how the social distancing

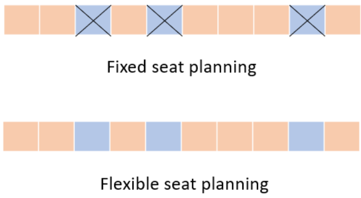


Figure 1.1: Seat Planning

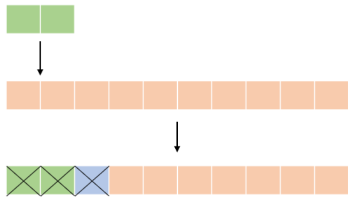


Figure 1.2: Seat Assignment

constraints impact the sellers and the specific policies formulated by the government to address social distancing concerns.

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

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

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

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

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

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

When groups arrive dynamically, we consider the seat assignment under fixed seat planning and flexible seat planning. For the former situation, we develop the group-type control policy, which is also a part of the dynamic seat assignment we designed for the latter situation. We mainly focuses on addressing the dynamic seat assignment problem under flexible seat planning. Solving the problem by dynamic programming can be prohibitive due to the curse of dimensionality, which arises when the problem involves a large number of variables or states. To mitigate this complexity, we begin by generating a seat planning in the stochastic situation. This seat planning acts as a foundation for the seat assignment. When the scale of the problem is small, we can solve the adjusted stochastic programming we developed to make the decision. To avoid situations where the calculation time is too long, we consider the LP relaxation of the stochastic programming when generating the seat planning. Then, we develop the dynamic seat assignment policy, which involves the group-type control policy and comparison the values of relaxed stochastic programming. This dynamic seat assignment policy guides the allocation of seats to the incoming groups sequentially.

In the numerical result, our policy performs well compared with other policies. We use $\tilde{T}$ to denote the gap point, which refers to the first time period at which, on average, the number of people accepted without social distancing is not less than that accepted with social distancing plus one. By sampling many probability combinations, the results show that $\tilde{T}$ and the corresponding occupancy rate, $\beta(\tilde{T})$, can be estimated with γ, the expected number of people per period. Different γ corresponds to different $\tilde{T}$. When the total number of periods, T, is less than $\tilde{T}$, we tend to accept all incoming groups. In this case, there is no difference whether to implement the social distancing restriction. When T is larger, there will be more groups rejected when implementing the social distancing requirement. The government can consider the potential losses when making policies regarding group size and occupancy rate. Similarly, the seller can implement corresponding measures to adhere to these requirements.

Our main contributions for this research are summarized as follows. First, this study presents the first attempt to consider the arrangement of seat assignments with social distancing under dynamic arrivals. While many studies in the literature highlight the importance of social distancing in controlling the spread of the virus, they often focus too much on the model and do not provide much insight into the operational significance behind social distancing [1,11]. Recent studies have explored the effects of social distancing on health and economics, mainly in the context of aircraft [13,27,28]. Our study provides a new perspective to help the government adopt a mechanism for setting seat assignments to implement social distancing during pandemic.

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

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

The rest of this thesis is structured as follows. The following chapter reviews relevant literature. We describe the deterministic problem in Chapter 3. In Chapter 4, we establish the stochastic model,analyze its properties and obtain the feasible seat planning. Chapter 5 demonstrates the seat assignment under the dynamic situation. Chapter 6 gives the numerical results and the insights of implementing social distancing. The conclusions are shown in Chapter 7. In Appendix A, we present the detail about the different policies under flexible seat assignment. Appendix B shows the specific proofs. Appendix C considers the late assignment under flexible seat assignment. Appendix D mentions some details about scenario-based stochastic programming and its relaxation.

Chapter 2

Literature Review

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

2.1 Seat Planning with Social Distancing

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

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

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

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

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

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

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

2.2 Dynamic Seat Assignment

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

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

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

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

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

Chapter 3

Seat Planning with Deterministic Demand

In this section, we consider the deterministic problem by incorporating the social distancing into seat planning. Then, we introduce the concept of the full or largest pattern. For the seat planning that does not utilize all seats, we develop a method to improve the seat planning composed of full or largest patterns.

3.1 Seat Planning with Social Distancing

We incorporate the social distancing in the seat planning problem. Consider a seat layout comprising N rows, with each row containing L_j^0 seats, where $j \in \mathcal{N} := \{1, 2, \ldots, N\}$. The seating arrangement is used to accommodate various groups, where each group consists of no more than M individuals. There are M distinct group types, denoted by group type i, where each group type consists of i people. The set of all group types is denoted by $\mathcal{M} := \{1, 2, \ldots, M\}$. The demand for each group type is represented by a demand vector $\mathbf{d} = (d_1, d_2, \ldots, d_M)^\intercal$, where d_i represents the number of group type i.

In order to comply with the social distancing requirements, individuals from the same group must sit together, while maintaining a distance from other groups. Let δ denote the social distancing, which could entail leaving one or more empty seats. Specifically, each group must ensure the empty seat(s) with the adjacent group(s).

To model the social distancing requirements into the seat planning process, we add the parameter, δ, to the original group types, resulting in the size of group type i being denoted as $n_i = i + \delta$, where $i \in \mathcal{M}$. Accordingly, the size of each row is also adjusted to accommodate the group sizes. Consequently, $L_j = L_j^0 + \delta$ represents the size of row j, where L_j^0 indicates the number of seats in row j. By incorporating the additional seat(s) and designating certain seat(s) for social distancing, we can integrate social distancing measures into the seat planning problem.

We introduce the term pattern to refer to 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 represents 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 represented by $\boldsymbol{H} = \{\boldsymbol{h}_1; \ldots; \boldsymbol{h}_N\}$, where H_{ji} represents the number of group type i in pattern j.

Let $|\boldsymbol{h}|$ indicate the number of people that can be assigned according to pattern $\boldsymbol{h}$, i.e., $|\boldsymbol{h}| = \sum_{i=1}^{M} i h_i$. The size of $\boldsymbol{h}$ provides a measure of the largest number of seats which 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 with respect to 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.*

Figure 3.1: Problem Conversion with One Seat as Social Distancing

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

Definition 1. *Consider a pattern* $\boldsymbol{h} = (h_1, \ldots, h_M)$ *for a row with size* L*. We refer to* $\boldsymbol{h}$ *as a full pattern if* $\sum_{i=1}^{M} n_i h_i = L$*,* $\boldsymbol{h}$ *as a largest pattern if it has a size* $|\boldsymbol{h}|$ *that is greater than or equal to the size* $|\boldsymbol{h}'|$ *of any other feasible pattern* $\boldsymbol{h}'$*.*

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

Proposition 1. *Given the parameters of a row, including its size* L*, the social distancing requirement* δ*, and the number of people in a group allowed* M*, for one possible largest pattern* $\boldsymbol{h}$*, the maximum number of people that can be accommodated is given by* $|\boldsymbol{h}| = qM + \max\{r - \delta, 0\}$*, where* $q = \lfloor \frac{L}{M+\delta} \rfloor$*,* $r \equiv L \bmod (M + \delta)$*.*

Proposition 1 states that if the size of a pattern is $|\boldsymbol{h}| = qM + \max\{r - \delta, 0\}$, then this pattern is a largest pattern. Furthermore, according to the definition of the largest pattern, the size of one possible largest pattern is given by $qM + \max\{r - \delta, 0\}$.

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

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

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

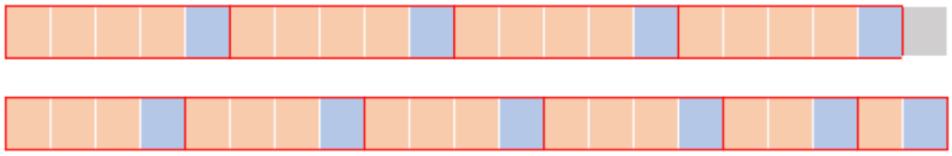


Figure 3.2: Largest and Full Patterns

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 the largest pattern.*

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

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{M} \sum_{j=1}^{N} (n_i - \delta) x_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{N} x_{ij} \leq d_i, \quad i \in \mathcal{M}, \\
& \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N}, \\
& x_{ij} \in \mathbb{N}, \quad i \in \mathcal{M}, j \in \mathcal{N}.
\end{aligned}
\tag{3.1}
$$

This seat planning problem can be regarded as a special case of the multiple knapsack problem. In this context, we define $\boldsymbol{X}$ as the aggregate solution, where $\boldsymbol{X} = (\sum_{j=1}^{N} x_{1j}, \ldots, \sum_{j=1}^{N} x_{Mj})^T$. Each element of $\boldsymbol{X}$, $\boldsymbol{X}_i = \sum_{j=1}^{N} x_{ij}$, represents the available supply for group type i.

In other words, $\boldsymbol{X}$ captures the number of each group type that can be allocated to the seat layout by summing up the supplies across all rows. By considering the monotone ratio between the original group sizes and the adjusted group sizes, we can determine the upper bound of supply corresponding to the optimal solution of the LP relaxation of Problem (3.1), as demonstrated in Proposition 2.

