Epidemiology-constrained seating plan problem

Jakub Dąbkowski, Przemysław Kacperski, Mariusz Kaleta *

Abstract. The emergence of an infectious disease pandemic may result in the introduction of restrictions in the distance and number of employees, as was the case of COVID-19 in 2020/2021. In the face of fluctuating restrictions, the process of determining seating plans in office space requires repetitive execution of seat assignments, and manual planning becomes a time-consuming and error-prone task. In this paper, we introduce the Epidemiology-constrained Seating Plan problem (ESP), and we show that it, in general, belongs to the NP-complete class. However, due to some regularities in input data that could affect computational complexity for practical cases, we conduct experiments for generated test cases. For that reason, we developed a computational environment, including the test case generator, and we published generated benchmarking test cases. Our results show that the problem can be solved to optimality by CPLEX solver only for specific settings, even in regular cases. Therefore, there is a need for new algorithms that could optimize seating plans in more general cases.

Keywords: seating plan, office seat allocation, distancing policy, mixed-integer programming, NP-complete problem

1. Introduction

In an epidemiological emergency, non-pharmaceutical interventions (NPIs) are one of the main tools of controlling the pace of the spread of the disease. Maintaining physical distance between people is one of the NPIs applied in an office space. Institutions may be obliged by governmental policy institutions to reorganize office space in a way that provides appropriate conditions for social distancing, as was the case with COVID in 2020/2021.

*Faculty of Electronics and Information Technology, Warsaw University of Technology, Warsaw {jakub.dabkowski.stud, przemyslaw.kacperski.stud, mariusz.kaleta}@pw.edu.pl

Typical restrictions apply to the maximum number of individuals in an enclosed space at the same time. This is usually expressed by the permissible percentage occupancy in an office building and by the minimal distance that must be maintained between employees' seats. Therefore, institutions must plan the deployment of personnel, so-called seating plans, to comply with the imposed restrictions. However, it brings severe economic and organizational problems. Seating plans are often related to the manual and time-consuming process of their creation which is prone to various types of errors and may lead to solutions far from optimal. Moreover, limits on the maximum number of people in an enclosed space at the same time can change dynamically, and the seating plan must be kept adjusted. As a result, the problem is complex, needs to be continuously being solved, and touches many private and public institutions. Therefore, any supporting tool and automatization in laying the seating plan can bring significant benefits to institutions.

There are several elements that could be automated in the whole process of seating plan arrangement. For instance, collecting information on possible seats and distances between them can be a challenging and significant obstacle. However, in this paper, we focus strictly on optimising the seating plan, ignoring the problem of data collection. We perceive optimization as one of the steps, but a key one, in the wider process of seating plan management.

The problem of creating seating plans has been considered in the literature in different contexts. Lewis and Carroll consider a mathematical model for seating guests at celebrations such as weddings [3]. The authors formulated a mathematical model for determining seating plans for guests at receptions, taking into account complex preferences for generating seating plans, such as the desire to seat related or friendly people together, as well as attempting to seat separately people between whom conflicts are known to exist. They developed an integer programming model and proved its NP-completeness. A two-stage heuristic algorithm based on tabu search has been proposed. In its first stage, the algorithm finds a solution that satisfies hard constraints, while in the second stage, the heuristic searches solution neighbourhood to improve soft constraints. The problem formulation and proposed algorithm are strictly related to complex preferences. Although such complex preferences may be included in our further work, the problem we consider in this paper is simpler in that field, but at the same time, it requires a new type of distancing constraint.

The problem of the seating plan is also considered in the context of a supply-demand system. The paper [6] describes a problem arising from the discrepancy between the variable passenger demand for seats of different categories and the way passengers are seated on intercity trains. The seating plan is arranged for trains under dynamic demand with the support of a genetic algorithm. The research shows that this way of seat allocation on trains positively affects both the passenger comfort and profits of railway operators. This paper presents a completely different approach compared to the one we consider in this paper. The specificity of the problem forces the use of a black-box simulation model. In contrast, the problem we formulate in this paper takes the form of an explicit programming model and allows us to use optimization techniques appropriate for this kind of model.

