Proceedings of the 2021 IISE Annual Conference
A. Ghate, K. Krishnaiyer, K. Paynabar, eds.

An Automated Tool for Optimal Classroom Seating Assignment with Social Distancing Constraints

Kyle K. Greenberg, Trey Hensel, Qihan Zhu, Sander Aarts, David B. Shmoys, Ph.D.
College of Engineering; Cornell University
Ithaca, NY, 14850, USA

Samuel C. Gutekunst, Ph.D.
Bucknell University
Lewisburg, PA, 17837, USA

Abstract

Social distancing has become a necessity due to COVID-19, requiring schools to reduce classroom capacities to host in-person students. In doing so, schools seek to maximize the number of seats that can be used within a classroom, while ensuring that no pair of usable seats violates social distancing guidelines. We model this problem as a graph-theoretic maximum independent set problem and develop a user-friendly tool that solves real instances of the problem. We then use that tool to create optimal classroom seating plans for a major university.

Our core model considers the case of a classroom with fixed seats. This problem can be expressed as a graph, identifying seats as nodes and inserting edges between seats that are closer than some prescribed threshold. A maximum independent set in this graph corresponds to an optimal seating plan. Our seat assignment tool allows any user to solve this problem by uploading an architect's drawing of a classroom. Then, computer vision aids the user by locating seats, and the tool finds and prints an optimal plan. Our tool also allows users to easily incorporate additional requirements, such as designated teacher spaces and the inclusion of movable chairs.

Our tool helped automate the classroom planning process at Cornell University where its ease of use allowed it to be run on hundreds of classrooms. Compared to initial reduced-capacity classroom estimates determined by the Office of the University Architect, it helped identify over 400 additional seats that could be used.

Keywords
COVID-19; Maximum Independent Set; Integer Programming; Discrete Optimization; Optimization Tool

1. Introduction

COVID-19 has caused substantial challenges for colleges and universities hoping to host in-person students. Operations Research has played a critical role in helping schools decide if they can reopen and, if so, mitigating the spread of COVID-19 [1, 2]. Hosting in-person classes has created the challenge of facilitating social distancing [3, 4], which greatly limits the maximum occupancy of each classroom. As students must be placed sufficiently far apart from each other, only a fraction of seats can be used, limiting capacity for in-person classes. As an extreme example, we found that a tightly-packed lecture hall with 275 seats at Cornell University could hold at most 32 students, if all students were spaced at least 6 feet apart.

In this paper, we study the socially-distanced seat assignment problem: selecting a set of seats within a classroom that is as large as possible, while ensuring that all selected seats are sufficiently far apart. We first formalize the seat assignment problem and show that it can be modeled as a maximum independent set problem. We then create a user-friendly tool that maximizes the number of seats in a classroom while following social distancing guidelines. Our tool outputs socially-distanced seating charts that can be used by universities to plan classrooms and facilitate a safer in-person semester. Finally, we run and analyze our tool on real classrooms; partnering with Cornell, we used the tool to help make improvements on 95 classrooms and identify over 400 potential additional seats.

In Section 2, we model the socially-distanced seat assignment problem as a maximum independent set problem and write an integer programming formulation. Then, in Section 3, we detail our tool and its design. Finally, in Section 4, we discuss our work with Cornell; we highlight our tool's improvements in finding extra seats in real classrooms.

2. Modeling Framework

In our core model, we consider the following version of the socially-distanced seat assignment problem (*seat assignment problem*): consider a classroom where all possible seat locations are fixed (e.g., a lecture hall with immovable rows). Given n seats labeled $1,2,\dots,n$, we want to identify a subset $S \subseteq \{1,2,\dots,n\}$ of *assigned seats* such that 1) S contains as many seats as possible and 2) each pair of assigned seats in S are sufficiently far apart from one another. Throughout this paper, we say that seats i and j are *sufficiently far apart* when the center of seat i is at least 6 feet away from the boundary of seat j. However, our model allows the user to modify this definition based on local COVID-19 guidance.

