Dynamic Seat Assignment with Social Distancing

February 21, 2024

1 Introduction

1.1 Background

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 [2].

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 [1].

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

The measures announced by the Hong Kong government mainly focus on limiting the number of people in each group and the seat occupancy rate. However, implementing these policies in operations can be challenging, especially for venues with fixed seating layouts. In our study, we will focus on addressing this challenge in commercial premises, such as cinemas and music concert venues. We aim to provide a practical tool for venues to optimize seat assignments while ensuring the safety of groups by proposing a seat assignment policy that takes into account social distancing requirements and the given seating layout. We strive to enable venues to implement social distancing measures effectively by offering a solution that provides specific seating arrangements.

1.2 Categories

We can categorize the seating situations into three distinct categories based on the demand and the corresponding decision-making process.

1. Deterministic Situation: In this scenario, 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.

2. Stochastic Situation: In this 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. By considering the expected demand distribution, we can optimize the seat planning to accommodate the maximum number of people while maintaining social distancing.

3. Dynamic Situation: In this scenario, the decision to accept or reject a group is made for each incoming group based on specific requirements. Depending on the situation, we may allocate seats immediately upon arrival or at a later time. This situation is commonly encountered in venues such as cinemas or music concerts, where decisions can be made on a group-by-group basis. The goal is to make timely decisions regarding seat assignments, considering factors such as available seating capacity, social distancing requirements and the specific size of each group.

By classifying the seating situations into these three categories and tailoring the decision-making process accordingly, we can effectively manage seat assignments, optimize seating layouts, and ensure a safe and enjoyable experience for all attendees.

1.3 Concepts and deterministic model

We 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 sizes, resulting in the new size of group type i being denoted as $n_i = i + \delta$, where $i \in \mathcal{M}$. Accordingly, the length of each row is also adjusted to accommodate the adjusted group sizes.

Consequently, $L_j = L_j^0 + \delta$ represents the length of row j, where L_j^0 indicates the number of seats in row j. By incorporating the additional seat(s) and designating certain seat(s) for social distancing, we can integrate social distancing measures into the seat planning problem.

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

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

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

In other words, X captures the number 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 (1), as demonstrated in Proposition 2.

Although the problem size is small and the optimal solution can be easily obtained using a solver, it is still important to analyze the problem further to gain additional insights and understanding. We introduce the term pattern to refer to the seat planning arrangement for a single row. A specific pattern can be represented by a vector $\boldsymbol{h} = (h_1, \ldots, h_M)$, where h_i represents the number of group type i in the row for $i = 1, \ldots, M$. This vector $\boldsymbol{h}$ 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 $\boldsymbol{h} \in \mathbb{Z}_+^M$. 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., $|\boldsymbol{h}| = \sum_{i=1}^{M} i h_i$. We also introduce the concept of loss, which is the number of unoccupied seats. Mathematically, the loss is defined as $L - \delta - |\boldsymbol{h}|$, where L denotes the length of the row. The loss provides a measure of the number of seats which cannot be taken due to the implementation of social distancing constraints. By examining the losses associated with different patterns, we can assess the effectiveness of various seat planning configurations with respect to accommodating the desired number of individuals while adhering to social distancing requirements.

Definition 1. *Given the length of a row, denoted as L, and the maximum size of a group allowed, denoted as M, we can define certain characteristics of a pattern $\boldsymbol{h} = (h_1, \ldots, h_M)$. We refer to a pattern $\boldsymbol{h}$ as a full pattern if it satisfies the condition $\sum_{i=1}^{M} n_i h_i = L$. 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 length of the row L. This ensures that the pattern fully occupies the available row seats.*

Furthermore, we define a pattern $\boldsymbol{h}$ 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 largest pattern is one that either has the maximum size or is equal in size to other patterns, ensuring that it can accommodate the most number of people within the given row length.

