# New-Normal Cinema Seating Planning

**Lalitha Boddapati, Rebecca Ruiter, Georgiana Juglan, Safia Aarouss, Yasmin van Dijk**
Computing Science (COSC), Utrecht University, The Netherlands

## Abstract

In this paper, we experimented on how to seat people in a cinema, confining to the COVID-19 social distancing norms using offline and online algorithms. We propose several offline and online algorithms for maximising the number of people in a cinema. The offline algorithms are First Fit, Best Fit, Branch and Bound and the online algorithms are Best Fit and First Fit. Depending on the cinema size and available time one of these algorithms for offline problem and online problem can be chosen. We calculated the Time and Memory complexity for the algorithms and the competitive ratio for the online algorithm Best Fit is analysed. Some experiments have been performed to assess the performance of the algorithms.

**2012 ACM Subject Classification** • Algorithms for Decision Support

**Keywords and phrases** Offline algorithm, Exact algorithm, Online algorithm, Competitive Ratio, Branch and Bound, First Fit, Best Fit

## 1 Introduction

We are given a problem of arranging the visitors in a cinema such that the groups are seated in a way that complies with the COVID-19 prevention rules as they are in 2020. As of now, 2020, a selection of rules are in place that dictate what distance must be held between people from different households and how many people may be in a given venue at once; businesses and events have suddenly been bestowed with this optimisation problem, and success is not always guaranteed when the visitor capacity of the venue has suddenly dropped from 'as many seats as there are and then some more' to 'thirty, unless those visitors cannot stay within a meter-and-a-half distance, in which case the capacity becomes entirely dependent on the size of the groups arriving'.

A reservation-based venue can massively profit from a system that can arrange their reservations for them, and companies have in turn profited from that need already [6]. We must however acknowledge that in practice it would not be so easy for companies to cancel group reservations based on the household not fitting in their optimal arrangement - applying our version of the offline algorithm in real-life is likely to lead to a consumer protection case.

The online algorithm has a more practical potential. We feel it is safe to assume that most venues are currently seating people with the greedy method as-is, considering it is an intuitive way to assign groups without much hassle as well as keep track of how many seats are still left. The improvements that we have made to online seat assignment, recalling the Best Fit/First Fit implementation, can be practically applied to get most out of a walk-in venue under the social distancing guidelines, far more than simply sending the new guests to the first spot that can accommodate them.

### 1.1 Literature

Considering that this problem is specific to the current time, there is not much literature on the problem yet, but we have a few pieces of precedent writing to base ourselves on.

A paper from the University of Eindhoven describes a similar problem in a trapezoid-style theatre seating[1]. They describe an integer linear programming solution to their problem,

and find that the most efficient seating arrangement can be found if the groups are divided among 2 shows, letting the program assign the most optimal group sizes to each.

In the 2010 conference Fun with Algorithms, Kranakis and Krizanc have presented, likely unaware of how topical their issue would become in the future, an article on "The Urinal Problem"[4]. They consider an online problem where a man would walk into a restroom and select which urinal would presently give him the most privacy, i.e. try to predict future movements without knowing it. They compare their case to the more well known "unfriendly seating arrangement" by Freedman and Shepp[2]. Their algorithm only considers single visitors, but gives us a valuable hint that maximizing between-group distance is not likely to be the optimal way.

Different solutions to "unfriendly" seating problems put forward a grid-based approach in order to prove an asymptotic limit to their arrangements[3]. However, due to the constraints in our problem we cannot follow a grid-based approach the way these papers have presented them. Due to the wider 'forbidden zone', different group sizes, and more complex theatre arrangement as described in the following paragraph, we have created solutions to the offline problem and the online problem with branch-and-bound and greedy approaches respectively.

## 1.2 Defining the Problem

Specifying seats by $x$ and $y$ for column and row co-ordinates respectively, considering $G$ a set of seating co-ordinates for a group and $G'$ a set of any other seating co-ordinates. Such a group of size $g$ will occupy seats as $(x+i, y) : i = 0, .., g-1$, but subsequently occupy a 'zone' where all other groups are prevented from sitting. If we take a group, the zone surrounding each seat in $G$ becomes unavailable for visitors not of that same group so that the set of the unavailable zone $\mathcal{F}$ becomes:

$$\mathcal{F}_{x,y,g} := G_{x,y,g} + \mathcal{F} = \{(x + x' + i, y + y' = 0, ..., (g-1), (x', y')) \ \in \mathcal{F}\}$$

Where $x'$ and $y'$ represent the unoccupied seats that are forbidden for all visitors not of that same group. We know that those variables are, in relation to the co-ordinate of the single seat, $(x \pm 1, y), (x \pm 2, y), (x, y \pm 1), (x \pm 1, y \pm 1)$, or a blocked zone of 10, which makes the total size of our zone when it is extended to multiple group sizes $2g + 8$ seats.

We try to maximise an arrangement of cinema $C$ by finding an arrangement of groups $G$ that is both correct and of the maximum size. The final cinema arrangement $\mathcal{C}$ is correct if

$$\mathcal{G}_{x,y,g} \cap \mathcal{F}_{x',y',g'} = \emptyset, \forall (x, y) \neq (x', y')$$

We present our solutions to an 'offline' version of this problem, where we handle a set of groups that can be arranged and refused in any possible way in order to find the most optimal seating, and the 'online' version, where we handle the groups in their given order and are not allowed to turn away any or remove any groups that otherwise fit.