We formulate an integer program for our problem as follows (see [5] for more details on integer programming). Let x_i be a binary decision variable that indicates whether seat i is in S ($x_i = 1$) or not in S ($x_i = 0$), for $i = 1,\dots,n$. Our objective is to maximize the number of assigned seats $|S| = \sum_{i=1}^{n} x_i$. We also need constraints that prevent us from including pairs of seats that are not sufficiently far apart from both being in S. Let

$$\mathcal{C} = \{\{i,j\} : \text{seats } i \text{ and } j \text{ are not sufficiently far apart}\}$$

denote the set of *conflicting pairs* (i.e., pairs of seats that cannot both be used in a socially-distanced classroom). Again, any definition of “sufficiently far apart” can be used in our model. We prevent both seats in a conflicting pair from being included in S by the constraint $x_i + x_j \leq 1$ for any $\{i,j\} \in \mathcal{C}$. Our integer program is:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} x_i && (1)\\ \text{subject to} \quad & x_i + x_j \leq 1, && \text{for all } \{i,j\} \in \mathcal{C},\\ & x_i \in \{0,1\}, && \text{for all } i = 1,\dots,n. \end{aligned}$$

An optimal solution to this integer program corresponds to an optimal solution to our original problem: a maximum-sized subset of seats that are all sufficiently far apart from each other. In Section 3, we design a computational tool to take in an architect's drawing of a classroom, automate the process of identifying the seats and conflicting pairs, solve the resulting integer program, and visually output an optimal solution.

The problem in (1) is a *maximum independent set problem* [6]. Given a graph $G = (V,E)$, an independent set is a subset of nodes $S \subseteq V$ such that if $i,j \in S$, then $\{i,j\} \notin E$. That is, any pair of nodes in S are nonadjacent. A maximum independent set is an independent set of maximum cardinality. For the seat assignment problem, V is the set of seats $\{1,\dots,n\}$ and E is the set of conflicting pairs $\mathcal{C}$. **Figure 1** shows a graphical representation. The set $\{1,4,9,12\}$ is an optimal solution (and maximum independent set) for the classroom in **Figure 1A**.

While our core seat assignment model handles classrooms with fixed seating, it can be extended to handle many other types of rooms including cases with non-fixed (i.e., movable) seats. For example, the classroom in **Figure 2** has fixed desks but non-fixed chairs. Instead of only using the seat locations drawn on the architect’s plan, we create nodes for different potential seat locations. Nodes included within the solution set indicate locations for physical chairs to be placed. In this way, our tool can improve upon a solution that only considered seat locations in the original architect's drawing. The original architect’s drawing in **Figure 2**, for example, has 91 possible chair locations. By adding possible chair locations spaced every 3.5 inches apart along each desk, we get 778 possible chair locations. Doing so allows us to improve the socially-distanced capacity from 22 to 28. This concept can also be applied to empty classrooms where chairs can be arbitrarily placed.

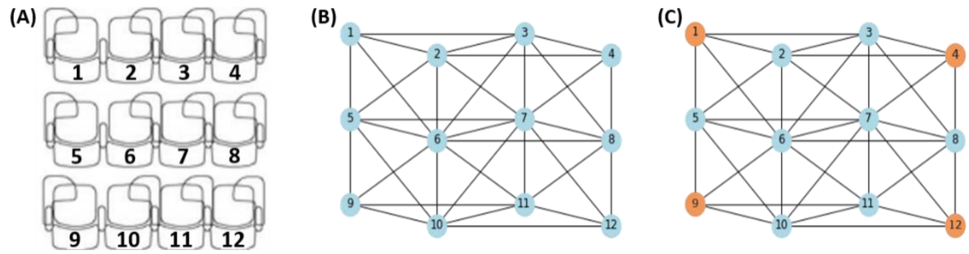

Figure 1: (A) An example classroom. **(B)** The corresponding graph. Nodes correspond to seats, and edges, to conflicting pairs (seats that are too close to each other to both be used in a socially-distanced classroom). **(C)** A maximum independent set is indicated in orange.

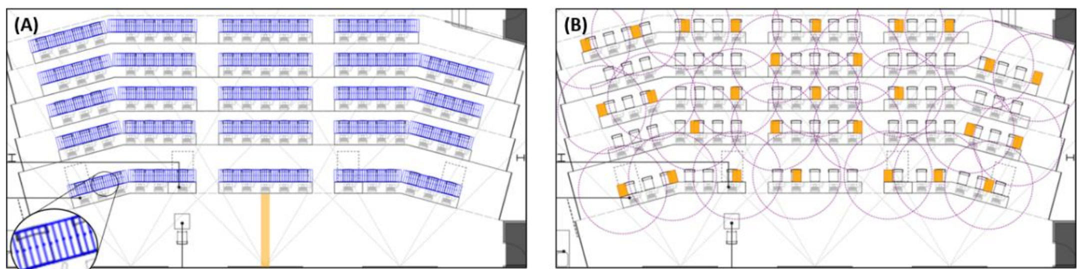

Figure 2: (A) The blue rectangles are possible chair locations spaced 3.5 inches apart along fixed tables. **(B)** The orange rectangles make up an optimal solution for this set-up. Black chair outlines are from the architect's drawing.