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

When there is high demand for seats, it is advantageous to prioritize the largest patterns. These patterns allow for the accommodation of the largest number of individuals due to social distancing requirements. On the other hand, in scenarios with moderate demand, adopting the full pattern becomes more feasible. The full pattern maximizes seating capacity by utilizing all available seats, except those empty seats needed for social distancing measures. By considering both the largest and full patterns, we can optimize seat planning configurations to efficiently accommodate a significant number of individuals while maintaining adherence to social distancing guidelines.

Proposition 1. *Given the parameters of a row, including its length L, the social distancing requirement δ, and the maximum size of 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 $q = \lfloor \frac{L}{M+\delta} \rfloor$, $r \equiv L \bmod (M + \delta)$. The corresponding loss of the largest pattern equals $q\delta - \delta + \min\{r, \delta\}$, represents the amount of empty seats due to the social distancing requirement.*

The largest pattern $\boldsymbol{h}$ is unique and full when $r = 0$, indicating that only one pattern can accommodate the maximal 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 1. *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 loss of the largest pattern can be calculated as $\lfloor \frac{21}{5} \rfloor - 1 + 1 = 4$. The largest patterns are the following:* $(1, 0, 1, 3)$, $(0, 1, 2, 2)$, $(0, 0, 0, 4)$, $(0, 0, 4, 1)$, *and* $(0, 2, 0, 3)$.

Through this example, we observe that the largest pattern does not exclusively consist of large groups but can also include smaller groups. This highlights the importance of considering the various group sizes when using the largest pattern. Another observation relates to the relationship between the largest patterns and full patterns. It is apparent that a full pattern may not necessarily be the largest pattern. For instance, consider the pattern $(1, 1, 4, 0)$, which is a full pattern as it utilizes all available seats. However, its loss value is 6, indicating that it is not the largest pattern. Conversely, a largest pattern may also not necessarily be a full pattern. Take the pattern $(0, 0, 0, 4)$ as an example. It is a largest pattern as it can accommodate the maximum number of individuals. However, it does not satisfy the requirement of fully utilizing all available seats since $4 \times 5 \neq 21$.

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

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

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

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

2 Deterministic Model

Firstly, we consider the deterministic model under social distancing constraints. When we have precise information about the number of people, we can utilize this model to arrange seats accordingly. For instance, during a company meeting where different group members need to sit together, we can determine the venue size based on a fixed number of attendees or accept only a portion of the demand based on the venue's capacity. The objective is to maximize the number of people sitting. The solver can solve this problem quickly with the moderate problem size. However, we find that the results show that most rows will not leave any empty seats. Based on this, we introduce the concepts of the largest pattern and full pattern for each row, and we can always use the largest pattern or full pattern as the optimal solution to meet the demand.

3 Stochastic Situation

Secondly, we consider the stochastic model under social distancing constraints. In certain scenarios, we may have demand data for multiple days, which includes information about the number of people in each group size. Examples of such scenarios could be assembling in a church or seating groups in a cathedral. In these cases, we can utilize the stochastic model to generate a seat planning that ensures social distancing requirements are met. To maintain social distancing effectively, the venue manager needs to enforce a fixed seat planning. It is crucial for the group to adhere to the designated seating arrangement.

scenario-based

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.

Now suppose the demand of groups is stochastic, the stochastic information can be obtained from scenarios through historical data. Use ω to index the different scenarios, each scenario $\omega \in \Omega$. Regarding the nature of the obtained information, we assume that there are $|\Omega|$ possible scenarios. A particular realization of the demand vector can be represented as $\mathbf{d}_\omega = (d_{1\omega}, d_{2\omega}, \ldots, d_{M,\omega})^\intercal$. Let p_ω denote the probability of any scenario ω, which we assume to be positive. To maximize the expected number of people accommodated over all the scenarios, we propose a scenario-based stochastic programming to obtain a seat planning.