Proposition 2. *For the LP relaxation of problem* (3.1)*, 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 Generate The Seat Planning Composed of Full or Largest Patterns

The seat planning obtained from problem (3.1) may not utilize all available seats, as it depends on the given demand. To improve a given seat planning and utilize all seats, we aim to generate a new seat planning composed of full or largest patterns while ensuring that the original group type requirements are met.

We can assign the maximum number of seats to the possible groups to realize our goal. Specifically, we can reallocate the seats to obtain a new seat planning, $\boldsymbol{H}'$, which can satisfy the realized demand $\sum_{j=1}^{N} H_{ji}$ for each group type i. Mathematically, we can solve the following optimization problem to obtain the desired seat planning:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{M}\sum_{j=1}^{N}(n_i-\delta)x_{ij} \\
s.t. \quad & \sum_{j=1}^{N}\sum_{k=i}^{M} x_{kj} \geq \sum_{k=i}^{M}\sum_{j=1}^{N} H_{ji}, i \in \mathcal{M} \\
& \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N} \\
& x_{ij} \in \mathbb{N}, i \in \mathcal{M}, j \in \mathcal{N}
\end{aligned} \tag{3.2}
$$

Since smaller groups can be satisfied by larger groups, for each group type i, the total quantity from group i to group M should be no less than the total satisfied demand from group type i to group type M, as shown in the first set of constraints in problem (3.2). The optimal solution obtained from problem (3.2) represents the desired seat planning $\boldsymbol{H}'$ and each pattern in $\boldsymbol{H}'$ is either largest or full.

Proposition 3. *Problem* (3.2) *can always generate a seat planning* $\boldsymbol{H}'$ *composed of full or largest patterns, given a feasible seat planning* $\boldsymbol{H}$*.*

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

Chapter 4

Seat Planning with Stochastic Demand

In this section, 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. A particular realization of the demand vector can be represented as $\mathbf{d}_\omega = (d_{1\omega}, d_{2\omega}, \ldots, d_{M,\omega})^\intercal$. Let p_ω denote the probability of any scenario ω, which we assume to be positive. To maximize the expected number of people accommodated over all the scenarios, we propose a scenario-based stochastic programming to obtain a seat planning.

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

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

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

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

Then we have the following formulation:

$$\max \quad E_\omega \left[(n_M - \delta)(\sum_{j=1}^{N} x_{Mj} - y^+_{M\omega}) + \sum_{i=1}^{M-1} (n_i - \delta)(\sum_{j=1}^{N} x_{ij} + y^+_{i+1,\omega} - y^+_{i\omega}) \right] \tag{4.1}$$

$$\text{s.t.} \quad \sum_{j=1}^{N} x_{ij} - y^+_{i\omega} + y^+_{i+1,\omega} + y^-_{i\omega} = d_{i\omega}, \quad i = 1, \ldots, M-1, \omega \in \Omega \tag{4.2}$$

$$\sum_{j=1}^{N} x_{ij} - y^+_{i\omega} + y^-_{i\omega} = d_{i\omega}, \quad i = M, \omega \in \Omega \tag{4.3}$$

$$\sum_{i=1}^{M} n_i x_{ij} \le L_j, j \in \mathcal{N} \tag{4.4}$$

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

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

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

The objective function consists of two parts. The first part represents the number of people in group type M that can be accommodated, given by $(n_M - \delta)(\sum_{j=1}^{N} x_{Mj} - y^+_{M\omega})$. The second part represents the number of people in group type i, excluding M, that can be accommodated, given by $(n_i - \delta)(\sum_{j=1}^{N} x_{ij} + y^+_{i+1,\omega} - y^+_{i\omega}), i = 1, \ldots, M-1$. The overall objective function is subject to an expectation operator denoted by E_ω, which represents the expectation with respect to the scenario set. This implies that the objective function is evaluated by considering the average values of the decision variables and constraints over the different scenarios.

By reformulating the objective function, we have

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

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

In the optimal solution, at most one of $y_{i\omega}^{+}$ and $y_{i\omega}^{-}$ can be positive for any i, ω. Suppose there exist i_0 and ω_0 such that $y_{i_0\omega_0}^{+}$ and $y_{i_0\omega_0}^{-}$ are positive. Substracting $\min\{y_{i_0,\omega_0}^{+}, y_{i_0,\omega_0}^{-}\}$ from these two values will still satisfy constraints (4.2) and (4.3) but increase the objective value when p_{ω_0} is positive. Thus, in the optimal solution, at most one of $y_{i\omega}^{+}$ and $y_{i\omega}^{-}$ can be positive.

Proposition 4. *The deterministic problem* (3.1) *is a special case of stochastic programming when the number of scenarios* $|\Omega|$ *is equal to 1.*

That is to say, the optimal solution to the deterministic problem is one of the optimal solutions to the stochastic programming problem, while the optimal values remain unchanged.

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

Proposition 5 is different from proposition 3. Proposition 3 can make sure that we can obtain a seat planning composed of full or largest patterns. Here, proposition 5 states that there always exists an optimal solution where the corresponding patterns are either full or largest. However, we may not be able to directly find such an optimal solution as described in Proposition 5. Instead, we try to obtain the seat planning composed of full or largest patterns, as stated in Section 4.3.

Then, we reformulate $\mathrm{SSP}(\mathbf{L}, \Omega)$ in a matrix form to apply the Benders decomposition technique. Let $\mathbf{n} = (n_1, \ldots, n_M)^\intercal$ represent the vector of seat sizes for each group type, where n_i denotes the size of seats taken by group type i. Let $\mathbf{L} = (L_1, \ldots, L_N)^\intercal$ represent the vector of row sizes, where L_j denotes the size of row j as defined previously. The constraint (4.4) 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 (4.2) and (4.3), can be expressed in matrix form as:

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

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

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

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

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

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

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal \mathbf{x} + z(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x}^\intercal \mathbf{n} \leq \mathbf{L} \\ & \mathbf{x} \in \mathbb{N}^{M \times N}, \end{aligned} \tag{4.5}$$

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 $\omega \in \Omega$,

$$\begin{aligned} z_\omega(\mathbf{x}) := \max \quad & \mathbf{f}^\intercal \mathbf{y}_\omega \\ \text{s.t.} \quad & \mathbf{V}\mathbf{y}_\omega = \mathbf{d}_\omega - \mathbf{x}\mathbf{1} \\ & \mathbf{y}_\omega \geq 0. \end{aligned} \tag{4.6}$$

We can solve problem (4.5) quickly if we can efficiently solve problem (4.6). Next, we will mention how to solve problem (4.6).

4.2 Solve SSP by Benders Decomposition

We reformulate problem (4.5) into a master problem and a subproblem (4.6). 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 (4.6), then we obtain the solution to the LP relaxation of problem (4.5) by the constraint generation.

4.2.1 Solve The Subproblem

Notice that the feasible region of the dual of problem (4.6) remains unaffected by $\mathbf{x}$. This observation provides insight into the properties of this problem. Let $\boldsymbol{\alpha}$ denote the vector of dual variables. For each ω, we can form its dual problem, which is

$$
\begin{aligned}
\min \quad & \boldsymbol{\alpha}_{\omega}^{\intercal}(\mathbf{d}_{\omega} - \mathbf{x}\mathbf{1}) \\
\text{s.t.} \quad & \boldsymbol{\alpha}_{\omega}^{\intercal}\mathbf{V} \geq \mathbf{f}^{\intercal}
\end{aligned} \tag{4.7}
$$

Lemma 1. *The feasible region of problem* (4.7) *is nonempty and bounded. Furthermore, all the extreme points of the feasible region are integral.*

Let $\mathbb{P}$ indicate the feasible region of problem (4.7). According to Lemma 1, the optimal value of the problem (4.6), $z_{\omega}(\mathbf{x})$, 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 (4.5) and conclude that problem (4.5) can thus be put in the form by setting z_{w} as the variable:

$$
\begin{aligned}
\max \quad & \mathbf{c}^{\intercal}\mathbf{x} + \sum_{\omega \in \Omega} p_{\omega} z_{\omega} \\
\text{s.t.} \quad & \mathbf{x}^{\intercal}\mathbf{n} \leq \mathbf{L} \\
& \boldsymbol{\alpha}_{\omega}^{\intercal}(\mathbf{d}_{\omega} - \mathbf{x}\mathbf{1}) \geq z_{\omega}, \forall \boldsymbol{\alpha}_{\omega} \in \mathcal{O}, \forall \omega \\
& \mathbf{x} \in \mathbb{N}^{M \times N}
\end{aligned} \tag{4.8}
$$

Before applying Benders decomposition to solve problem (4.8), it is important to address the efficient computation of the optimal solution to problem (4.7). When we are given $\mathbf{x}^{*}$, the demand that can be satisfied by the seat planning is $\mathbf{x}^{*}\mathbf{1} = \mathbf{d}_{0} = (d_{1,0}, \ldots, d_{M,0})^{\intercal}$. By plugging them in the subproblem (4.6), we can obtain the value of

$y_{i,\omega}$ recursively:

$$\begin{aligned}
y^-_{M\omega} &= (d_{M\omega} - d_{M0})^+ \\
y^+_{M\omega} &= (d_{M0} - d_{M\omega})^+ \\
y^-_{i\omega} &= \left(d_{i\omega} - d_{i0} - y^+_{i+1,\omega}\right)^+, i = 1, \ldots, M-1 \\
y^+_{i\omega} &= \left(d_{i0} - d_{i\omega} + y^+_{i+1,\omega}\right)^+, i = 1, \ldots, M-1
\end{aligned} \tag{4.9}$$

The optimal solutions to problem (4.7) can be obtained according to the value of $\mathbf{y}_\omega$.