It is important to note that an optimal solution to the integer program for fixed seating is an optimal use of classroom seats. In contrast, an optimal solution to the integer program for the non-fixed seating problem depends on the number of possible chair locations our model considers; by adding more possible chair locations, we might obtain better solutions. However, the computational complexity increases as more chair locations are considered; we explore this trade-off in Section 4.

3. Tool Design

To use the integer program (1) from Section 2, we need only identify the seats and conflicting pairs in an architectural drawing. To ease and expedite this process, we designed a tool that allows non-technical users to upload an architectural drawing of a classroom; from that drawing, the tool automates the process of identifying seats and conflicting pairs, automatically creates and solves the integer program (1), and outputs an image clearly showing an optimal assignment of seats. Importantly, the tool is designed to be broadly accessible and does not require any experience with programming or optimization. The user uploads an image of a classroom layout containing a scale (e.g., **Figure 3B** before the seats/scale are highlighted, or **Figure 4A** for an empty classroom). Then, a simple graphical user interface provides the user with a menu to navigate between steps. The tool provides users with simple instructions at each individual step and keeps track of which steps the user has completed (see **Figure 3A**). Our full code for the tool, and a sample video showing its use, are available on GitHub [7].

Under the hood, our tool combines the user inputs and computer vision to construct an instance of the seat assignment problem. Once an initial classroom diagram is uploaded, the user first highlights the image's scale by clicking its corners, allowing the tool to calculate distances within the room. The user can then designate the different chair types within the room in a similar manner (see **Figure 3B**, which shows the result of a user highlighting the scale and three distinct chair types in the room). Once each chair type has been highlighted, the user can trace out the border of each chair type (allowing the program to identify the center and boundary of each chair type for use in identifying conflicting pairs). Then the user clicks to run chair recognition on the room (see the menu in **Figure 3A**). Chair recognition is completely handled by the tool and relies on Opencv2's computer vision capabilities [9]. (See [8] for detailed treatment and reference on computer vision algorithms.) As chair recognition completes most of the remaining input (see the results of chair recognition in **Figure 3C**), all that is left for the user is to confirm the results and make any necessary adjustments (e.g., clicking to add an extra chair). Altogether, these features allow the user to spend as little as five minutes processing a classroom.