The seat planning can be represented by decision variables $\mathbf{x} \in \mathbb{Z}_+^{M \times N}$. Here, x_{ij} 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, $\mathbf{y}^+ \in \mathbb{Z}_+^{M\times|\Omega|}$ and $\mathbf{y}^- \in \mathbb{Z}_+^{M\times|\Omega|}$. Each component of $\mathbf{y}^+$, denoted as $y^+_{i\omega}$, represents the 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, $i \le M-1$,

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

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 formulation of SSP:

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

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

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

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

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

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

The objective function consists of two parts. The first part represents the number of people in the largest group type 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

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

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

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

Proposition 3. *There exists an optimal solution to the stochastic programming problem such that the patterns associated with this optimal solution are composed of the full or largest patterns under any given scenarios.*

Given a specific pattern, we can convert it into a largest or full pattern while ensuring that the original group type requirements are met. When multiple full patterns are possible, our objective is to generate the pattern with minimal loss. Mathematically, for any pattern $\boldsymbol{h} = (h_1, \ldots, h_N)$, we seek to find a pattern $\boldsymbol{h}' = (h_1{}', \ldots, h_N{}')$ that maximizes $|\boldsymbol{h}'|$ while satisfying the following constraints: $h_1{}' \geq h_1$, $h_1{}' + h_2{}' \geq h_1 + h_2$, $\cdots$, $h_1{}' + \ldots + h_N{}' \geq h_1 + \ldots + h_N$. In other words, we want to find a pattern $\boldsymbol{h}'$ where each element h_i' is greater than or equal to the corresponding element h_i in $\boldsymbol{h}$, and the cumulative sums of the elements in $\boldsymbol{h}'$ are greater than or equal to the cumulative sums of the elements in $\boldsymbol{h}$. By finding such a pattern $\boldsymbol{h}'$, we can ensure that the converted pattern meets or exceeds the requirements of the original group types. Among the possible full patterns that satisfy these constraints, we prioritize the one with the smallest loss.

Now, we demonstrate the specific allocation scheme. Let $\beta = L_j - \sum_i n_i x_{ij}$. If row j is not the largest or full, then $\beta > 0$. We aim to allocate the remaining unoccupied seats in row j in a way that maximizes the number of planned groups that become the largest in size. Find the smallest group type in the pattern denoted as k. If $k = M$, it means that this row corresponds to a largest pattern. If $k \neq M$, we reduce the number of group type k by one and increase the number of group type $\min\{(k + \beta), M\}$ by one, the number of unoccupied seats will be reduced correspondingly.

We continue this procedure until either all the planned groups become the largest or $\beta = 0$. If $\beta = 0$, it indicates that the pattern is full. In this case, we have assigned all the unoccupied seats to the existing groups without incurring any additional loss. Therefore, this full pattern has the minimal loss while satisfying the groups requirement. However, if all the planned groups become the largest and $\beta \neq 0$, we can repeatly follow the steps outlined below to obtain the largest pattern:

- If $\beta \geq n_M$, we can assign n_M seats to a new group type M.
- If $n_1 \leq \beta < n_M$, we can assign β seats to a new group type $\beta - n_1 + 1$.
- If $0 < \beta < n_1$, it means that the current pattern is already the largest possible pattern because all the planned groups in the pattern are the largest.

By following these steps and always prioritizing the largest group type for seat planning, we can achieve either the largest pattern or a full pattern with minimal loss. This approach guarantees efficient seat allocation, maximizing the utilization of available seats while still accommodating the original groups' requirements.

To construct the largest or full pattern for each row, we can employ the following algorithm. Since patterns are independent of each other, we can process them row by row within a given seat planning. This enables us to optimize the seat planning by maximizing the utilization of available seats and effectively accommodating the arriving groups.