$x$ = Occupied, $x$ = Unavailable, 1 = Available

Incorrect:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | $x$ | $x$ | $x$ | $x$ | $x$ | 1 | 1 |
| $x$ | $x$ | $x$ | $x$ | $x$ | $x$ | $x$ | 1 |
| 1 | $x$ | $x$ | $x$ | $x$ | $x$ | 1 | 1 |

Correct:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | $x$ | $x$ | $x$ | $x$ | $x$ | $x$ | 1 |
| $x$ | $x$ | $x$ | $x$ | $x$ | $x$ | $x$ | $x$ |
| 1 | $x$ | $x$ | $x$ | $x$ | $x$ | $x$ | 1 |

## 2 Definitions

We use the concept of a 'greedy' algorithm as defined by the National Institute of Standards and Technology as:

> "An algorithm that always takes the best immediate, or local, solution while finding an answer."[5]

and 'branch and bound' as:

> "An algorithmic technique to find the optimal solution by keeping the best solution found so far. If a partial solution cannot improve on the best, it is abandoned."[5]

## 3 Algorithms

In this section we describe the algorithms we used for the offline and the online problem. In section 3.1 we describe the offline algorithms and the problems we devised. In section 3.2 we describe the online algorithms, the problems we encountered for various instances. This section will be followed by the time and memory complexity analysis of the algorithms.

### 3.1 Offline Algorithms

#### 3.1.1 First Fit

The first and most accessible approach to seat people in the socially-distanced cinema is the First Fit approach. Here, the idea is that the incoming groups are seated in the first found left-most available seats. If a group does not fit anymore then we skip it and move to the next incoming group.

We implemented First Fit in two versions as discussed below:

##### 3.1.1.1 First Fit Big First

Intuitively, the first manner in which we iterated through the coming groups was starting by placing the largest groups first, so trying to maximise the number of people seated by placing as many large groups as possible. However, this approach deemed to not necessarily maximise the total number of people that can be seated. In fact, it acted more like a local maximum for our optimisation problem, rather than guaranteeing reaching a global optimum. Figure 2 illustrates a scenario where such a local maximum is reached: given 1 incoming group of 3 people and 2 incoming groups of 2 people, if we seat the group of 3 people in the middle row, we would not be able to accommodate any other group of 2. However, if we seat the groups of 2 and then we will have seated 4 people instead of 3.

##### 3.1.1.2 First Fit Small First

In order to address this sub-optimal scenario, we complemented this approach by devising a version of this algorithm that seats the smallest groups first, focussing thus on the idea that the number of incoming people would be maximised by seating as many smaller groups as possible.

Although this algorithm addressed the problem of overcoming local maxima of First Fit Small First, it performed worse than the First Fit Big First algorithm, as it will be discussed in the Experiments Section.

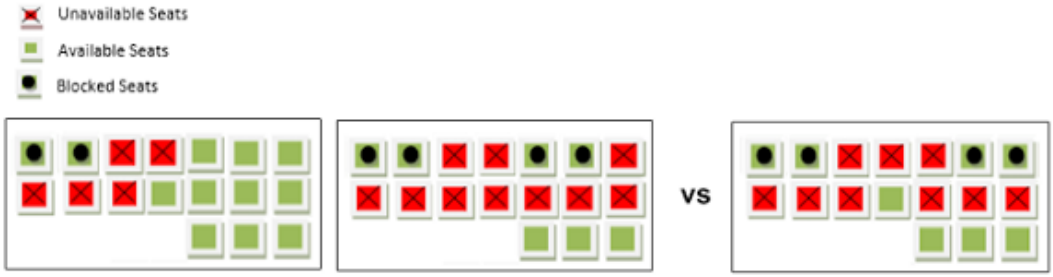



**Figure 1** Example of Suboptimal Placement when Seating First Fit Left-most


**Figure 2** Example of Suboptimal Placement when Seating Larger Groups First

One problem that arose in both versions of First Fit is that when seating people in the first left-most available seats, there can be seating instances that have more unavailable seats created, leaving less seats available for the next incoming groups. Figure 1 illustrates how seating people from the first available left-most seat can create sub-optimal unavailable places: by having already placed a group of 2 in the upper-left corner, and then placing another group of 2 on the first left-most available seats (fifth seat, first row), there are left only 3 seats available. However, placing the incoming group of two on the upper-right corner would create less unavailable seats, leaving room for accommodating 4 people, instead of 3. Hence, seating people in a First Fit left-most manner may not be optimal.

### 3.1.2 Best Fit

The second approach to seat people in the socially-distanced cinema is Best Fit. Instead of seating groups in the left-most first available seats, groups are seated on the first occurrence of the seats that create the least unavailable seats. In this way we minimize the unavailable seats and maximize the number of seats where incoming groups could sit.

Similar as for the First Fit, the Best Fit algorithm has two approaches: the first one is maximizing the number of people seated by seating as many large groups as possible (Best Fit Big First). The second approach is the complemented version where the smallest groups are first placed (Best Fit Small First).