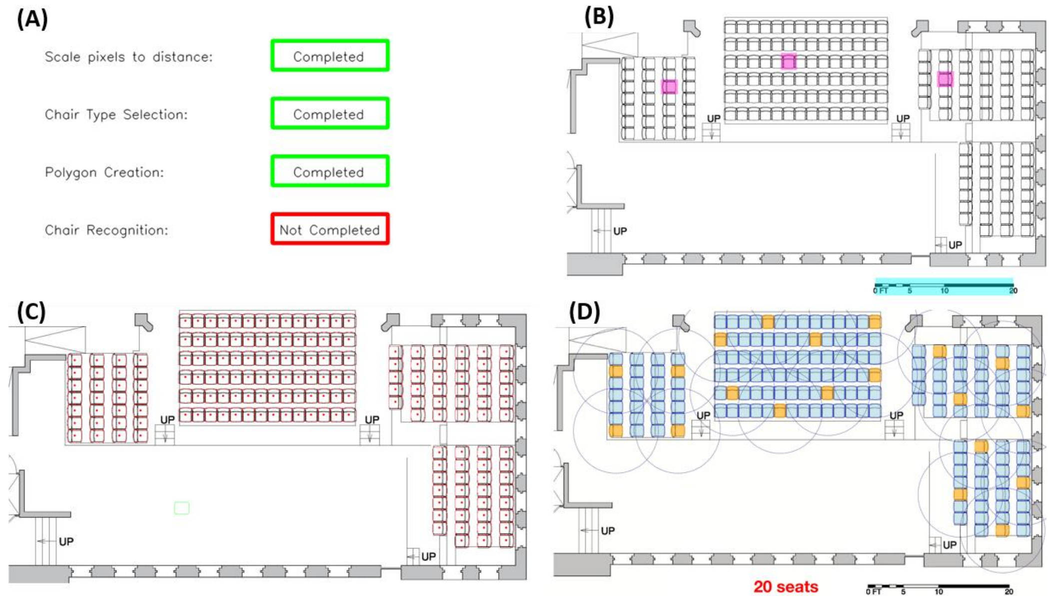

Figure 3: (A) The menu after the first three input methods have been completed. **(B)** The total input from the first two steps. **(C)** The results from chair recognition. **(D)** The output seating plan showing an optimal assignment.

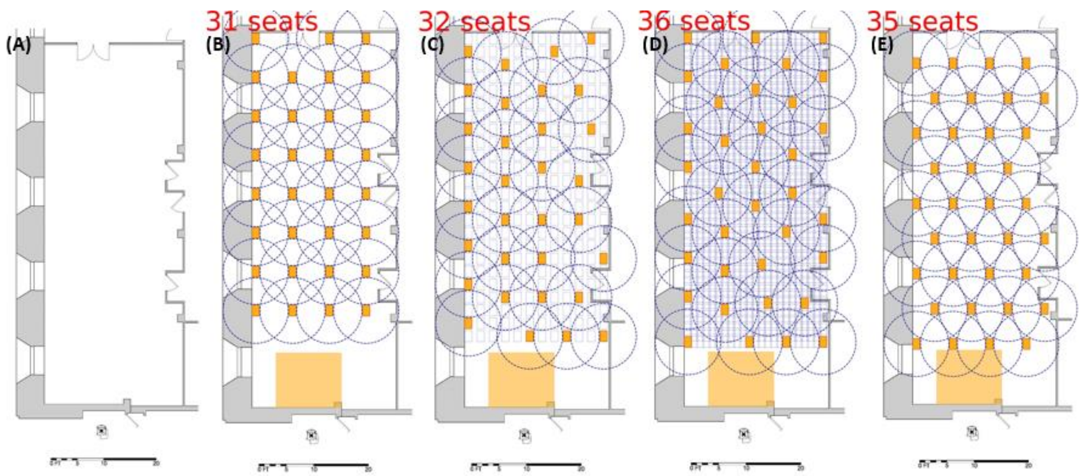

Figure 4: (A) An empty room's raw diagram. **(B)-(D)** Solutions from grids with increasing density of possible chair locations. **(E**) Solution from an intelligent hexagonal pattern.

With these inputs defined, the tool finds an optimal seating assignment at the click of a button. Our tool converts the inputted information to a graph by creating a node for each chair in the diagram, then comparing the distances between chairs to create the edge set (the set of conflicting pairs $\mathcal{C}$). Our tool then solves the integer program (1) using the solver OR-tools [10] to obtain an optimal seating assignment. Finally, the tool creates a user-friendly representation of the seating assignment, which can be shared with anyone involved in planning the classroom (see **Figure 3D**).

Our tool has several additional features. The user can add and remove chairs (e.g., to incorporate additional possible seat locations in the room). Second, in classrooms with non-fixed seating, the tool allows the user to input large numbers of possible chair locations using intelligent grid and hexagonal patterns (see **Figure 4**; **Figures 4B-4D** show the improvements from using an increasingly-dense grid pattern, and **Figure 4E** shows the results of a hexagonal pattern). The user can further define areas that cannot accommodate seating (e.g., to prevent seats from being placed too close to an instructor). Finally, our tool is capable of sensitivity analysis to help identify rooms where precise chair locations and measurements are particularly important.

4. Computational Results

After implementing the tool, we worked with the Office of the University Architect at Cornell University to create seating plans for classrooms. Cornell has a wide range of classroom types, including both small classrooms and large lecture halls. First, we applied the tool to 191 classrooms where the architect's office had already drawn initial socially-distanced seating plans by hand; these plans provide a benchmark for our tool. The distribution of pre-COVID capacities of these 191 classrooms is right-skewed (**Figure 5A**), with a respective mean and median of 99 seats and 49 seats. Our tool found improvements over the university architect's initial socially-distanced seating assignments in 96 of these 191 rooms, yielding a total of 422 possible extra seats. The majority of these rooms (72 of 96) saw a potential improvement of at least 2 seats. Although the architect's office could not use all of the additional seats found (e.g., accounting for factors like seat accessibility), the architect's office increased seating capacities across drafted classrooms by 152 seats based on our tool; the majority of those seats (129 of 152) were explicitly identified by our tool. Each added seat meant that the classroom could accommodate an extra in-person student in every class block.