The research that comes closest to the problem in terms of distance constraints is derived from the exam session planning problems. Chaki and Anirban[1] developed the exam session planning problem previously considered in the literature into more detailed seating arrangements in examination rooms in such a way as to make cheating during exams more difficult. They developed a set of constructive heuristics and the results they obtained show that the optimized seating plans are better than those generated manually, even tough they could not achieve an optimal solution in general. Their approach differs from the problem we consider in this paper in that it considers room assignment and distancing model is strictly related to the distribution of different versions of exam tests.

Seating arrangement is also broadly discussed in a range of other papers of a social background. McCorskey and McVetta [4] focus on the instructional communication between students in classrooms, while Mehrabian and Diamond [5] examine the impact of such factors as sensitivity to rejection on one's seating preferences. Sailer [7] proposed an interesting overview of the parliament seating plans and political cultures. Those papers, though investigating different aspects of arranging seating space, are focused more on the social and psychological factors, in which they differ from the approach proposed by the authors.

Despite some similarities between the researches related to seating plan problems and the problem we consider, to the best of our knowledge there is no work in the literature on the epidemiology-constrained seating plan for office spaces. In this paper we formulate the epidemiology-constrained seating plan problem and we show that it belongs to the NP-complete class. Since office settings reveal some typical structure we analyse computational complexity for this setting and we show that the problem can be solved to optimality for real cases of moderate sizes. For that purpose we developed testing environment that includes problem generator, visualisation and testing tools and resulting set of benchmarking test cases. We believe that all of that can support further research related to different or more complex settings and larger cases. Since for the biggest test cases we obtained suboptimal results with GAP around 20%, it reveals a need for more advanced and dedicated algorithms to be developed in further research.

2. Mathematical model

2.1. Problem statement

Let us consider a working space in which there is a set S of possible workplaces, hereinafter referred to as "seats". Two seats, $s, t \in S$, are at distance D_{st}, where D is the symmetric square matrix, where each value is an Euclidean distance between two given seats. If D_{st} is lower than the minimum permitted distance D^{min} between two seats, the seats s and t cannot be used at the same time. The **Epidemiology-constrained Seating Plan problem** (ESP) is to find a subset of seats that satisfy minimal distance constraint. We define two variants of the ESP problem.

Problem ESP-ME. In the Epidemiology-constrained Seating Plan problem with Maximal number of Employees, we are interested in finding a seating plan that maximizes the number of occupied seats. Let E be the number of employees, and let us assume that not all employees E can be hosted in the working space at the same time. Only one employee can occupy each seat, so the problem is to maximize the number of occupied seats. The result is given as a set $X \subseteq S$ of seats to be in use.

Problem ESP-AD. In the Epidemiology-constrained Seating Plan problem with maximal Average distance, the number of employees E to be seated is small enough, so all employees are given a seat. The problem is to choose the best seats, so the average distance between occupied seats is maximized.

2.2. Complexity of the problem

First, let us consider the ESP-ME problem. Let the graph $G = (S, L)$ be a graph induced by the set of nodes S, and let L be a set of edges $\{(s,t) : s, t \in S, D_{st} \geqslant D^{min}\}$, so there is no edge between nodes representing seats that are too close. Then, the problem is equivalent to the problem of finding the maximal clique in graph G. Since the maximal clique problem can be reduced to the ESP-ME problem in linear time, it is clear that the ESP-ME problem is NP-complete.

The problem ESP-AD is equivalent to finding the clique of size E. Although the decision problem to check if a k-Clique exists in a given graph can be solved in polynomial time for a fixed k, in general, assuming that k is a part of input, the clique decision problem is known to be NP-complete [2], so the ESP-AD is NP-complete.

2.3. Mixed-integer linear programming model

We formulate the ESP-AD problem as a mixed-integer linear programming (MILP) one. Let E be the number of employees to be seated. The task is to assign employees E to some seats from S or, in other words, to select E seats from S. We assume that $E \leqslant S^{max} \cdot |S|$, where $S^{max} \in \langle 0; 1\rangle$ is a share of seats that can be occupied at the same time. Therefore, the only epidemic-related constraint for the seating plan is the one related to the minimum required distance between each pair of assigned seats.