### 3.1.3 Branch and Bound

The third approach to seating people in the socially-distanced cinema is Branch and Bound. In this approach partial solutions are created and evaluated. Partial solutions consist of partially filled cinemas and the remaining groups. The partial solutions are stored in a priority queue in descending order based on the number of people placed. The first partial solution that is added to the priority queue is the empty cinema layout and all the groups that needed to be placed.

The algorithm then keeps taking items from the queue until it is empty or until a timeout is reached. When a partial solution is taken from the queue, the bounding condition is checked first. If the number of people placed plus the number of remaining available seats in the partial solution is less than the number of people placed in the best found solution so far, the partial solution can not improve on the best found solution so far and thus gets discarded.

Otherwise, new partial solutions are created. The first group of horizontally adjacent available seats is found. Then for all remaining groups a new partial solution is created if they fit there, by placing them left most in that group of seats and updating the remaining groups. Another partial solution is created by marking the left most seats as unavailable, so no partial solutions are missed.

Branch and bound is guaranteed to find the optimal solution, although generating and evaluating all partial solutions can take a lot of time for larger cinema sizes, even with the bounding condition. Because the number of partial solutions might become too much to store in memory for larger cinemas, the size of the queue is limited.

## 3.2 Online Algorithms

### 3.2.1 First Fit

The First Fit algorithm is also applied to the online problem. However, due to its online nature, the groups coming in cannot be sorted on group size anymore, so they are placed in the same order as they arrive. Similarly as for the offline problem, we are also facing the same problems regarding suboptimality, as illustrated in Figures 1 and 2: seating people in the left-most seats might create unnecessarily more newly unavailable seats, and seating larger groups first might create a local optimum. Unlike the offline problem, we cannot address the latter issue anymore, due to the inability of choosing the order in which groups are placed. However, the former one is addressed in the next section, through the Best Fit strategy.

### 3.2.2 Best Fit

The Best Fit strategy is also applied to the online problem. The groups are placed in the order in which they arrive, and are placed where they create the least amount of new unavailable spaces. Even though this does not always lead to the optimal solution, it does tend to find better solutions than the First Fit strategy.

## 3.3 Competitive Ratio

For the competitive ratio we assume that the cinemas we use are full blocks of seats and do not contain hallways or gaps. We analyze the lower bound of the Best Fit algorithm, since it has performed the best of all cases.

### 3.3.1 Upper Bound

If we decide to ignore cinemas smaller than $3 \times 8$ (more on those later), we find that the upper bound evens out at a $\frac{2}{3}$ through the following examples:

$x$ = Occupied, $x$ = Unavailable, 1 = Available

$$\begin{matrix} x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \end{matrix}$$

In this example, the most groups you can seat with the ideal visitor distribution is 16 in a space of 24 seats, or $\frac{2}{3}$. Consider varying the size of the cinema in the following ways in order to try to find a higher upper bound:

$$\begin{matrix} x & x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x & x \end{matrix}$$

Initially expanding our field on the x-axis will lead to another column of blocked seats, and we can intuitively understand that this only lowers the total occupation of seats. Expanding to another column leads to a new seat being freed up. Upon placing a single person there we find that $\frac{17}{30}$ seats are occupied, and $\frac{17}{30} < \frac{2}{3}$. Consider an expansion on the y-axis:

$$\begin{matrix} x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \end{matrix}$$

Again we find that one extra row on the y-axis only lowers our bound, whereas the second extra row results in a ratio of $\frac{3}{5}$. $\frac{3}{5} < \frac{2}{3}$, therefore our higher upper bound is not found expanding the cinema this way either.

Combining these approaches we can now infer that a cinema of $5 \times 10$ can accommodate 26 people. This is a ratio of $\frac{26}{50}$, which is also less than $\frac{2}{3}$. When we make the cinema field go to $y = 2$ instead, we find that the maximum occupation for the cinema goes to $\frac{1}{2}$. As we have demonstrated in this section, the maximum occupation of the cinema does not exceed $\frac{2}{3}$, but, interestingly, never falls below $\frac{1}{2}$ either. For the purpose of our competitive ratio calculation we will use $\frac{2}{3}$ as our upper bound.

### 3.3.2 Competitive Ratio Best Fit

For the competitive ratio of Best Fit, we first need to find a lower bound. For its adversary, we will use a situation where only groups of 1 are coming in, since those are the least efficient for space distribution. We can demonstrate the efficiency of the groups when we recall our function for blocked seats and rewrite it as a ratio of occupied seats:

$$\frac{g}{(2(g-1)+10)+g}$$

And consider that group sizes $\{1, 2, 3...8\}$ result in increasing ratios of $\{\frac{1}{11}, \frac{1}{7}, \frac{3}{17}...\frac{1}{4}\}$. Since our algorithm will attempt to place the groups in the locations where they create the least

amount of new unavailable spaces, the occupation of single-person groups on a $3 \times 8$ When we apply these inefficient groups to our upper bound example:

$$\begin{array}{cccccccc} x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \end{array}$$

Giving us a lower bound of $\frac{1}{4}$. Therefore, our competitive ratio is

$$\frac{1/4}{2/3} = \frac{3}{8}$$

### 3.3.3 Density Function

In order to calculate the upper bound, we will do so by focussing on the concept of density: the maximal number of people that can be seated in a square area of $k \times k$ seats [1]. In this way, we can calculate the density $d_t$ for each group of $t$ people. Intuitively,

$$d_t \leq \frac{tn_t}{k^2}$$

where $n_t$ is the number of groups of size $t$. This formula tells us that in an area of $k \times k$ we can seat at most all groups of size $t$.