Algorithm 1: Construct The Largest or Full Pattern

while $\beta > 0$ **do**
    $k \leftarrow \min_i\{h_i \neq 0\}$; `/* Find the smallest group type in the pattern */`
    if $k \neq M$ **then**
        $h_k \leftarrow h_k - 1$;
        $h_{\min\{k+\beta,M\}} \leftarrow h_{\min\{k+\beta,M\}} + 1$;
        $\beta \leftarrow \beta - \max\{1, M-k\}$; `/* Change the current group type to a group type as large as possible */`
    else
        if $\beta \geq n_M$ **then**
            $q \leftarrow \lfloor \frac{\beta}{n_M} \rfloor$;
            $\beta \leftarrow \beta - q n_M$;
            $h_M \leftarrow h_M + q$; `/* Assign seats to as many the largest group type as possible */`
        else
            if $n_1 \leq \beta < n_M$ **then**
                $h_{\beta-n_1+1} \leftarrow h_{\beta-n_1+1} + 1$;
                $\beta \leftarrow 0$;
            else
                $\boldsymbol{h}$ is the largest;
                $\beta \leftarrow 0$;
            end
        end
    end
end

Let $\mathbf{n} = (n_1, \ldots, n_M)$ represent the vector of seat sizes for each group type, where n_i denotes the size of seats taken by group type i. Let $\mathbf{L} = (L_1, \ldots, L_N)$ represent the vector of row lengths, where L_j denotes the length of row j as defined previously. The constraint (5) can be expressed as $\mathbf{nx} \leq \mathbf{L}$. This constraint ensures that the total size of seats occupied by each group type, represented by $\mathbf{nx}$, does not exceed the available row lengths $\mathbf{L}$. 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 (3) and (4), can be expressed in matrix form as:

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

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

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

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

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

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

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal\mathbf{x} + z(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{n}\mathbf{x} \leq \mathbf{L} \\ & \mathbf{x} \in \mathbb{Z}_+^{M \times N}, \end{aligned} \tag{6}$$

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 $\omega \in \Omega$,

$$\begin{aligned} z_\omega(\mathbf{x}) := \max \quad & \mathbf{f}^\intercal\mathbf{y}_\omega \\ \text{s.t.} \quad & \mathbf{V}\mathbf{y}_\omega = \mathbf{d}_\omega - \mathbf{x1} \\ & \mathbf{y}_\omega \geq 0. \end{aligned} \tag{7}$$

We can solve problem (6) quickly if we can efficiently solve problem (7). Next, we will mention how to solve problem (7).

3.1 Solve SSP by Benders Decomposition

We reformulate problem (6) into a master problem and a subproblem (7). 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 (7), then we obtain the solution to the LP relaxation of problem (6) by the constraint generation.

3.1.1 Solve The Subproblem

Notice that the feasible region of the dual of problem (7) 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

$$\begin{aligned} \min \quad & \boldsymbol{\alpha}_\omega^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \\ \text{s.t.} \quad & \boldsymbol{\alpha}_\omega^\intercal \mathbf{V} \geq \mathbf{f}^\intercal \end{aligned} \tag{8}$$

Lemma 1. *Let $\mathbb{P} = \{\boldsymbol{\alpha} \in \mathbb{R}^M | \boldsymbol{\alpha}^\intercal \mathbf{V} \geq \mathbf{f}^\intercal\}$. The feasible region of problem (8), $\mathbb{P}$, is nonempty and bounded. Furthermore, all the extreme points of $\mathbb{P}$ are integral.*

Therefore, the optimal value of the problem (7), $z_\omega(\mathbf{x})$, is finite and can be achieved at extreme points of the set $\mathbb{P}$. Let $\mathcal{O}$ be the set of all extreme points of $\mathbb{P}$. That is, we have $z_\omega(\mathbf{x}) = \min_{\boldsymbol{\alpha}_\omega \in \mathcal{O}} \boldsymbol{\alpha}_\omega^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1})$.