For the purpose of modeling the assignment decisions, we define a binary variable x_s which is 1 if the seat s is occupied by some employee, and 0 otherwise. We also introduce variable o_s which is the sum of the distances between place s and all other occupied places in S. Then the problem takes the following form of a MILP one:

$$\min \sum_{s \in S} o_s \tag{1}$$

subject to

$$o_s \geqslant \sum_{t \in S, t \neq s} D_{st} \cdot (x_s + x_t - 1) \quad \forall_{s \in S} \tag{2}$$

$$x_s + x_t \leqslant 1 \quad \forall_{s,t \in S} s \neq t \wedge D_{st} < D^{min} \tag{3}$$

$$\sum_{s \in S} x_s = E \tag{4}$$

$$o_s \geqslant 0 \quad s \in S \tag{5}$$

$$x_s \in \{0,1\} \quad s \in S \tag{6}$$

Constraint (2) defines a lower bound on the decision variable o_s with the sum of the distances between seat s and the other seats in S. The bound is positive only if two seats, that is s and t, are chosen. Taking into account non-negative values of o_s defined by (5) and minimization pressure in the objective, o_s takes the value equal to the sum of appropriate distances in the optimal solution. The exclusion of seats that are too close is implemented in constraint (3), and constraint (4) ensures that the sum of all occupied seats must be equal to the number of employees E.

Chosen seats create a subset of S that induces a complete graph. For a pair of seats (s, t) the distance is included twice as D_{st} and D_{ts} in the sum $\sum_{s \in S} o_s$. Since the number of edges in a complete graph with E nodes is equal to $\frac{E(E-1)}{2}$, then the average distance between seats can be expressed by

$$\frac{\sum_{s \in S} o_s}{E(E-1)}. \tag{7}$$

For simplicity of notation, we omit constant factor in the objective (1), which still chooses the solution with the minimum of average distances between occupied seats.

3. Computational environment

Although the ESP-AD problem belongs to the NP-complete class, one can expect that the practical computational complexity of real cases may be lower due to specific settings of working spaces. We developed a set of tools for generated test cases and conducted computational experiments. The environment consists of the following components:

1. Generator of test cases
2. Optimisation component for the ESP-AD problem
3. Visualisation tool for resulting seating plan
4. Component automating the tasks mentioned in points 1-3.

3.1. Test cases generator

Since there are no benchmarking data sets that could be used for the ESP-AD problem, we developed the test case generator. The generator assumes that seats are arranged in sectors. A sector is a group of seats and desks that are adjacent to each other. Figure 1 illustrates a sector of size six.

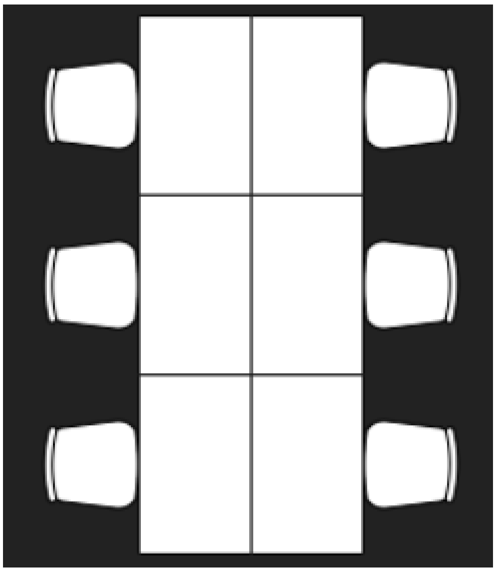

Figure 1. An example appearance of a sector with six places arranged in two columns and three rows.

In a test case, seats are composed of a sequence of adjacent sectors. The generator script takes in the following parameters:

- the size of a sector,
- the number of seats,
- the number of employees E considered in the seating process,
- the minimum distance between occupied places D^{min},
- the maximum percentage occupancy of the number of places available S^{max}.