Proposition 6. *The optimal solutions to problem* (4.7) *are given by*

$$\begin{aligned}
\alpha_i = 0 &\quad \text{if } y^-_{i\omega} > 0, i = 1, \ldots, M \text{ or } y^-_{i\omega} = y^+_{i\omega} = 0, y^+_{i+1,\omega} > 0, i = 1, \ldots, M-1 \\
\alpha_i = \alpha_{i-1} + 1 &\quad \text{if } y^+_{i\omega} > 0, i = 1, \ldots, M \\
0 \leq \alpha_i \leq \alpha_{i-1} + 1 &\quad \text{if } y^-_{i\omega} = y^+_{i\omega} = 0, i = M \text{ or } y^-_{i\omega} = y^+_{i\omega} = 0, y^+_{i+1,\omega} = 0, i = 1, \ldots, M-1
\end{aligned} \tag{4.10}$$

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

4.2.2 Constraint Generation

Due to the computational infeasibility of solving problem (4.8) 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 (4.8). This results in a modified problem known as the Restricted Benders Master Problem (RBMP). To find the optimal solution to problem (4.8), we employ the technique of constraint generation. It involves iteratively solving the RBMP and incrementally adding more constraints until the optimal solution to problem (4.8) is obtained.

We can conclude that the RBMP will have the form:

$$
\begin{aligned}
\max \quad & \mathbf{c}^{\intercal}\mathbf{x} + \sum_{\omega \in \Omega} p_{\omega} z_{\omega} \\
\text{s.t.} \quad & \mathbf{x}^{\intercal}\mathbf{n} \leq \mathbf{L} \\
& \boldsymbol{\alpha}_{\omega}^{\intercal}(\mathbf{d}_{\omega} - \mathbf{x}\mathbf{1}) \geq z_{\omega}, \boldsymbol{\alpha}_{\omega} \in \mathcal{O}^{t}, \forall \omega \\
& \mathbf{x} \in \mathbb{N}^{M \times N}
\end{aligned}
\tag{4.11}
$$

Given the initial $\mathcal{O}^t$, we can have the solution $\mathbf{x}^*$ and $\mathbf{z}^* = (z_1^*, \dots, z_{|\Omega|}^*)$. Then $c^{\intercal}\mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega z_\omega^*$ is an upper bound of problem (4.11). When $\mathbf{x}^*$ is given, the optimal solution, $\tilde{\boldsymbol{\alpha}}_\omega$, to problem (4.7) can be obtained according to Proposition 6. 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 (4.11) 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 (4.8).

If for every scenario ω, the optimal value of the corresponding problem (4.7) 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 (4.8). However, if there exists at least one scenario ω for which the optimal value of problem (4.7) 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 7. *RBMP is always bounded with at least one constraint for each scenario.*

From Proposition 7, 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 (4.11) 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 (4.5). Then, we generate an integral seat planning from this solution.

Algorithm 1: Benders Decomposition

Input: Initial problem (4.11) 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 (4.11);
  $UB \leftarrow c^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega z_\omega^*$;
  for $\omega = 1, \ldots, |\Omega|$ **do**
    Obtain $\tilde{\boldsymbol{\alpha}}_\omega$ from Proposition 6;
    $\tilde{z}_\omega = (\tilde{\boldsymbol{\alpha}}_\omega)^\intercal (\mathbf{d}_\omega - \mathbf{x}^* \mathbf{1})$;
    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 (4.11);
    end
  end
  $LB \leftarrow c^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega \tilde{z}_\omega$;
end

4.3 Obtain The Seat Planning Composed of Full or Largest Patterns

We may obtain a fractional optimal solution when we solve the LP relaxation of problem (4.5). This solution represents the optimal allocations of groups to seats but may involve fractional values, indicating partial assignments. Based on the fractional solution obtained, we use the deterministic model to generate a feasible seat planning. The objective of this model is to allocate groups to seats in a way that satisfies the supply requirements for each group without exceeding the corresponding supply values obtained from the fractional solution. To accommodate more groups and optimize seat utilization, we aim to construct a seat planning composed of full or largest patterns based on the feasible seat planning obtained in the last step.

Let the optimal solution to the LP relaxation of problem (4.11) 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 (3.2) with $\boldsymbol{H}$.

The seat planning composed of full or largest patterns can be generated through the following algorithm.

Algorithm 2: Construction of Seat Planning Composed of Full or Largest Patterns

Obtain the optimal solution, $\mathbf{x}^*$, from the LP relaxation of problem (4.11);

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 $\mathrm{SPP}(\{\tilde{X}_i\})$;

Construct the seat planning by problem (3.2) with $\boldsymbol{H}$;

Chapter 5

Seat Assignment with Dynamic Demand

In this section, we discuss how to assign the arriving groups in the dynamic situation. We mainly focus on the real-time seat assignment which is hard to make. Thus, we consider an alternative way to assign seats with the aid of seat planning.

5.1 Real-time Seat Assignment

To model this problem, we adopt a discrete-time framework. Time is divided into T periods, indexed forward from 1 to T. We assume that in each period, at most one group arrives and the probability of an arrival for a group of size i is denoted as p_i, where i belongs to the set $\mathcal{M}$. The probabilities satisfy the constraint $\sum_{i=1}^{M} p_i \leq 1$, indicating that the total probability of any group arriving in a single period does not exceed one. We introduce the probability $p_0 = 1 - \sum_{i=1}^{M} p_i$ to represent the probability of no arrival in a given period t. To simplify the analysis, we assume that the arrivals of different group types are independent and the arrival probabilities remain constant over time. This assumption can be extended to consider dependent arrival probabilities over time if necessary.

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

Upon the arrival of a group type i at time t, the seller needs to make a decision denoted by $u_{i,j}^t$, where $u_{i,j}^t = 1$ indicates acceptance of group type i in row j during period t, while $u_{i,j}^t = 0$ signifies rejection of that group type in row j. The feasible decision set is defined as

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

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

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

$$V^t(\mathbf{L}) = \max_{u_{i,j}^t \in U^t(\mathbf{L})} \left\{ \sum_{i=1}^{M} p_i (\sum_{j=1}^{N} i u_{i,j}^t + V^{t+1}(\mathbf{L} - \sum_{j=1}^{N} n_i u_{i,j}^t \mathbf{e}_j)) + p_0 V^{t+1}(\mathbf{L}) \right\} \tag{5.1}$$

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

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

Solving the dynamic programming problem described in equation (5.1) 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 that consists of the largest or full patterns, as outlined in section 4. This initial seat planning acts as a foundation for our heuristic method. When the supply for group type i is not sufficient, i.e., $X_i = 0$, the seat assignment process involves two main steps. First, we determine the appropriate group type used to accommodated the arriving group. The second step is to choose a specific row according to the group type and make the final decision by evaluating values of stochastic programming. For the whole periods, we regenerate the seat planning under certain conditions to optimize computational efficiency.

5.2 Seat Assignment under Flexible Seat Planning

This situation involves a flexible seat planning approach, where decisions need to be made when a group requests seats. In this case, we dynamically determine the optimal seat planning based on the group's size and the current availability of seats, taking into account social distancing requirements. In a more realistic scenario, groups arrive sequentially over time, and the seller must promptly make group assignments upon each arrival while maintaining the required spacing between groups. When a group is accepted, the seller must also determine which seats should be assigned to that group. It is essential to note that each group must be either accepted in its entirety or rejected entirely; partial acceptance is not permitted. Once the seats are confirmed and assigned to a group, they cannot be changed or reassigned to other groups.

5.2.1 Assignment Based on The Adjusted SSP

Initially, we obtain the seat planning, $x_{ij}, \forall i, j$, by SSP. The seat planning can be obtained by either solving the LP relaxation of SSP or the original SSP, depending on the computational time. Let the corresponding supply be denoted as $[X_1, \ldots, X_M]$, where $X_i = \sum_j x_{ij}$ for all i.

When a group of type i arrives, if $X_i > 0$, we accept the group directly and assign them the seats originally planned for group type i, adhering to any break tie rules that may exist. If $X_i = 0$, we make the decision with the help of the adjusted SSP.