Our results also provide useful data for universities looking to estimate how much their classroom capacity is decreased by social distancing. Using the tool, the average optimal room capacity under social distancing reduces to 23 seats. Optimally socially-distanced rooms have an average capacity of about 23% of their pre-COVID capacity. As with pre-COVID capacities, the socially-distanced capacities are right-skewed as shown in **Figure 5B**. Note that the histograms in **Figure 5A** and **Figure 5B** exclude 24 large rooms with pre-COVID capacities between 202 and 1010.

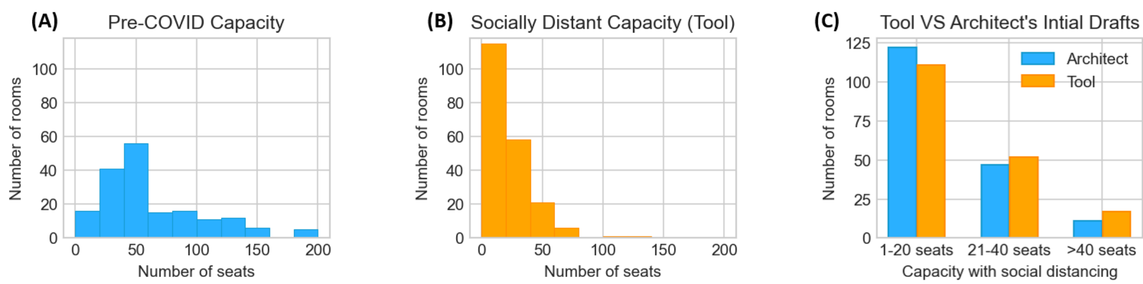

Figure 5: (A) The distribution of pre-COVID capacities of classrooms where the architect's office provided an initial layout. **(B)** The distribution of socially-distanced capacities after using our tool. **(C)** Socially-distanced capacities from our tool versus from initial architectural drafts.

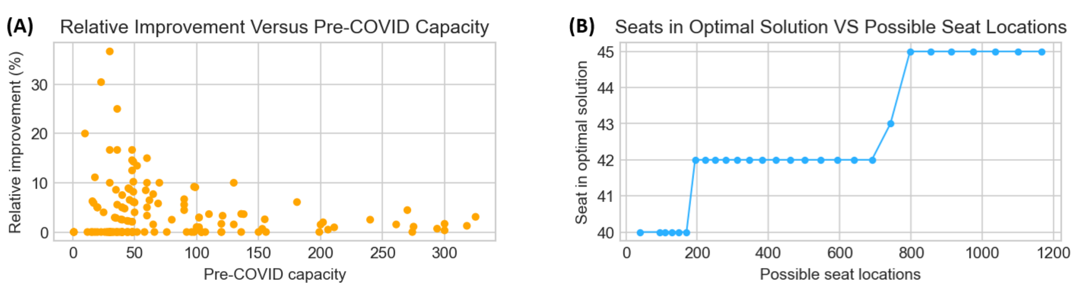

Figure 6: (A) A plot comparing the relative percent improvement of our tool's solutions by pre-COVID classroom capacity. **(B)** A plot comparing graph instance sizes to solution capacity for a room.

Figure 5C and **Figure 6A** showcase the performance of our tool relative to the architect's initial seating assignments. **Figure 5C** shows the distribution of socially-distanced seating capacities produced by our tool, juxtaposed with the distribution of capacities of the architect's initial assignments. Notice that seating assignments produced by our tool give rise to fewer small classrooms (socially-distanced capacity of less than 20) and to more large classrooms compared to the architect's initial assignments, indicating improvement in capacities. Next, we define a room's *relative improvement* as the difference between the tool-generated capacity and the initial socially-distanced architect's capacity, divided by the room's pre-COVID capacity. This relative improvement is plotted against pre-COVID capacity in **Figure 6A**. The relative increase was particularly large for medium-sized classrooms (those holding 20-50 seats pre-COVID), where our tool often found improvements over the architect's socially-distanced capacities of 10% or more. Thus the tool provided the largest improvements on rooms in the most common size range.