Although this affirmation is correct, it is not a tight bound. Therefore, we will try to further bound down the value of $n_t$, based on the seating restrictions that are created around a seated group: whenever we seat a group of $t$ people, there are at most $2t + 3$ newly created unavailable spaces. This number is lower if a person is seated on marginal rows or columns, so in order to stay as closely as possible to this $2t + 3$ value, we would need to extend our area of interest from $k \times k$ to $(k + 1)(k + 2)$. This is due to the fact that, for instance, when seating someone in the left-most upper corner, seats in one row above the and 2 columns to the left of it will also be counted in the $2t + 3$ newly unavailable seats.

Hence, we can find a new tighter upper bound for the number of groups that can be seated in an area of size $k \times k$: the number of total seats available (extended area included) divided by the number of unavailable seats created:

$$n_t \leq \frac{(k + 1)(k + 2)}{2t + 3}$$

Plugging this in into the density equation, we are left with a tighter upper bound on the density

$$d_t \leq \frac{t}{k^2} * \frac{(k + 1)(k + 2)}{2t + 3} \leq \frac{tn_t}{k^2}$$

Lastly, we calculate the competitive ratio $\frac{ALG}{OPT}$, where $ALG$ is our adversarial for our Best Fit Online Algorithm. Given a cinema with at least 8 columns, take any square seating area of $k \times k, k \geq 8$. Suppose that the coming in groups are only groups of 1 and 8 people. In the optimal, best-case scenario, $OPT$ will seat all groups of 8 first, computing a density of $d_8$ whereas if the groups of 1 are coming in first, $ALG$ will try to accommodate them, leaving no space for the groups of 8, thus computing at most $d_1$. In this case, the competitive ratio $\frac{ALG}{OPT}$ will be equivalent to the density fraction.

$$\frac{d_1}{d_8} \leq \frac{\frac{1*(k+1)(k+2)}{2*1+3}}{\frac{8*(k+1)(k+2)}{2*8+3}} = \frac{(k + 1)(k + 2)}{5} * \frac{19}{8(k + 1)(k + 2)} = \frac{19}{40} = 0.475$$

Using the Density Function, we can discuss the upper bound of the competitive ratio of our algorithm being:

$$\frac{ALG}{OPT} \geq \frac{19}{40}$$

If we apply this upper bound to our Best Fit lower bound, we find a competitive ratio of:

$$\frac{1/4}{19/40}$$

Therefore, we can claim that the competitive ratio of our Best Fit algorithm is in the range of $\frac{3}{8}$ and $\frac{10}{19}$.

# 4 Time and Memory Complexity

## 4.1 First Fit

This method takes $O(N + DG)$ where $N$ is the size of the cinema, $D$ represents the size of the list of available seats on each row and $G$ the number of groups that are coming in. The creation of the list is done in $O(N)$. This list is iterated through sequentially for each group, which takes $O(DG)$ time (once per group). Because the groups are seated in the first available seats, most of the time the list will not be fully iterated, hence the upper bound. Based on the previously discussed data structures, the memory complexity is $O(N + D + G)$.

## 4.2 Best Fit

Similarly to First Fit, Best Fit is also implemented by keeping a list of available seats of each row, resulting in a $\Theta(N + DG)$, $N$ being the size of the cinema, $D$ representing the size of the list of available seats on each row, and $G$ the number of groups that are coming in. The creation of the list is also done in $O(N)$. However, in this case, the list is completely iterated through sequentially for each group, resulting in an exact bound. The memory complexity lies at $\Theta(N + D + G)$.

## 4.3 Branch and Bound

The branch and bound method creates all the possible partial solutions for each seating possibility. If we reduce this problem to permutations, the time complexity becomes $O(N!)$. As far as the memory complexity is concerned, each solution is comprised of a cinema of size $N$, which raises the memory used to $O(N \times N!)$.

# 5 Experiments

We devised a number of experiments in order to assess the performance of our algorithms as follows: given the provided dataset from the course, for both the offline and online problems, we ran each of our algorithms, and analysed their performance.

## 5.1 The Dataset

The dataset used for analysis is comprised of 21 cinema instances for the offline testing, and 18 instances for the online testing. The inputs were sorted increasingly from small cinemas ($4 \times 4$), to large ones ($1000 \times 1000$). We noticed that the number of seats and the number of

coming people was somewhat proportional, so there were no cases in which there would be 1000 people coming, but only 5 seats available, for example. This aspect allowed us to draw more robust conclusions about the performance of our algorithms based on the increasing cinema sizes. Moreover, for each instance, we calculated the cinema size, number of seats, and number of coming people, in order to later use them for statistical purposes.

## 5.2 Setup

The algorithms were implemented in C# and run on a Windows 10 machine, with a quad-core Intel Core i5-7400 3.00 GHz processor and 8GB of RAM.

Each offline and online algorithm was run against its corresponding test instances, and the number of seated people, and the run-times were saved in CSV files. We had a 30-minute (180000ms) time-limit to account for the high running time of some of our algorithms. Once this limit is reached, we output the best found solution found so far. The statistical results were analysed using Python with the Pandas and Matplotlib libraries. All of them can be found in the Appendix.