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 (6) and conclude that problem (6) can thus be put in the form by setting z_w as the variable:

$$\begin{aligned} \max \quad & \mathbf{c}^\intercal \mathbf{x} + \sum_{\omega \in \Omega} p_\omega z_\omega \\ \text{s.t.} \quad & \mathbf{n}\mathbf{x} \leq \mathbf{L} \\ & \boldsymbol{\alpha}_\omega^\intercal(\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \geq z_\omega, \forall \boldsymbol{\alpha}_\omega \in \mathcal{O}, \forall \omega \\ & \mathbf{x} \in \mathbb{Z}_+ \end{aligned} \tag{9}$$

Before applying Benders decomposition to solve problem (9), it is important to address the efficient computation of the optimal solution to problem (8). 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 (7), we can obtain the value of $y_{i,\omega}$ recursively:

$$\begin{aligned} y_{M\omega}^- &= (d_{M\omega} - d_{M0})^+ \\ y_{M\omega}^+ &= (d_{M0} - d_{M\omega})^+ \\ y_{i\omega}^- &= \left(d_{i\omega} - d_{i0} - y_{i+1,\omega}^+\right)^+, i = 1, \ldots, M-1 \\ y_{i\omega}^+ &= \left(d_{i0} - d_{i\omega} + y_{i+1,\omega}^+\right)^+, i = 1, \ldots, M-1 \end{aligned} \tag{10}$$

The optimal solutions to problem (8) can be obtained according to the value of $\mathbf{y}_\omega$.

Proposition 4. *The optimal solutions to problem* (8) *are given by*

$$\begin{aligned} \alpha_i &= 0 \quad \text{if } y^-_{i\omega} > 0, i = 1, \ldots, M \text{ or } y^-_{i\omega} = y^+_{i\omega} = 0, y^+_{i+1,\omega} > 0, i = 1, \ldots, M-1 \\ \alpha_i &= \alpha_{i-1} + 1 \quad \text{if } y^+_{i\omega} > 0, i = 1, \ldots, M \\ 0 \le \alpha_i &\le \alpha_{i-1} + 1 \quad \text{if } y^-_{i\omega} = y^+_{i\omega} = 0, i = M \text{ or } y^-_{i\omega} = y^+_{i\omega} = 0, y^+_{i+1,\omega} = 0, i = 1, \ldots, M-1 \end{aligned} \tag{11}$$

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 $\alpha_{M\omega}$.

3.1.2 Constraint Generation

Due to the computational infeasibility of solving problem (9) 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 (9). This results in a modified problem known as the Restricted Benders Master Problem(RBMP). To find the optimal solution to problem (9), we employ the technique of constraint generation. It involves iteratively solving the RBMP and incrementally adding more constraints until the optimal solution to problem (9) is obtained.

We can conclude that the RBMP will have the form:

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

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

Proposition 5. *RBMP is always bounded with at least any one feasible constraint for each scenario.*

Given the initial $\mathcal{O}^t$, we can have the solution $\mathbf{x}^*$ and $\mathbf{z}^* = (z_1^*, \ldots, z_{|\Omega|}^*)$. Then $c^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega z_\omega^*$ is an upper bound of problem (12). When $\mathbf{x}^*$ is given, the optimal solution, $\tilde{\boldsymbol{\alpha}}_\omega$, to problem (8) can be obtained according to Proposition 4. 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 (12) 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 (9).

If for every scenario ω, the optimal value of the corresponding problem (8) 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 (9). However, if there exists at least one scenario ω for which the optimal value of problem (8) is less than z_ω^*, indicating that the constraints are not fully satisfied, we need to add a new constraint $(\tilde{\boldsymbol{\alpha}}_\omega)^\intercal (\mathbf{d}_\omega - \mathbf{x}\mathbf{1}) \ge z_\omega$ to RBMP.

From Proposition 5, 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.