The generator creates an arrangement of the workspace made of the sectors of a given size and given the total number of seats. Sectors consist of 2 columns and in each column there is a certain number of seats defined in structures that describe the entire seating area. Predefined structures also make it possible to define specific distances of seats next to and opposite to each other, distances between sectors, dimensions of desks and chairs and their location in the office space.

Generator constructs the matrix D_{st} by calculating distances for each pair of seats. It should be emphasized that the distance between each pair of seats is calculated regardless of whether the seats are in the same sector or not.

The test case generator has been implemented as a script in Python, and input parameters are in the form of a *.py* file containing the data structure (dictionary) of the above parameters. The script generates two output files: a *.dat* file with the input data for the solver and a *.xml* file describing the objects that are visualized with an external program. The first one contains the generated matrix of the distance between the places in the considered office space and the parameters needed to solve the task of allocating places to employees: the number of places, the number of employees, the minimum distance that must be kept between employees and the percentage of places in the office. The *.dat* file allows to create input data for the IBM CPLEX STUDIO IDE program, which solves the problem of determining places to be occupied for given parameters. All test cases have been published as an open repository at the following address https://github.com/JacobJustCoding/Epidemiology-constrained-seating-plan-problem.

3.2. Optimization component

The mathematical model (1)-(6) has been implemented in the IBM CPLEX STUDIO IDE. The model is completed with a data file produced by a test case generator, and then solved by the CPLEX solver.

3.3. Visualisation tool

The visualizer is an important tool for previewing the created test cases. With its help, the process of verifying created test cases becomes much simpler due to the possibility of depicting the problem in a graphical environment. The visualization is performed on the basis of the *.xml* file created by the generator, compatible with the draw.io tool used to visualize the diagram. To run the visualization, it is enough to open the generated file in the draw.io program.

3.4. Automation component

To ensure the efficiency of the testing procedure, the whole process consisting of the following steps:

- creation of a test case according to the input parameters,
- generating a parameter file for the model implemented in CPLEX,
- running the model and getting the data about the place allocations

has been automated by the script.

4. Conducted experiments

In order to check the amount of time needed to compute the result, experiments were carried out consisting of establishing seating plans for various sets of input data. All computational experiments were performed on a personal computer with an Intel i7-5600U processor and 16 GB of DDR3 RAM, running under the Windows 10 operating system. In order to eliminate disturbances in computational times to reduce the influence of random external factors on the results, each of the test cases was run four times, and then the average computational time was calculated.

4.1. Test cases

We generated test cases grouped in the size of sectors. The sector sizes considered are as follows:

- 6 seats (2 columns and 3 rows),
- 10 seats (2 columns and 5 rows),
- 12 seats (2 columns and 6 rows),
- 20 seats (2 columns and 10 rows),
- 24 seats (2 columns and 12 rows),

which constitutes 5 basic test cases.

In turn, the factors whose influence on the time of solving the task of designating places for employees was examined are as follows:

- increase in the number of seats with a simultaneous and proportional increase in the number of employees: the minimum distance between the seats and the percentage of allowable occupancy of seats,
- increase in the minimum distance between seats with a fixed number of seats, employees, and the percentage of allowable occupancy.

For each basic test case, we checked first increasing size in terms of the number of seats and employees and then increasing D^{min}. We solved the model for four increasing values in both cases, which in total gives 5 cases * 2 parameters * 4 values = 40 optimization tasks.

We need to emphasize that in each test case, we assumed $E = S^{max} * |S|$, which makes the cases very tight, which seems to be reasonable from the practical perspective since institutions tend to maximize E. Table 1 presents results for an increasing number of seats and employees for each basic test case. The number of feasible solutions is the number of solutions found during the search of branch&cut tree. The number of iterations is the number of simplex iterations that CPLEX has performed in solving continuous relaxations of subproblems in branch&cut tree nodes, which is a kind of measure of computational effort. Computational times and the number of iterations increase along with the increase of E and size of S, and the pace of the increase suggests exponential-like growth. For the biggest instances, that is for 192 employees and 384 seats, it was not possible to solve the problem. Moreover, the quality of the solution was quite poor, for example, after 1500[s], the typical GAP is around 20%.