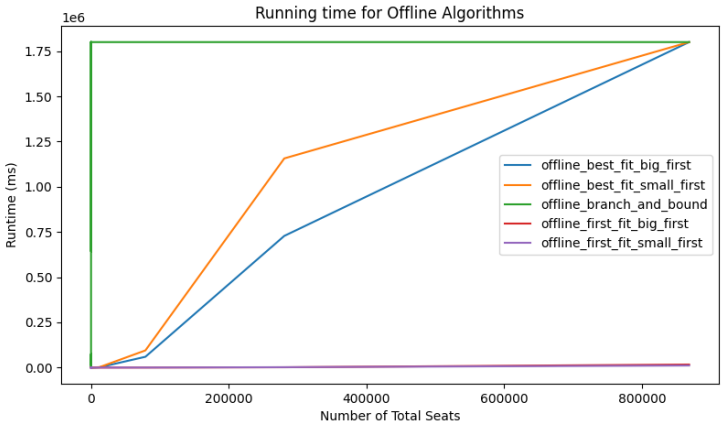
## 5.3 Results

Firstly, we will analyse the running time of the algorithms. As depicted in Figure 3, it seems that all algorithms have had the expected running time: Branch and Bound sky-rocketed with a calculated bound of $O(N!)$, where $N$ is the cinema size, reaching the 30-minute time limit from the beginning. On the other hand, Best Fit exponentially increased towards reaching the time-limit as the cinema size increased, for both the online and offline cases. As far as First Fit is concerned, it appears to run in linear time, as also shown in Subfigure 3c, definitely faster than all the other algorithm implementations.

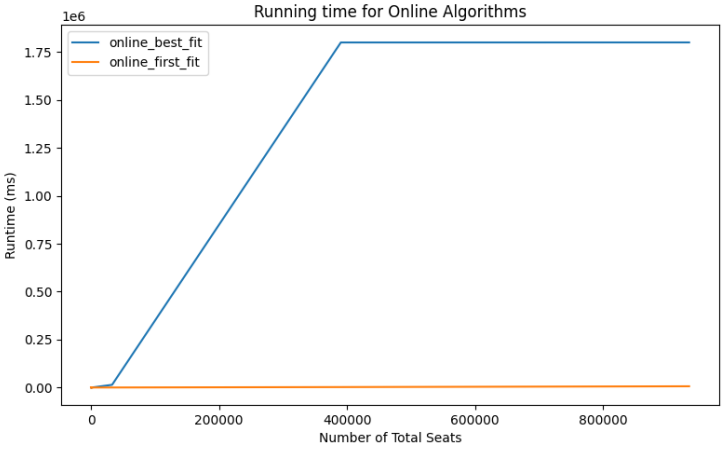
In terms of the ratio of people seated, we noticed significant differences between the algorithms. In figure 4, we analysed the ratio of people seated when compared to the number of total people coming ($\frac{\text{nr people seated}}{\text{nr total people coming}}\%$). For both online and offline algorithms, it is evident that most of them managed to seat around 80% of the coming people. While the First Fit implementations kept their constant performance, Branch and Bound and Best Fit started seating less and less people, as cinemas grew larger. This is accounted for by the fact that in their case, the time limit was often reached.

Another interesting observation is that First Fit Big First performed overall better for bigger cinemas than First Fit Small First, seating around 80% of the people, while the latter managed to seat around 60%. Hence, although we thought that First Fit Small First would solve the suboptimality problem where the algorithm is stuck in a local optimum, it turned out that the local optimum found by Big First was better than the one found by Small First.

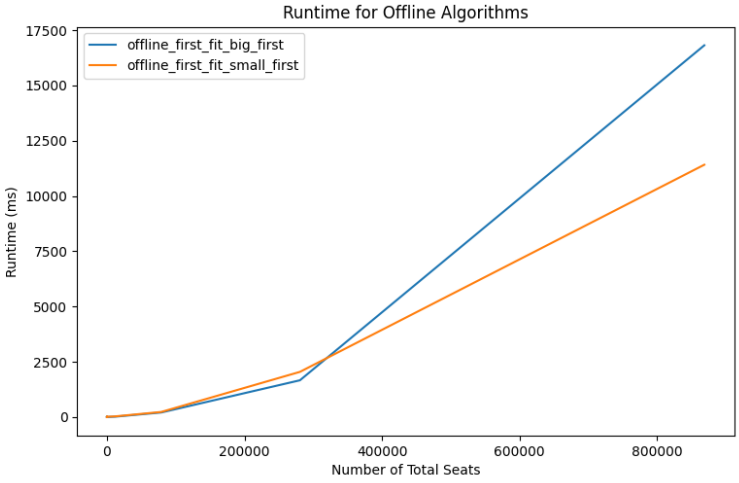
From subfigures (a) and (b) it is however unclear what happens for smaller cinemas, thus we will look at that case more in-depth in (c), where we compare the performance of Best Fit Big First and Branch and Bound. It is proven that Branch and Bound finds an optimal solution. However, that is not reflected in our graphs. As observed from this subfigure, Branch and Bound times out in most of the cases (the triangles), even when less than 500 people are coming, whereas that is not the case for Best Fit. When it times out, Branch and Bound outputs the best (partial) solution found until then, which results in its performance being worse than Best Fit, although the solution is not empty. Additionally, due to these time limitations, we cannot infer any general conclusions about the comparison of the algorithms for very small cinema sizes, as Best Fit and Branch and Bound perform arbitrarily better in some cases than others, very much dependant on the cinema layout.