Adjusted SSP

We introduce the decision variables $I_j, j \in \mathcal{N}$ indicating whether we accept the arriving type i' in row j.

$$
\begin{aligned}
\max \quad & \sum_j i' I_j + E_\omega \left[\sum_{i=1}^{M-1} (n_i - \delta)(\sum_{j=1}^{N} x_{ij} + y_{i+1,\omega}^{+} - y_{i\omega}^{+}) + (n_M - \delta)(\sum_{j=1}^{N} x_{Mj} - y_{M\omega}^{+}) \right] \\
\text{s.t.} \quad & \sum_{j=1}^{N} x_{ij} - y_{i\omega}^{+} + y_{i+1,\omega}^{+} + y_{i\omega}^{-} = d_{i\omega}, \quad i = 1, \ldots, M-1, \omega \in \Omega \\
& \sum_{j=1}^{N} x_{ij} - y_{i\omega}^{+} + y_{i\omega}^{-} = d_{i\omega}, \quad i = M, \omega \in \Omega \\
& \sum_{i=1}^{M} n_i x_{ij} \leq L_j - n_{i'} I_j, j \in \mathcal{N} \\
& \sum_{j=1}^{N} I_j \leq 1 \\
& x_{ij} \in \mathbb{N}, \quad i \in \mathcal{M}, j \in \mathcal{N}, y_{i\omega}^{+}, y_{i\omega}^{-} \in \mathbb{N}, \quad i \in \mathcal{M}, \omega \in \Omega, I_j \in \{0,1\}, j \in \mathcal{N}.
\end{aligned}
\tag{5.2}
$$

When $X_i = 0$, we can solve the aforementioned problem to make the decision and update the seat planning accordingly. Once the decision is made, we proceed to assign seats to the next incoming group based on the updated seat planning. This iterative process ensures that the seat assignments are continuously adjusted and optimized as new groups arrive.

5.2.2 Assignment Based on The LP Relaxation of SSP

In a similar manner, denote the corresponding supply as $[X_1, \ldots, X_M]$, where $X_i = \sum_j x_{ij}$ for all i. When a group type i arrives, if $X_i > 0$, we accept the group directly and assign them the seats originally planned for group type i. However, if $X_i = 0$, we determine the group type that can accommodate the arriving group first. We choose a specific row based on the group type and the break-tie rule, and then make the final decision by evaluating the values obtained from the relaxed SSP.

To optimize the computational efficiency, we consider certain conditions to determine when to regenerate the seat planning, ensuring that it is not regenerated unnecessarily in every period. This iterative process continues for the next incoming group, enabling real-time seat assignments based on the current seat planning, while minimizing the frequency of seat planning regeneration. In the following part, we will refer to this policy as dynamic seat assignment.

Determine The Group Type

The group-type control aims to find the group type to assign the arriving group, that helps us narrow down the option of rows for seat assignment. Seat planning serves as a representation of the supply available for each group type. Based on the supply, we can determine whether to accept an incoming group. When a group arrives, if there is sufficient supply available for an arriving group, we will accept the group and choose the group type accordingly. However, if there is no corresponding supply available for the arriving group, we need to decide whether to use a larger group's supply to meet the need of the arriving group. When a group is accepted and assigned to larger-size seats, the remaining empty seat(s) can be reserved for future demand without affecting the rest of the seat planning. To determine whether to use larger seats to accommodate the incoming group, we compare the expected values of accepting the group in the larger seats and rejecting the group based on the current seat planning. Then we identify the possible rows where the incoming group can be assigned based on the group types and seat availability.

Specifically, suppose the supply is $[X_1^t, \dots, X_M^t]$ at period t, the number of remaining periods is $(T-t)$. For the arriving group type i when $X_i^t = 0$, we demonstrate how to decide whether to accept the group to occupy larger-size seats. For any $\hat{i} = i+1, \dots, M$, we can use one supply of group type $\hat{i}$ to accept a group type i. In that case, when $\hat{i} = i+1, \dots, i+\delta$, the expected number of accepted people is i and the remaining seats beyond the accepted group, which is $\hat{i} - i$, will be wasted. When $\hat{i} = i+\delta+1, \dots, M$, the rest $(\hat{i}-i-\delta)$ seats can be provided for one group type $(\hat{i}-i-\delta)$ with δ seats of social distancing. Let $D_{\hat{i}}^{t}$ be the random variable that indicates the number of group type $\hat{i}$ in t periods. The expected number of accepted people is $i+(\hat{i}-i-\delta)P(D_{\hat{i}-i-\delta}^{T-t} \geq X_{\hat{i}-i-\delta}^{t}+1)$, where $P(D_{i}^{T-t} \geq X_{i}^{t})$ is the probability that the demand of group type i in $(T-t)$ periods is no less than X_{i}^{t}, the remaining supply of group type i. Thus, the term, $P(D_{\hat{i}-i-\delta}^{T-t} \geq X_{\hat{i}-i-\delta}^{t}+1)$, indicates the probability that the demand of group type $(\hat{i}-i-\delta)$ in $(T-t)$ periods is no less than its current remaining supply plus 1.

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

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

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

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

Decision on Assigning The Group to A Specific Row

To make the final decision, first, we determine a specific row by the tie-breaking rule, then we assign the group based on the values of relaxed stochastic programming.

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.

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

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

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

Compare The Values of Relaxed SSP

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

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

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

By combining the group-type control strategy with the evaluation of relaxed stochastic programming values, we obtain a comprehensive decision-making process within a single period. This integrated approach enables us to make informed decisions regarding the acceptance or rejection of incoming groups, as well as determine the appropriate row for the assignment when acceptance is made. By considering both computation time savings and potential revenues, we can optimize the overall performance of the seat assignment process.

Regenerate The Seat Planning

To optimize computational efficiency, it is not necessary to regenerate the seat planning for every period. Instead, we can employ a more streamlined approach. Considering that largest group type can meet the needs of all smaller group types, thus, if the supply for the largest group type diminishes from one to zero, it becomes necessary to regenerate the seat planning. This avoids rejecting the largest group due to infrequent regenerations. Another situation that requires seat planning regeneration is when we determine whether to assign the arriving group to a larger group. In such case, we can obtain the corresponding seat planning after solving the relaxed stochastic programmings. By regenerating the seat planning in such situations, we ensure that we have an accurate supply and can give the allocation of seats based on the group-type control and the comparisons of VoA and VoR.

The dynamic seat assignment algorithm is shown below.

Algorithm 3: Dynamic Seat Assignment

for $t = 1, \ldots, T$ **do**
  Observe group type i;
  if $X_i > 0$ **then**
    Find row k such that $H_{ki} > 0$ according to tie-breaking rule;
    Accept group type i in row k, $L_k \leftarrow L_k - n_i$;
    $H_{ki} \leftarrow H_{ki} - 1$, $X_i \leftarrow X_i - 1$; `/* Accept group type` i `when the supply is sufficient */`
    if $i = M$ *and* $X_M = 0$ **then**
      Regenerate $\boldsymbol{H}$ from Algorithm 3;
      Update the corresponding $\boldsymbol{X}$; `/* Regenerate the seat planning when the supply of the largest group type is 0 */`
    end
  else
    Calculate $d^t(i, j^*)$;
    if $d^t(i, j^*) > 0$ **then**
      Find row k such that $H_{kj^*} > 0$ according to tie-breaking rule;
      Calculate the VoA under scenario Ω_A^t and the VoR under scenario Ω_R^t;
      if $VoA > VoR$ **then**
        Accept group type i, $L_k \leftarrow L_k - n_i$;
        Regenerate $\boldsymbol{H}$ from Algorithm 3;
        Update the corresponding $\boldsymbol{X}$;
      else
        Reject group type i;
        Regenerate $\boldsymbol{H}$ from Algorithm 3;
        Update the corresponding $\boldsymbol{X}$;
      end
    else
      Reject group type i;
    end
  end
end

Chapter 6

Results

In this chapter, we analyze our approach in Section 5.2. We carried out several experiments, including analyzing the performances of different policies, evaluating the impact of implementing social distancing. In the experiment, we set the following parameters. The seat layout consists of 10 rows, each with the same size of 21 seats. To account for social distancing measures, one seat is designated as one dummy seat, i.e., $\delta = 1$. The group sizes considered range from 1 to 4 people, i.e., $M = 4$. In our experiments, we simulate the arrival of exactly one group in each period, i.e., $p_0 = 0$. The average number of people per period, denoted as γ, can be expressed as $\gamma = p_1 \cdot 1 + p_2 \cdot 2 + p_3 \cdot 3 + p_4 \cdot 4$, where p_1, p_2, p_3, and p_4 represent the probabilities of groups with one, two, three, and four people, respectively. We assume that p_4 always has a positive value.

6.1 Performances of Different Policies

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

Parameters Description

We give the description of the parameters that will be used in the numerical results. M = 4, $\delta = 1$, $|\Omega| = 1000$, N = 10, $L_j = 21, j \in \mathcal{N}$.

Consider three sets of probability distributions with the same expectation of demand each period: D1: [0.25, 0.25, 0.25, 0.25] D2: [0.25, 0.35, 0.05, 0.35] D3: [0.15, 0.25, 0.55, 0.05]

These distributions were designed to have a common mean γ, ensuring that the expected number of people for each period remained consistent.

Each entry in the result columns is the average of 100 instances

Numerical Results

In order to assess the effectiveness of different policies across different demand levels, we conducted experiments spanning a range of 60 to 100 periods.