As noted in Section 3, our tool also allowed us to find good seating plans for classrooms with non-fixed seating by using several heuristics to place potential seats across the room. Using these features, we were able to help the architect's office generate socially-distanced capacity improvements beyond those summarized above (our previous analysis is restricted to classrooms where we had a previously-drawn socially-distanced seating plan as a benchmark). For example, we made the first socially-distanced seating layouts for 22 classrooms and assisted in breaking up a large auditorium into several smaller seating sections. Across these additional drafts, and working closely with the architect, we were further able to provide a total of 1388 possible seats for in person learning.

When looking at the classrooms, we considered the trade-off between the number of possible seat locations and the size of the resulting seat assignment. Guaranteeing an optimal solution would require an arbitrarily large – and computationally infeasible – set of possible chair locations. To empirically understand the relationship between the number of possible chair locations and the size of an optimal solution to the integer program, we started with a small set of possible chair locations. We then gradually increased the number of possible chair locations (see **Figures 4B-4D**) and solved the resulting integer program. The results of this process are shown in **Figure 6B** for an empty classroom where 40 seats could be placed in a grid (with chairs exactly 6 feet apart). **Figure 6B** shows how the optimal solution depended on input size as we gradually increased the possible chair locations in each row and column; following this process allowed us to add five extra seats.

5. Conclusions

Our paper has three main contributions. First, we modeled the socially-distanced seat assignment problem as a maximum independent set problem, allowing us to write an integer program formulation. Second, we developed an easy-to-use tool that allows a non-technical user to quickly build and solve the socially-distanced seat assignment problem for a given classroom. For classrooms with fixed seating, our tool provides an optimal solution to this seat assignment problem. Finally, we used this tool in conjunction with Cornell University, identifying over 400 potential extra seats. The COVID-19 pandemic has forced social distancing constraints in many different settings including restaurants and entertainment venues, and future work could apply our tool to these settings. On a practical level, our tool is particularly useful in large venues with fixed seating. In these settings, our tool can be used to find optimal assignments, generally finding solutions both more quickly and with more usable seats than an expert. Beyond classrooms, venues here it might be particularly useful include movie theater or stadium sections.

Acknowledgements

The authors gratefully thank the Office of the University Architect for helping develop the tool and providing classroom plans, and thank EY for funding student work. We also thank the referees for their valuable feedback.

References

[1] R. Bahl, N. Eikmeier, A. Fraser, M. Junge, F. Keesing, K. Nakahata, and L. Z. Wang, “Modeling covid-19 spread in small colleges,” *arXiv preprint*, arXiv:2008.09597, 2020.

[2] J. M. Cashore, N. Duan, A. Janmohamed, J. Wan, Y. Zhang, S. Henderson, D. Shmoys, and P. Frazier, “Covid-19 mathematical modeling for Cornell’s fall semester,” 2020. Available: https://cpb-us-w2.wpmucdn.com/sites.coecis.cornell.edu/dist/3/341/files/2020/10/COVID_19_Modeling_Jun15-VD.pdf.

[3] K. A. Weeden and B. Cornwell, “The small-world network of college classes: implications for epidemic spread on a university campus,” *Sociological Science*, vol. 7, pp. 222-241, 2020.

[4] L. Burke, “Cornell researchers say in-person semester for university safer than online one,” Inside Higher Ed, July 1, 2020. Available: https://www.insidehighered.com/news/2020/07/01/cornell-researchers-say-person-semester-university-safer-online-one.

[5] M. Conforti, G. Cornuejols, and G. Zambelli, *Integer Programming*. Berlin: Springer, 2014.

[6] R. E. Tarjan and A. E. Trojanowski, “Finding a maximum independent set,” *SIAM Journal on Computing*, vol. 6, no. 3, pp. 537-546, 1977.

[7] K. K. Greenberg, T. Hensel, Q. Zhu, S. Aarts, S. C. Gutekunst, and D. B. Shmoys, “Classroom seating planning tools.” https://github.com/KylekG/Classroom-Seating-Planning-Tools, 2021.

[8] R. Szeliski, *Computer Vision: Algorithms and Applications*. London: Springer, 2011.

[9] OpenCV, *Open Source Computer Vision Library*. 2015.

[10] L. Perron and V. Furnon, *OR-Tools*. Google, 2019. Available: https://developers.google.com/optimization/.