**(a)** Running Time for All Offline Algorithms

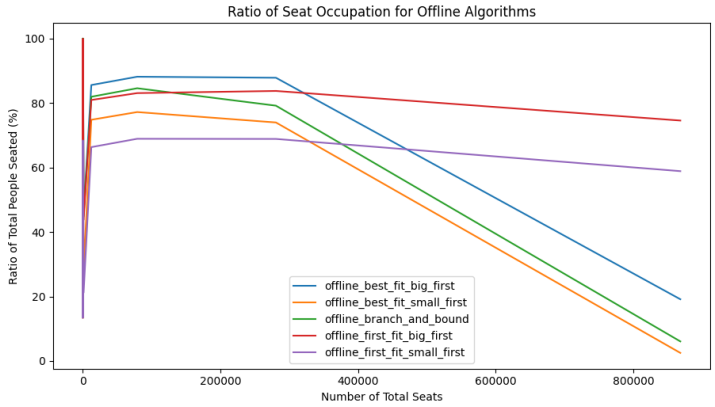



**(b)** Running Time for All Online Algorithms

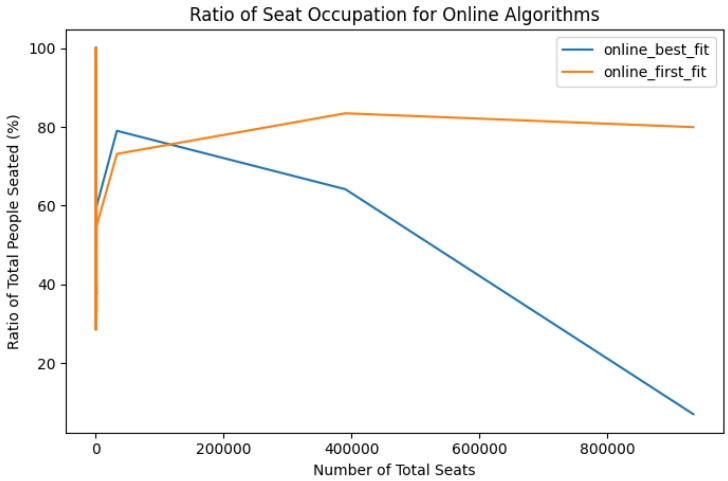



**(c)** Running Time for First Fit Offline Algorithms

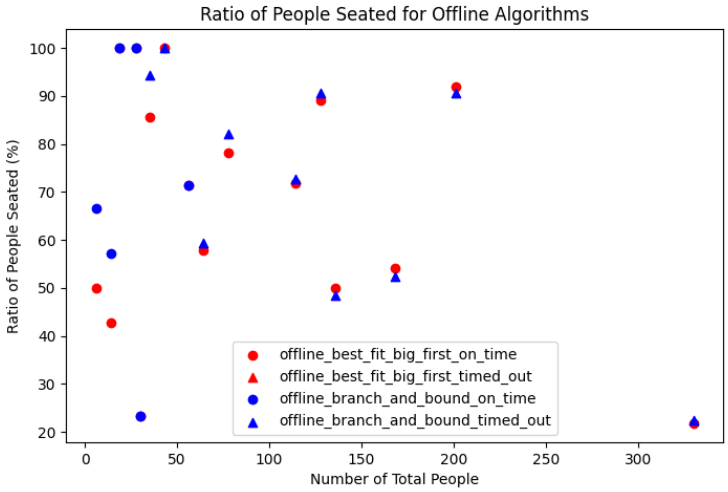
**Figure 3** Running Time of Offline and Online Algorithms.




**(a)** Ratio of People Seated for All Offline Algorithms.




**(b)** Ratio of People Seated for All Online Algorithms.




**(c)** Ratio of People Seated for Branch and Bound and Best Fit Offline Algorithms.

**Figure 4** Ratio of People Seated for Offline and Online Algorithms.

# 6 Further Research

When discussing about the nature of our algorithms, we can claim that we have also tried a number of other different approaches that turned out to be troublesome in terms of implementation: creating an Integer Linear Program, and modelling our problem as a graph and solving the problems using maximum independent set properties. Because we did not manage to finish implementing these two other ideas, we believe that they would be worth looking into for solving the offline problem in future research, as they are both proven to find optimal solutions.

# 7 Conclusion

## 7.1 Offline Problem

For the offline problem, the branch and bound algorithm is guaranteed to give the optimal solution. For larger cinemas however, the runtimes are very high (factorial). Depending on the available time and cinema size, Best Fit or First Fit can be used instead for very large cinemas. Although Best Fit and First Fit are not guaranteed to find the optimal solution, the runtimes are much lower and could therefore be used in a real-case/real-time scenario. First Fit has a lower runtime (linear) than Best Fit (exponential), but Best Fit tends to find better solutions. For much smaller cinemas (lower than $20 \times 20$), Branch and Bound can be used effectively. Hence, it is a matter of trading off between the solution quality (and the costs it infers) and the time spent on finding it.