The table presents the performance results of five different policies: DSA, DP1, 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 A. Each entry in the table represents the average performance across 100 instances. Performance is evaluated by comparing the ratio of the number of accepted people under each policy to the number of accepted people under the optimal policy, which assumes complete knowledge of all incoming groups before making seat assignments.

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

Table 6.1: Performances of Different Policies

Distribution	T	DSA (%)	DP1 (%)	Bid (%)	Booking (%)	FCFS (%)
D1	60	99.12	98.42	98.38	96.74	98.17
	70	98.34	96.87	96.24	97.18	94.75
	80	98.61	95.69	96.02	98.00	93.18
	90	99.10	96.05	96.41	98.31	92.48
	100	99.58	95.09	96.88	98.70	92.54
D2	60	98.94	98.26	98.25	96.74	98.62
	70	98.05	96.62	96.06	96.90	93.96
	80	98.37	96.01	95.89	97.75	92.88
	90	99.01	96.77	96.62	98.42	92.46
	100	99.23	97.04	97.14	98.67	92.00
D3	60	99.14	98.72	98.74	96.61	98.07
	70	99.30	96.38	96.90	97.88	96.25
	80	99.59	97.75	97.87	98.55	95.81
	90	99.53	98.45	98.69	98.81	95.50
	100	99.47	98.62	98.94	98.90	95.25

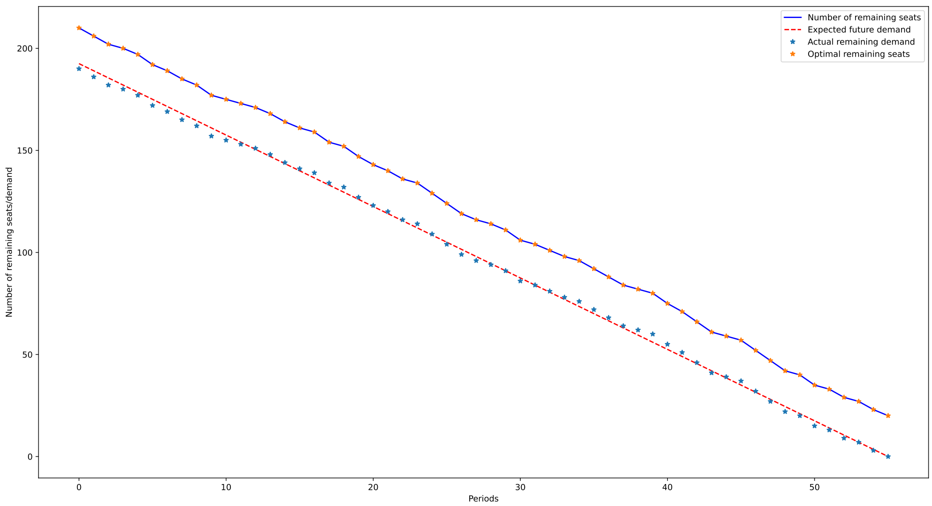
The performance of DSA, DP Base-heuristic, and bid-price policies follows a pattern where it initially decreases and then gradually improves as T increases. When T is small, the demand for capacity is generally low, allowing these policies to achieve relatively optimal performance. However, as T increases, 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. When T increases to infinity, DSA can always generate the largest pattern in each row, thus, the performance will converges to 100% compared to the optimal solution.

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

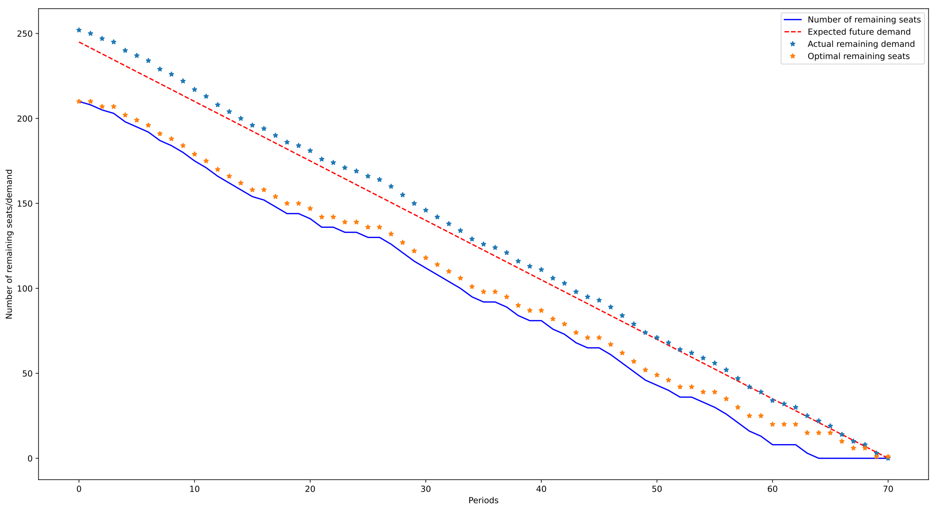
According to the results of policies, we can develop an easy-to-implement policy. When $T \leq \frac{\sum_j L_j}{\sum_i i p_i + 1}$, we accept the groups based on the first come first served policy. When $T > \frac{\sum_j L_j}{\sum_i i p_i + 1}$, we adopt the booking-limit policy, i.e., assign the seats according to the seat planning obtained from problem 3.1.

6.2 Analysis of DSA

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



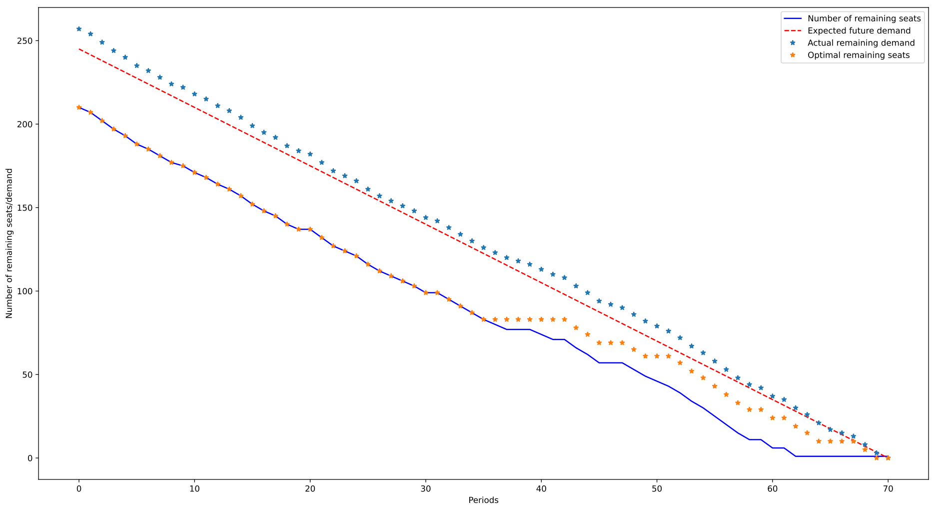
(a) T = 55



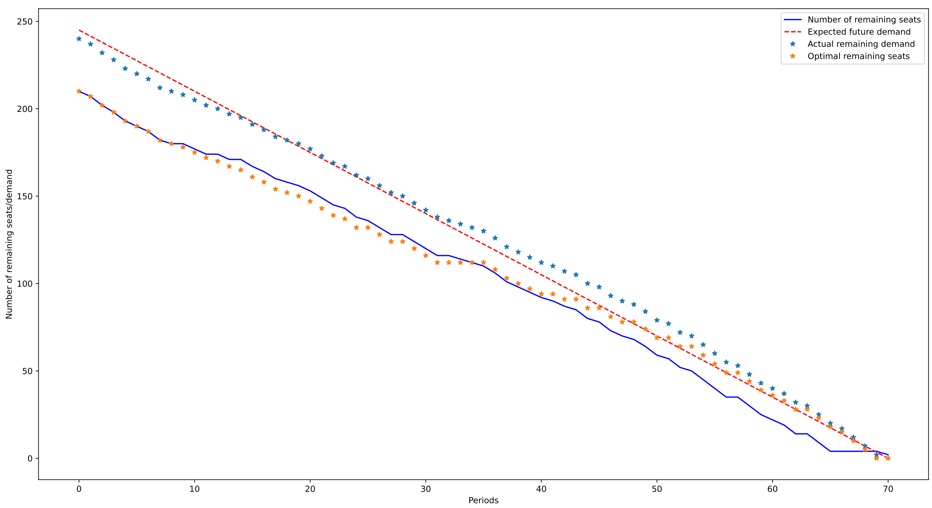
(b) T = 70

Figure 6.1: Arrival Paths

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



(a) T = 55



(b) T = 70

Figure 6.2: Arrival Paths

From the figures, we can obtain several conclusions, first, even at the very beginning of the period, DSA and optimal still reject groups, as the horizontal parts in the line show. Second, DSA is prone to accept the groups earlier compared to the optimal because it makes decision based on the expected demand in the future. Third, when the actual remaining demand is lower than the expected one, the optimal remaining seats will be lower than that in DSA and vice versa.