Algorithm 2: Benders Decomposition

Input: Initial problem (12) 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 (12);
  $UB \leftarrow c^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega z_\omega^*$;
  for $\omega = 1, \ldots, |\Omega|$ **do**
    Obtain $\tilde{\boldsymbol{\alpha}}_\omega$ from Proposition 4;
    $\tilde{z}_\omega = (\tilde{\boldsymbol{\alpha}}_\omega)^\intercal (\mathbf{d}_\omega - \mathbf{x}^* \mathbf{1})$;
    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 (12);
    end
  end
  $LB \leftarrow c^\intercal \mathbf{x}^* + \sum_{\omega \in \Omega} p_\omega \tilde{z}_\omega$;
end

However, solving problem (12) 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 (6). Then, we generate an integral seat planning from this solution.

3.2 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 (6). 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 (12) be $\mathbf{x}^*$. Aggregate $\mathbf{x}^*$ to the number of each group type, $X_i^* = \sum_j x_{ij}^*, i \in \mathbf{M}$. Replace the vector $\mathbf{d}$ with X^* in the deterministic model, we have the following problem,

$$\left\{ \max \sum_{j=1}^{N} \sum_{i=1}^{M} (n_i - \delta) x_{ij} : \sum_{i=1}^{M} n_i x_{ij} \leq L_j, j \in \mathcal{N}; \sum_{j=1}^{N} x_{ij} \leq X_i^*, i \in \mathcal{M}; x_{ij} \in Z_+ \right\} \tag{13}$$

Then solve the resulting problem (13) to obtain the optimal solution, $\tilde{\mathbf{x}}$, which represents a feasible seat planning. We can construct the largest or full patterns by Algorithm 1.

Algorithm 3: Seat Planning Construction

Obtain the optimal solution, $\mathbf{x}^*$, from the LP relaxation of problem (12);
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 problem (13) with $\tilde{X}$;
Construct the full or largest patterns by Algorithm 1 with $\tilde{\mathbf{x}}$ and $\boldsymbol{H}$;

3.3 Based on the fixed seat planning

To verify the performance of the seat planning obtained from stochastic programming.

4 Dynamic Situation

Thirdly, we consider the dynamic model under social distancing constraints. In this scenario, we encounter two different situations. The first situation involves a fixed seat planning that is set based on the management's requirements. When a group arrives, they can choose seats from the available planning options. The predetermined seat arrangements ensure that social distancing measures are maintained, and groups can select seats that best suit their needs while adhering to the established seating arrangement. The second 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. By utilizing the dynamic model and considering both fixed and flexible seat planning approaches, we can effectively manage the seating arrangements while adhering to social distancing guidelines.

By considering these different models under social distancing constraints, we can effectively allocate seats and ensure a safe and comfortable environment for all attendees.

4.1 Make the instant allocation

In a more realistic scenario, groups arrive sequentially over time, and the seller must promptly make group assignments upon each arrival while maintaining the required spacing between groups. When a group is accepted, the seller must also determine which seats should be assigned to that group. It is essential to note that each group must be either accepted in its entirety or rejected entirely; partial acceptance is not permitted. Once the seats are confirmed and assigned to a group, they cannot be changed or reassigned to other groups.

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

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

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

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

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

$$V^t(\mathbf{L}) = \max_{u_{i,j}^t \in U^t(\mathbf{L})} \left\{ \sum_{i=1}^{M} p_i (\sum_{j=1}^{N} i u_{i,j}^t + V^{t+1}(\mathbf{L} - \sum_{j=1}^{N} n_i u_{i,j}^t \mathbf{e}_j)) + p_0 V^{t+1}(\mathbf{L}) \right\} \tag{14}$$

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 (14) 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 3. This initial seat planning acts as a foundation for our heuristic method. In section 4, building upon the generated seat planning, we further develop a dynamic seat assignment policy which guides the allocation of seats to the incoming groups sequentially.

4.1.1 Based on the flexible seat planning

4.2 Make the instant decision but late allocation

5 Conclusion

References

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

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

Proof

(Theorem 1). ☐

(Lemma 1). ☐

(Lemma 2). ☐

(Theorem 2). ☐