## 7.2 Online Problem

For the online problem Best Fit finds better results than First Fit when it does not time out. The runtime for Best Fit is higher (exponential), so depending on the available time and the cinema size First Fit can be used instead. Both these online algorithms are not guaranteed to find the optimal solution, as groups are placed when they arrive and the order of the groups that arrive is not known in advance. We can however discuss that Best Fit is at least $\frac{3}{8}$ and at most $\frac{10}{19}$-competitive, which is an adequate seating placement in an online setting, given that in most test cases more than 70% of the people coming in could be placed.

## References


1 Frits C.R. Spieksma Danny Blom, Rudi Pendavingh. Filling a theatre in times of corona. *INFORMS Journal on Applied Analytics*, 2020.

2 Dave Freedman and Larry Shepp. An unfriendly seating arrangement. 1962.

3 Konstantinos Georgiou, Evangelos Kranakis, and Danny Krizanc. Random maximal independent sets and the unfriendly theater seating arrangement problem. *Discrete Mathematics*, 309(16):5120–5129, 2009.

4 Evangelos Kranakis and Danny Krizanc. The urinal problem. In *International Conference on Fun with Algorithms*, pages 284–295. Springer, 2010.

5 National Institute of Standards and Technology. Dictionary of algorithms and data structures.

6 Softjourn. Social distancing checkerboard algorithm.

# A Appendix

## Link to github

https://github.com/yasminvandijk/ADS_Cinema_Seating_Planning

## Dataset

| | testcase | cinema size | nr seats | nr coming people |
|---|---|---|---|---|
| 0 | Exact01 | 12 | 8 | 6 |
| 1 | Exact02 | 16 | 16 | 30 |
| 2 | Exact03 | 30 | 22 | 14 |
| 3 | Exact04 | 81 | 56 | 28 |
| 4 | Exact05 | 84 | 49 | 19 |
| 5 | Exact06 | 96 | 80 | 64 |
| 6 | Exact07 | 117 | 84 | 56 |
| 7 | Exact08 | 143 | 69 | 35 |
| 8 | Exact09 | 144 | 73 | 43 |
| 9 | Exact10 | 180 | 131 | 136 |
| 10 | Exact11 | 225 | 168 | 78 |
| 11 | Exact12 | 225 | 163 | 330 |
| 12 | Exact13 | 255 | 200 | 168 |
| 13 | Exact14 | 400 | 252 | 128 |
| 14 | Exact15 | 403 | 191 | 114 |
| 15 | Exact16 | 493 | 405 | 201 |
| 16 | Exact17 | 1271 | 1065 | 1000 |
| 17 | Exact18 | 13221 | 12289 | 5587 |
| 18 | Exact19 | 82368 | 78991 | 34737 |
| 19 | Exact20 | 295108 | 280662 | 124786 |
| 20 | Exact21 | 991014 | 868537 | 442956 |
| 21 | Online01 | 24 | 9 | 6 |
| 22 | Online02 | 28 | 14 | 9 |
| 23 | Online03 | 49 | 38 | 17 |
| 24 | Online04 | 81 | 56 | 42 |
| 25 | Online05 | 85 | 60 | 29 |
| 26 | Online06 | 96 | 80 | 65 |
| 27 | Online07 | 136 | 99 | 46 |
| 28 | Online08 | 140 | 123 | 40 |
| 29 | Online09 | 225 | 165 | 140 |
| 30 | Online10 | 255 | 200 | 168 |
| 31 | Online11 | 288 | 128 | 52 |
| 32 | Online12 | 440 | 367 | 190 |
| 33 | Online13 | 837 | 586 | 552 |
| 34 | Online14 | 1271 | 1065 | 1000 |
| 35 | Online15 | 1404 | 1020 | 756 |
| 36 | Online16 | 41118 | 32911 | 13165 |
| 37 | Online17 | 406164 | 390268 | 171488 |
| 38 | Online18 | 982065 | 934415 | 414666 |

### Offline Best Fit Big First

| testcase | nr occupied seats | runtime (ms) |
|---|---|---|
| Exact01 | 3 | 0 |
| Exact02 | 7 | 0 |
| Exact03 | 6 | 0 |
| Exact04 | 28 | 0 |
| Exact05 | 19 | 0 |
| Exact06 | 37 | 0 |
| Exact07 | 40 | 0 |
| Exact08 | 30 | 0 |
| Exact09 | 43 | 0 |
| Exact10 | 68 | 0 |
| Exact11 | 61 | 0 |
| Exact12 | 72 | 0 |
| Exact13 | 91 | 0 |
| Exact14 | 114 | 0 |
| Exact15 | 82 | 0 |
| Exact16 | 185 | 1 |
| Exact17 | 458 | 8 |
| Exact18 | 4782 | 1488 |
| Exact19 | 30629 | 59720 |
| Exact20 | 109632 | 728160 |
| Exact21 | 84795 | 1800130 |