6.3 Impact of Social Distancing in DSA

In this section, our focus is to analyze the influence of social distancing on the number of accepted individuals. Intuitively, when demand is small, we will accept all arrivals, thus there is no difference whether we implement the social distancing. What is interesting for us is when the difference occurs. Our primary objective is to determine the first period at which, on average, the number of people accepted without social distancing is not less than the number accepted with social distancing plus one. This critical point, referred to as the gap point, denoted by $\tilde{T}$, is of interest to us. Additionally, we will examine the corresponding occupancy rate, $\beta(\tilde{T})$, at this gap point. It should be noted that the difference at a specific period varies depending on the total number of periods considered.

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

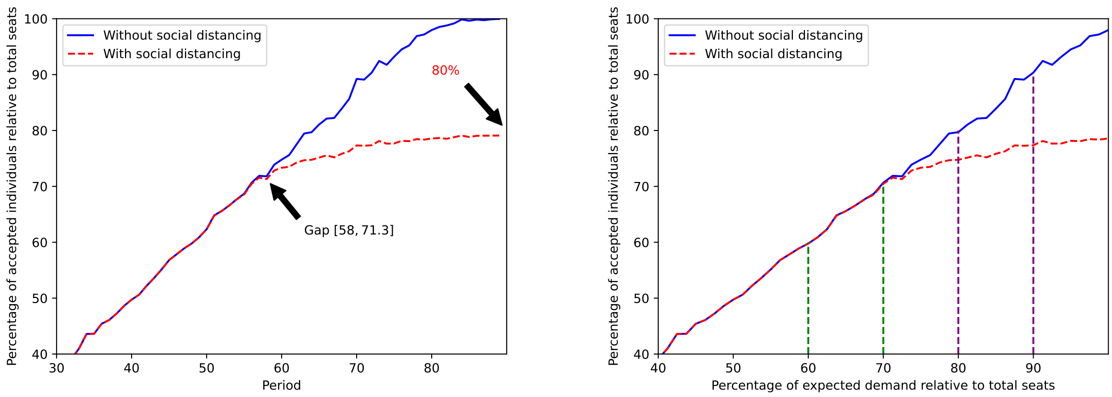
By analyzing and comparing the data, we can gain insights into the relation between demand, social distancing, the number of accepted individuals, and occupancy rates. This information is valuable for understanding the impact of social distancing policies on overall capacity utilization and making informed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

6.3.1 Impact of Social Distancing as The Demand Increases

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

The figure below displays the outcomes of groups who were accepted under two different conditions: with social distancing measures and without social distancing measures.

For the former case, we employ DSA to obtain the results. In this case, we consider the constraints of social distancing and optimize the seat allocation accordingly. For the latter case, we adopt a different approach. We simply accept all incoming groups as long as the capacity allows, without considering the constraints of social distancing. The occupancy rate at different demands is calculated as the mean of these 100 samples. The figures depicting the results are presented below. The difference between these two figures are the x-axis, the left one is period, while the right one is the percentage of expected demand relative to total seats.



(a) (b)

Figure 6.3: The occupancy rate over demand

Under this even probability distribution, we obtain the gap point is 58, the corresponding occupancy rate is 71.3%. To better analyze the situation under different demands, we plot the second figure. When the capacity is sufficient, in this case when the expected demand is less than 71.3%, the outcome remains unaffected by the implementation of social distancing measures. When the expected demand is larger than 71.3%, the difference between the outcomes with and without social distancing measures becomes more pronounced. As the expected demand continues to increase, both situations reach their maximum capacity acceptance. At this point, the gap between the outcomes with and without social distancing measures begins to converge. For the social distancing situation, according to Proposition 1, when the largest pattern is assigned to each row, the resulting occupancy rate is $\frac{16}{20} = 80\%$, which is the upper bound of occupancy rate.

6.3.2 Estimation of Gap Points

To find such a first period, we aim to find the maximum period such that we could assign all the groups during these periods into the seats, i.e., for each group type i, we have $\sum_j x_{ij} \geq d_i$, where x_{ij} is the number of group type i in row j. Meanwhile, we have the capacity constraint $\sum_i n_i x_{ij} \leq L_j$, thus, $\sum_i n_i d_i \leq \sum_i n_i \sum_j x_{ij} \leq \sum_j L_j$. Notice that $E(d_i) = p_i T$, we have $\sum_i n_i p_i T \leq \sum_j L_j$ by taking the expectation. Let $\tilde{L} = \sum_j L_j$, representing the total number of seats, $\gamma = \sum_i i p_i$, representing the average number of people who arrive in each period, we can obtain $T \leq \frac{\tilde{L}}{\gamma+\delta}$, then the upper bound of the expected maximum period is $T' = \frac{\tilde{L}}{\gamma+\delta}$.

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

To analyze the relation between the increment of γ and the gap point, we define each combination (p_1, p_2, p_3, p_4) satisfying $p_1 + p_2 + p_3 + p_4 = 1$ as a probability combination. We conducted an analysis using a sample of 200 probability combinations. The figure below illustrates the gap point as a function of the increment of γ, along with the corresponding estimations. For each probability combination, we considered 100 instances and plotted the gap point as blue points. Additionally, the occupancy rate at the gap point is represented by red points.

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

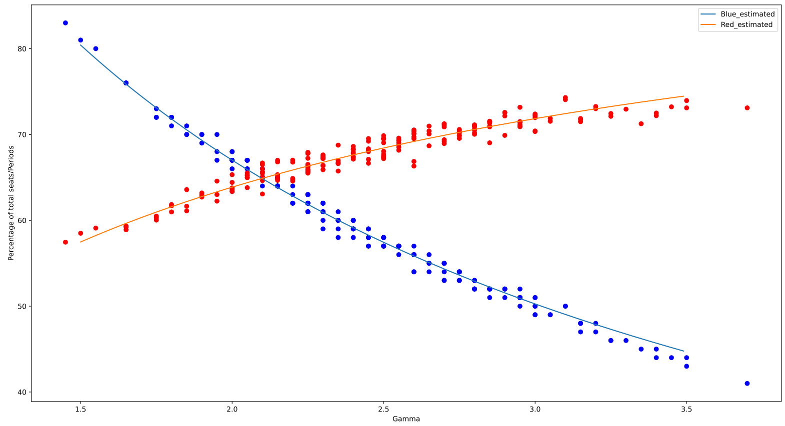


Figure 6.4: Gap points and their estimation under 200 probabilities

The fitting values of c_1 and c_2 can be affected by different seat layouts. To investigate this impact, we conduct several experiments using different seat layouts, specifically with the number of rows $\times$ the number of seats configurations set as 10×16, 10×21, 10×26 and 10×31. Similarly, we perform an analysis using a sample of 100 probability combinations, each with a mean equal to γ. The values of γ range from 1.5 to 3.4. We employed an Ordinary Least Squares (OLS) model to fit the data and derive the parameter values. The goodness of fit is assessed using the R-square values, which are found to be 1.000 for all models, indicating a perfect fit between the data and the models.

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

Table 6.2: Fitting values of c_1 and c_2

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

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

The below table presents the occupancy rate differences for different demand levels (130, 150, 170, 190, 210).

Table 6.3: Gap points and occupancy rate differences under different demands of different gammas

γ	$\tilde{T}$	$\beta(\tilde{T})$	$\Delta\beta(T)$ under different demands				
			130	150	170	190	210
1.9	69	65.52	0.25	5.82	12.82	20.38	24.56
2.1	64	67.74	0.05	4.11	11.51	18.77	21.87
2.3	61	69.79	0	2.29	10.21	17.36	21.16
2.5	57	70.89	0	1.45	9.30	15.78	19.80
2.7	53	71.28	0	1.38	7.39	14.91	19.14

As the value of γ increases, the period of gap point will decrease and the corresponding occupancy rate will increase. This suggests that larger groups are more likely to be accepted, resulting in a smaller number of accepted groups overall. Consequently, the occupancy rate increases due to the allocation of seats to larger groups. The percentage difference is negligible when the demand is small, but it becomes more significant as the demand increases.

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

6.4 Seat Assignment under Fixed Seat Planning

This situation involves a fixed seat planning that is predetermined based on the venue management's requirements. When a group of customers arrives, they can choose seats from the available planning options. The pre-determined seat arrangements ensure that social distancing measures are maintained, even in cases where some individuals may not fully comply with the distancing requirements. This approach allows groups to select seats that best suit their needs, while still adhering to the established seating arrangement designed to enforce social distancing.

The seats, which were arranged for social distancing purposes, need to be dismantled before people arrive to prevent them from occupying those seats. When each group arrives, we make decisions regarding whether to accept or reject them based on the predetermined seat planning. We use the group-type control. Obtain the seat planning from stochastic programming. Suppose the corresponding supply is $[X_1, \ldots, X_M]$. For the arrival of group type i, if $X_i > 0$, accept it directly, assign it the seats planned for group type i; if $X_i = 0$, determine which group type to accept it.