It is worth paying attention to the relatively small number of feasible solutions found. We expect that this is a result of assumption $E = S^{max}$ and the regular arrangement of seats. Figure 2 illustrates an exemplary allocation of employees to seats. One can observe a kind of regularity. However, even with this regularity and relatively low number of feasible solutions still, the problem could not be solved to optimality for instances that are of reasonable sizes. Interestingly, the computational time for a given number of seats and employees is decreasing along with sector size increase. For instance, the computational time for 24 seat sectors is approximately 42 times shorter than in the case of 6 seat sectors, for $E = 96$ and $|S| = 192$.

Table 2 presents results for increasing values of parameter D^{min} assuming $E = 96$ and $|S| = 192$. Increased D^{min} may lead to a situation in which the required number of employees E cannot be achieved, and the problem is infeasible. In such cases, CPLEX proposed the relaxation of constraint (4) automatically and solved the resulting relaxed problem. It can be observed that computational times rapidly decrease when the minimal required distance D^{min} increases. On the other hand, if the distance D^{min} is reduced to 1.5[m] no test case could be solved to optimality. We interpret this as with lower D^{min} number of feasible seating arrangement increases, and it creates larger solution space for which the classical branch&cut approach cannot close tree nodes efficiently. Once again, this is a strong signal that the problem will probably be difficult to solve for moderate-size cases and more irregular arrangements.

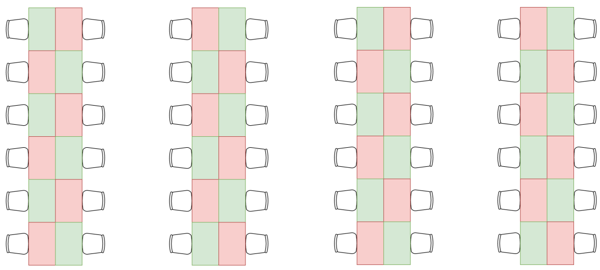

Figure 2. Exemplary allocation for four sectors of size 12, total number of seats equal to 48, E=24, $D^{min} = 2.5$[m]. Green and red colors represent seats occupied and unoccupied, respectively.

5. Summary

In the paper, we formulated the Epidemiology-constrained Seating Plan problem (ESP). We show that the problem is NP-complete. Knowing that, we conducted com-

Table 1. Results for growing number of seats and the number of employees with $D^{min} = 2.5$[m] and $S^{max} = 0.5$ (up to 50% of seats can be occupied).

No.	**Sector size**	**Parameter**	**Value of parameter**	**Average time of execution [s]**	**Number of feasible solutions found**	**Number of iterations**
1	6 seats (2 columns, 3 rows)	$\|S\|$	48	0.40	6	206
		E	24			
2		$\|S\|$	96	3.37	8	33700
		E	48			
3		$\|S\|$	192	105.04	8	1173886
		E	96			
4		$\|S\|$	384	*	*	*
		E	192			
5	10 seats (2 columns, 5 rows)	$\|S\|$	48	0.36	5	187
		E	24			
6		$\|S\|$	96	0.74	5	4047
		E	48			
7		$\|S\|$	192	20.63	14	252694
		E	96			
8		$\|S\|$	384	*	*	*
		E	192			
9	12 seats (2 columns, 6 rows)	$\|S\|$	48	0.35	5	159
		E	24			
10		$\|S\|$	96	0.65	3	3842
		E	48			
11		$\|S\|$	192	35.15	13	651470
		E	96			
12		$\|S\|$	384	*	*	*
		E	192			
13	12 seats (2 columns, 10 rows)	$\|S\|$	48	0.33	4	168
		E	24			
14		$\|S\|$	96	0.46	5	336
		E	48			
15		$\|S\|$	192	3.05	5	23438
		E	96			
16		$\|S\|$	384	*	*	*
		E	192			
17	24 seats (2 columns, 12 rows)	$\|S\|$	48	0.31	3	167
		E	24			
18		$\|S\|$	96	0.43	3	354
		E	48			
19		$\|S\|$	192	2.48	4	10916
		E	96			
20		$\|S\|$	384	*	*	*
		E	192			

* No optimal solution found within 7200[s] of computation.