### Offline Best Fit Small First

| testcase | nr occupied seats | runtime (ms) |
|---|---|---|
| Exact01 | 4 | 0 |
| Exact02 | 4 | 0 |
| Exact03 | 4 | 0 |
| Exact04 | 23 | 0 |
| Exact05 | 19 | 0 |
| Exact06 | 26 | 0 |
| Exact07 | 32 | 0 |
| Exact08 | 23 | 0 |
| Exact09 | 30 | 0 |
| Exact10 | 47 | 0 |
| Exact11 | 52 | 1 |
| Exact12 | 54 | 0 |
| Exact13 | 66 | 0 |
| Exact14 | 85 | 1 |
| Exact15 | 59 | 1 |
| Exact16 | 147 | 2 |
| Exact17 | 282 | 34 |
| Exact18 | 4180 | 2341 |
| Exact19 | 26821 | 94846 |
| Exact20 | 92316 | 1156715 |
| Exact21 | 11073 | 1800008 |

### Offline First Fit Big First

| testcase | nr occupied seats | runtime (ms) |
|---|---|---|
| Exact01 | 3 | 0 |
| Exact02 | 7 | 0 |
| Exact03 | 6 | 0 |
| Exact04 | 28 | 0 |
| Exact05 | 19 | 0 |
| Exact06 | 35 | 0 |
| Exact07 | 34 | 0 |
| Exact08 | 30 | 0 |
| Exact09 | 41 | 0 |
| Exact10 | 66 | 0 |
| Exact11 | 61 | 0 |
| Exact12 | 67 | 0 |
| Exact13 | 90 | 0 |
| Exact14 | 106 | 0 |
| Exact15 | 80 | 0 |
| Exact16 | 177 | 0 |
| Exact17 | 438 | 0 |
| Exact18 | 4523 | 15 |
| Exact19 | 28863 | 204 |
| Exact20 | 104521 | 1661 |
| Exact21 | 330348 | 16814 |

### Offline First Fit Small First

| testcase | nr occupied seats | runtime (ms) |
|---|---|---|
| Exact01 | 4 | 5 |
| Exact02 | 4 | 6 |
| Exact03 | 8 | 5 |
| Exact04 | 14 | 4 |
| Exact05 | 13 | 5 |
| Exact06 | 26 | 5 |
| Exact07 | 24 | 0 |
| Exact08 | 23 | 0 |
| Exact09 | 23 | 0 |
| Exact10 | 47 | 0 |
| Exact11 | 42 | 0 |
| Exact12 | 51 | 0 |
| Exact13 | 66 | 0 |
| Exact14 | 72 | 0 |
| Exact15 | 54 | 0 |
| Exact16 | 135 | 0 |
| Exact17 | 212 | 12 |
| Exact18 | 3706 | 29 |
| Exact19 | 23941 | 228 |
| Exact20 | 85944 | 2042 |
| Exact21 | 260815 | 11412 |

### Offline Branch and Bound

| testcase | nr occupied seats | runtime (ms) |
|---|---|---|
| Exact01 | 4 | 66 |
| Exact02 | 7 | 7 |
| Exact03 | 8 | 5 |
| Exact04 | 28 | 75533 |
| Exact05 | 19 | 8789 |
| Exact06 | 38 | 1800001 |
| Exact07 | 40 | 641347 |
| Exact08 | 33 | 1800001 |
| Exact09 | 43 | 1800001 |
| Exact10 | 66 | 1800001 |
| Exact11 | 64 | 1800001 |
| Exact12 | 74 | 1800001 |
| Exact13 | 88 | 1800001 |
| Exact14 | 116 | 1800001 |
| Exact15 | 83 | 1800001 |
| Exact16 | 182 | 1800001 |
| Exact17 | 449 | 1800001 |
| Exact18 | 4579 | 1800001 |
| Exact19 | 29386 | 1800002 |
| Exact20 | 98818 | 1800047 |
| Exact21 | 26857 | 1800078 |

### Online First Fit

| testcase | nr occupied seats | runtime (ms) |
|---|---|---|
| Online01 | 5 | 54 |
| Online02 | 9 | 0 |
| Online03 | 15 | 0 |
| Online04 | 12 | 0 |
| Online05 | 22 | 0 |
| Online06 | 29 | 0 |
| Online07 | 41 | 0 |
| Online08 | 38 | 0 |
| Online09 | 69 | 0 |
| Online10 | 74 | 0 |
| Online11 | 45 | 0 |
| Online12 | 144 | 0 |
| Online13 | 234 | 0 |
| Online14 | 330 | 0 |
| Online15 | 414 | 0 |
| Online16 | 9629 | 170 |
| Online17 | 143089 | 2335 |
| Online18 | 331483 | 6462 |

### Online Best Fit

| testcase | nr occupied seats | runtime (ms) |
|---|---|---|
| Online01 | 6 | 1 |
| Online02 | 5 | 0 |
| Online03 | 15 | 0 |
| Online04 | 12 | 0 |
| Online05 | 22 | 0 |
| Online06 | 33 | 0 |
| Online07 | 34 | 0 |
| Online08 | 40 | 0 |
| Online09 | 71 | 0 |
| Online10 | 82 | 0 |
| Online11 | 52 | 0 |
| Online12 | 148 | 1 |
| Online13 | 236 | 2 |
| Online14 | 374 | 16 |
| Online15 | 452 | 12 |
| Online16 | 10401 | 14122 |
| Online17 | 110059 | 1800014 |
| Online18 | 29458 | 1800240 |