6.4.1 Performances

We consider three probability distributions for the group types: $[0.25, 0.25, 0.25, 0.25]$, $[0.25, 0.35, 0.05, 0.35]$, and $[0.15, 0.25, 0.55, 0.05]$, such that the expected number of people in one period is fixed. The size of each row is 21. FSP in the table refers to the result of seat assignment based on the fixed seat planning, where we use the group-type control to accept or reject the incoming groups. Expected Demand column shows the results when the initial seat planning is obtained from the seat planning problem with the expected demand, shown in the Expected Demand column. Each entry in the FSP and Expected Demand columns represents the average performance across 100 instances. Performance is evaluated by comparing the ratio of the number of accepted people under each approach to the number of accepted people under the optimal policy, which assumes complete knowledge of all incoming groups before making seat assignments.

When the demand is low, FSP approach results in many seats being reserved for the

Distribution	T	# of rows	FSP (%)	Expected Demand (%)
D1	70	10	94.97	94.71
	80		96.48	96.16
	90		97.94	97.36
	100		98.91	96.27
D2	70	10	95.90	95.60
	80		97.06	96.69
	90		98.58	98.58
	100		99.47	95.97
D3	70	10	97.41	96.70
	80		98.85	96.06
	90		98.73	97.63
	100		98.46	98.19
D1	140	20	95.83	95.78
	160		97.46	96.89
	180		99.05	96.42
	200		99.74	97.57

smaller group types. These seats cannot be combined to accommodate a larger group, and thus they remain underutilized. As a result, the performance of FSP compared to the optimal policy is not as good in low demand scenarios. As the demand increases, the performance of FSP is getting better.

6.5 Result of Late Assignment

To demonstrate the impact of immediate seat assignment, we adopt a decision-making process where instant decisions are made but seat assignments are conducted at a later stage. In our numerical experiments, we utilize bid-price control and dynamic programming to evaluate the effectiveness of this approach.

We introduce the following LP,

$$\{\max : \sum_{i=1}^{M} \sum_{j=1}^{N} (n_i - \delta) x_i | \sum_{j=1}^{N} x_i \leq d_i, i \in \mathcal{M}, \sum_{i=1}^{M} n_i x_i \leq L, x_i \geq 0, i \in \mathcal{M}\}, \tag{6.1}$$

where x_i denotes the number of group type i in one row with the size of L.

It is evident that the optimal solution to this LP equals the aggregate optimal solution to LP relaxation of problem (3.1).

For each arriving group, the option to assign seats at a later time is available. Therefore, when deciding to accept a group, it is crucial to ensure that a feasible seat assignment can be made. In practice, a capacity threshold is often set. If the cumulative size of all accepted groups does not exceed the threshold, which can be calculated as $\sum_j [L_j - (n_M - 1)] = L - N(n_M - 1)$, a feasible seat assignment for the current accepted groups always exists. However, if the cumulative size surpasses this threshold, we need to utilize the problem 3.1 to determine whether a feasible seat assignment is possible.

The specific algorithms are shown in the following.

Algorithm 4: Bid-price Control Algorithm with Late Allocation

---

for $t = 1, \ldots, T$ **do**
- Observe group type i;
- Solve the LP relaxation of problem (6.1) with $d_i^t = (T - t) \cdot p_i$ and L^t;
- Obtain $\tilde{i}$ such that the aggregate optimal solution is $xe_{\tilde{i}} + \sum_{i=\tilde{i}+1}^{M} d_i e_i$;
- **if** $i \geq \tilde{i}$ *and there is a feasible assignment* **then**
  - Accept the group ;
  - $L^{t+1} \leftarrow L^t - n_i$;
- **else**
  - Reject the group;
  - $L^{t+1} \leftarrow L^t$;
- **end**

end

The numerical results are presented in the following table. Through a comparison of these values, it becomes evident that late seat assignment leads to an improvement in the occupancy rate, resulting in enhanced performance when compared to optimal seat assignment.

Distribution	T	DP1-A (%)	Bid-A (%)	DP1 (%)	Bid (%)
D1	60	99.52	99.44	98.42	98.38
	70	99.32	98.97	96.87	96.24
	80	99.34	99.30	95.69	96.02
	90	99.55	99.49	96.05	96.41
	100	99.78	99.66	95.09	96.88
D2	60	99.50	99.37	98.26	98.25
	70	99.40	98.97	96.62	96.06
	80	99.46	99.24	96.01	95.89
	90	99.59	99.35	96.77	96.62
	100	99.77	99.61	97.04	97.14
D3	60	99.57	99.54	98.72	98.74
	70	99.46	99.39	96.38	96.90
	80	99.50	99.30	97.75	97.87
	90	99.34	99.44	98.45	98.69
	100	99.34	99.55	98.62	98.94

Chapter 7

Conclusion

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

We conducted several experiments to investigate various aspects of our approach. These experiments include analyzing different policies for dynamic seat assignment and evaluating the impact of implementing social distancing. In terms of dynamic seat assignment policies, we consider the classical bid-price control, booking limit control in revenue management, dynamic programming-based heuristics, and the first-come-first-served policy. Comparatively, our proposed policy exhibited superior performance.

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

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

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Appendix A

Policies under Flexible Seat Assignment

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 (3.1) is:

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

In (A.1), 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* (A.1) *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 (3.1) with d_i^t = (T − t) · p_i and L^t;
    Obtain ĩ such that the aggregate optimal solution is x e_ĩ + ∑_{i=ĩ+1}^{M} d_i e_i;
    if i ≥ ĩ and max_j L_j ≥ n_i then
        Accept the group and assign the group to row k such that L_k ≥ n_i;
        L^{t+1} ← L^t − n_i e_k;
    else
        Reject the group;
        L^{t+1} ← L^t;
    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 (3.1), 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 (3.1) 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 (3.1) 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 $x_{ik} > 0$;
        $\mathbf{L}^{t+1} \leftarrow \mathbf{L}^t - n_i \mathbf{e}_k$;
    else
        Reject the group;
        $\mathbf{L}^{t+1} \leftarrow \mathbf{L}^t$;
    end
end

Dynamic Programming Base-heuristic

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

Let u denote the decision, where $u^t = 1$ if we accept a request in period t, $u^t = 0$ otherwise. Similar to the DP in section 5.2, the DP with one row can be expressed as:

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

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

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

Algorithm 7: Dynamic Programming Base-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 $L_k \geq n_i$;
        $L_k^{t+1} \leftarrow L_k^t - n_i$, $l^{t+1} \leftarrow l^t - n_i$;
    else
        Reject the group;
        $L_k^{t+1} \leftarrow L_k^t$, $l^{t+1} \leftarrow l^t$;
    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 8: FCFS Policy Algorithm

```
for t = 1, ..., T do
    Observe group type i;
    if ∃k such that L_k ≥ n_i then
        Accept the group and assign the group to row k;
        L_k ← L_k − n_i;
    else
        Reject the group;
    end
end
```

Appendix B

Proofs

(Proof of Proposition 1). *First, we construct a feasible pattern with the size of* $qM + \max\{r - \delta, 0\}$*, then we prove this pattern is largest. We can utilize a greedy approach to construct a pattern, denoted as* $\boldsymbol{h}_g$*, by following the steps outlined below. This approach aims to generate a pattern that maximizes the number of people accommodated within the given constraints.*

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

Let $L = n_M \cdot q + r$*, where* q *represents the number of times* n_M *is selected (the quotient), and* r *represents the remainder, indicating the number of remaining seats. It holds that* $0 \leq r < n_M$*.*

The number of people accommodated in the pattern $\boldsymbol{h}_g$ *is given by* $|\boldsymbol{h}_g| = qM + \max\{r - \delta, 0\}$*. To establish the optimality of* $|\boldsymbol{h}_g|$ *as the largest possible number of people accommodated given the constraints of* L*,* δ*, and* M*, we can employ a proof by contradiction.*

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

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

Breaking down the above inequality into two cases:

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

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

Therefore, $\boldsymbol{h}$ cannot exist. All largest patterns can accommodate the same maximum number of people and have the same loss. Hence, the maximum number of people that can be accommodated in the largest pattern is $qM + \max\{r - \delta, 0\}$. Correspondingly, the loss of the largest pattern $|\boldsymbol{h}_g|$ is $q\delta - \delta + \min\{r, \delta\}$. □

(Proof of Proposition 2)**.** *Treat the groups as the items, the rows as the knapsacks. There are M types of items, the total number of which is $K = \sum_i d_i$, each item k has a profit p_k and weight w_k.*

Then this Integer Programming is a special case of the Multiple Knapsack Problem (MKP). Consider the solution to the linear relaxation of (3.1)*. Sort these items according to profit-to-weight ratios $\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \ldots \geq \frac{p_K}{w_K}$. Let the break item b be given by $b = \min\{j : \sum_{k=1}^{j} w_k \geq L\}$, where $L = \sum_{j=1}^{N} L_j$ is the total size of all knapsacks. Then the Dantzig upper bound [8] becomes $u_{\text{MKP}} = \sum_{j=1}^{b-1} p_j + \left(L - \sum_{j=1}^{b-1} w_j\right) \frac{p_b}{w_b}$. The corresponding optimal solution is to accept the whole items from 1 to $b-1$ and fractional $(L - \sum_{j=1}^{b-1} w_j)$ item b.*