Table 2. Experiments carried out for the increasing minimum distance between employees in the office space. For all test cases presented: number of seats = 192, number of employees = 96, maximum occupancy by employees = 0.5 (50%).

No.	**Sector size**	D^{min} [m]	**Average time of execution [s]**	**Was relaxation performed? ****	**Number of feasible solutions found**	**Number of iterations [s]**
21	6 seats (2 columns, 3 rows)	1.5	*	*	*	*
22		2.5	105.04	No	8	1173886
23		3.0	2.65	Yes, $\tilde{E} = 64$	0	206
24		4.0	6.67	Yes, $\tilde{E} = 32$	0	209
25	10 seats (2 columns, 5 rows)	1.5	*	*	*	*
26		2.5	15.80	No	14	252694
27		3.0	2.96	Yes, $\tilde{E} = 58$	0	208
28		4.0	8.50	Yes, $\tilde{E} = 39$	0	227
29	12 seats (2 columns, 6 rows)	1.5	*	*	*	*
30		2.5	30.78	No	13	651470
31		3.0	2.93	Yes, $\tilde{E} = 48$	0	212
32		4.0	9.59	Yes, $\tilde{E} = 32$	0	227
33	20 seats (2 columns, 10 rows)	1.5	*	*	*	*
34		2.5	2.87	No	5	23438
35		3.0	3.20	Yes, $\tilde{E} = 50$	0	209
36		4.0	9.93	Yes, $\tilde{E} = 40$	0	228
37	24 seats (2 columns, 12 rows)	1.5	*	*	*	*
38		2.5	2.51	No	4	10916
39		3.0	3.00	Yes, $\tilde{E} = 48$	0	213
40		4.0	12.08	Yes, $\tilde{E} = 32$	0	230

* No optimal solution found within 7200[s] of computation.

** The problem has been relaxed and solved for decreased number of employees $\tilde{E}$.

putational experiments for test cases with regular arrangements of possible settings, that could be met in practice. We also assumed that the number of employees to be seated is equal to the maximal number of persons allowed to stay in the workspace. This seems to be reasonable from a practical perspective and also strongly reduces the space of feasible solutions. Our results show that the problem can be solved to optimality with CPLEX only for moderated and small sizes. It was not possible to solve the problem for 192 employees and 2.5[m] distancing and after decreasing minimal distance to 1.5[m] even for 96 employees.

As a result, it is clear that further research on solving methods must be conducted to make the problem solvable in larger practical cases. We published a set of benchmarking test cases, however, it is now limited to regular structures. In further work, it is needed to consider other regular structures, like arrangements of sectors in two dimensions, as well as irregular structures. On the other hand, we believe that for regular structures, efficient constructive algorithms can be developed in the future.

References

[1] Chaki P. K. and Anirban S. Algorithm for efficient seating plan for centralized exam system. In 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), pages 320–325, 2016.

[2] Karp R. M. Reducibility among combinatorial problems. In Miller R. E., Thatcher J. W., and Bohlinger J. D., editors, Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, pages 85–103, Boston, MA, 1972. Springer US.

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

[4] McCorskey J. C. and McVetta R. W. Classroom seating arrangements: Instructional communication theory versus student preferences. 27(2):99–111. Publisher: Routledge.

[5] Mehrabian A. and Diamond S. G. Seating arrangement and conversation. 34(2):281–289. Publisher: [American Sociological Association, Sage Publications, Inc.].

[6] Nakagawa S., Shibata M., and Fukasawa N. Optimization system of reserved/non-reserved seating plans for improving convenience and revenue of intercity express trains. International Journal of Transport Development and Integration, 1(3):359–370, 2017.

[7] Sailer K. Degrees of opposition and cooperation. How seating plans and parliament layouts reflect and give rise to political cultures.

Please leave some space here for submission and acceptance date