Suppose the item b belong to type $\tilde{i}$, then for $i < \tilde{i}$, $x^*_{ij} = 0$; for $i > \tilde{i}$, $x^*_{ij} = d_i$; for $i = \tilde{i}$, $\sum_j x^*_{ij} = (L - \sum_{i=\tilde{i}+1}^{M} d_i n_i)/n_{\tilde{i}}$. □

(Proof of Proposition 4). *When $|\Omega| = 1$ in SSP formulation, the stochastic programming will be*

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

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

(Proof of Proposition 5). *In any optimal solution where one of the corresponding patterns is not full or largest, we have the flexibility to allocate the remaining unoccupied seats. These seats can be assigned to either a new seat planning or added to an existing seat planning. Importantly, since a group can utilize the seat planning of a larger group, the allocation scheme based on the original optimal solution will not affect the optimality of the solution. For each row, there are three situations to allocate the seats. First, when the rest seats can be allocated to the existing groups, then the corresponding pattern becomes a full pattern. Second, when all the existing groups are the largest groups and the rest seats cannot construct a new group, the pattern becomes the largest. Third, when all the existing groups are the largest groups and the rest seats can construct new groups, the rest seats can be used to construct the largest groups until there is no enough capacity, then the pattern becomes the largest. Finally, we can allocate the seats such that each row in the seat planning becomes either full or largest.* □

(Proof of Lemma 1). *Note that* $\mathbf{f}^\intercal = [-\mathbf{1},\ \mathbf{0}]$ *and* $V = [W,\ I]$*. Based on this, we can derive the following inequalities:* $\boldsymbol{\alpha}^\intercal W \geq -\mathbf{1}$ *and* $\boldsymbol{\alpha}^\intercal I \geq \mathbf{0}$*. These inequalities indicate that the feasible region is nonempty and bounded. Moreover, let* $\alpha_0 = 0$*. From this, we can deduce that* $0 \leq \alpha_i \leq \alpha_{i-1} + 1$ *for* $i \in \mathcal{M}$*. Consequently, all extreme points within the feasible region are integral.* □

(Proof of Proposition 6). *According to the complementary slackness property, we can obtain the following equations*

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

When $y^-_{i\omega} > 0$*, we have* $\alpha_i = 0$*; when* $y^+_{i\omega} > 0$*, we have* $\alpha_i = \alpha_{i-1} + 1$*. Let* $\Delta d = d_\omega - d_0$*, then the elements of* Δd *will be a negative integer, positive integer and zero. When* $y^+_{i\omega} = y^-_{i\omega} = 0$*, if* $i = M$*,* $\Delta d_M = 0$*, the value of objective function associated with* α_M *is always* 0*, thus we have* $0 \leq \alpha_M \leq \alpha_{M-1} + 1$*; if* $i < M$*, we have* $y^+_{i+1,\omega} = \Delta d_i \geq 0$*. If* $y^+_{i+1,\omega} > 0$*, the objective function associated with* α_i *is* $\alpha_i \Delta d_i = \alpha_i y^+_{i+1,\omega}$*, thus to minimize the objective value, we have* $\alpha_i = 0$*; if* $y^+_{i+1,\omega} = 0$*, we have* $0 \leq \alpha_i \leq \alpha_{i-1} + 1$*.* □

(Proof of Proposition 7). *Suppose we have one extreme point* $\boldsymbol{\alpha}^0_\omega$ *for each scenario. Then we have the following problem.*

$$
\begin{aligned}
\max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} p_\omega z_\omega \\
s.t. \quad & \mathbf{x}^\intercal \mathbf{n} \leq \mathbf{L} \\
& (\boldsymbol{\alpha}^0_\omega)^\intercal \mathbf{d}_\omega \geq (\boldsymbol{\alpha}^0_\omega)^\intercal \mathbf{x}\mathbf{1} + z_\omega, \forall \omega \\
& \mathbf{x} \in \mathbb{N}^{M \times N}
\end{aligned}
\tag{B.2}
$$

Problem (B.2) *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

$$
\begin{aligned}
\max \quad & \mathbf{c}^{\intercal}\mathbf{x} - \sum_{\omega} p_{\omega} (\boldsymbol{\alpha}_{\omega}^{0})^{\intercal} \mathbf{x} \mathbf{1} + \sum_{\omega} p_{\omega} (\boldsymbol{\alpha}_{\omega}^{0})^{\intercal} \mathbf{d}_{\omega} \\
s.t. \quad & \mathbf{n}\mathbf{x} \leq \mathbf{L} \\
& \mathbf{x} \in \mathbb{N}^{M \times N}
\end{aligned}
\tag{B.3}
$$

Notice that $\mathbf{x}$ *is bounded by* $\mathbf{L}$, *then the problem* (B.2) *is bounded. Adding more constraints will not make the optimal value larger. Thus, RBMP is bounded.* □

(Proof of Proposition 3). *First of all, we demonstrate the feasibility of problem* (3.2). *Given the feasible seat planning* $\boldsymbol{H}$ *and* $\tilde{d}_i = \sum_{j=1}^{N} H_{ji}$, *let* $\hat{x}_{ij} = H_{ji}, i \in \mathcal{M}, j \in \mathcal{N}$, *then* $\{\hat{x}_{ij}\}$ *satisfies the first set of constraints. Because* $\boldsymbol{H}$ *is feasible,* $\{\hat{x}_{ij}\}$ *satisfies the second set of constraints and integer constraints. Thus, problem* (3.2) *always has a feasible solution.*

Suppose there exists at least one pattern $\boldsymbol{h}$ *is neither full nor largest in the optimal seat planning obtained from problem* (3.2). *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* $j = \min\{M, \beta + i\}$. *In this way, the constraints will still be satisfied but the objective value will increase when the pattern* $\boldsymbol{h}$ *changes. Therefore, by contradiction, problem* (3.2) *always generate a seat planning composed of full or largest patterns.* □

(Proof of Lemma 2). *According to the Proposition 2, the aggregate optimal solution to relaxation of problem* (3.1) *takes the form* $xe_h + \sum_{i=h+1}^{M} d_i e_i$, *then according to the complementary slackness property, we know that* $z_1, \ldots, z_h = 0$. *This implies that* $\beta_j \geq \frac{n_i - \delta}{n_i}$ *for* $i = 1, \ldots, h$. *Since* $\frac{n_i - \delta}{n_i}$ *increases with* i, *we have* $\beta_j \geq \frac{n_h - \delta}{n_h}$. *Consequently, we obtain* $z_i \geq n_i - \delta - n_i \frac{n_h - \delta}{n_h} = \frac{\delta(n_i - n_h)}{n_h}$ *for* $i = h+1, \ldots, M$.

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

Appendix C

Relaxed SSP and SSP

C.1 Scenario Generation

When we calculate the relaxed SSP and SSP, we need to generate some scenarios according to the probabilities. In this section, we discuss how to generate the scenarios. An arrival sequence can be expressed as $\{y_1, y_2, \ldots, y_T\}$. Let $N_i = \sum_t I(y_t = i)$, i.e., the number of times group type i arrives during T periods. Then the scenarios, $(N_1, \ldots, N_M)$, follow a multinomial distribution,

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

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

C.2 Running times of solving SSP directly and solving SSP with Benders Decomposition

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

Table C.1: Running times of solving SSP directly and SSP with Benders Decomposition

# of scenarios	Demands	# of rows	# of groups	# of seats	Running time of SSP (s)	Benders (s)
1000	(150, 350)	30	8	(21, 50)	4.12	0.13
5000					28.73	0.29
10000					66.81	0.54
50000					925.17	2.46
1000	(1000, 2000)	200	8	(21, 50)	5.88	0.18
5000					30.0	0.42
10000					64.41	0.62
50000					365.57	2.51
1000	(150, 250)	30	16	(41, 60)	17.15	0.18
5000					105.2	0.37
10000					260.88	0.65
50000					3873.16	2.95

The parameters in the columns of the table are the number of scenarios, the range of demands, running time of SSP, running time of Benders decomposition method, the number of rows, the number of group types and the number of seats for each row, respectively. Take the first experiment as an example, the scenarios of demands are generated from (150, 350) randomly, the number of seats for each row is generated from (21, 50) randomly.

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

C.3 Performance When the Fixed Seat Plannings Are from relaxed SSP and SSP

We compare the performance of seat assignment under fixed seat planning when the fixed seat plannings are from relaxed SSP and SSP, respectively. The meanings of the parameters are the same as what we mentioned in 6.4.1. Each entry of people served is the average of 50 instances. SSP will spend more than 2 hours in some instances, as 'NA' showed in the table. The number of seats is 20 when the number of rows is 8, the number of seats is 40 when the number of rows is 30.

The performance is shown in the following table.

Table C.2: Seat planning from relaxed SSP versus seat planning from SSP

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

As we can see, the two seat planning generation approaches have a close performance when considering the group-type control policy under the fixed seat planning. Given this, we can adopt the relaxed SSP to generate the seat planning, in order to save calculation time, without sacrificing much performance compared to the optimal policy.