

LupSeat – A Smart Seat Assignment Generator

Hiroya Gojo, Joël Porquet-Lupine
University of California, Davis. Dept of Computer Science

Introduction

When students are allowed to sit next to whoever they want in an exam, it may increase the chance of plagiarism and cheating. On the other hand, assigning random seats to students prior to exams is time-consuming, especially for large classes, since instructors have to assign seats by hand. There are also risks of clerical errors, such as forgetting to assign a student. Existing software solutions only focus on relatively small classrooms. The goal of **LupSeat** is to automate the seat assignment process in an easy way to solve this problem. Using an algorithmic approach simplifies the process of spacing out students and minimizes the chance of errors. **LupSeat** is easy for instructors to use and instructors only need to provide student information and the representation of the room layout.

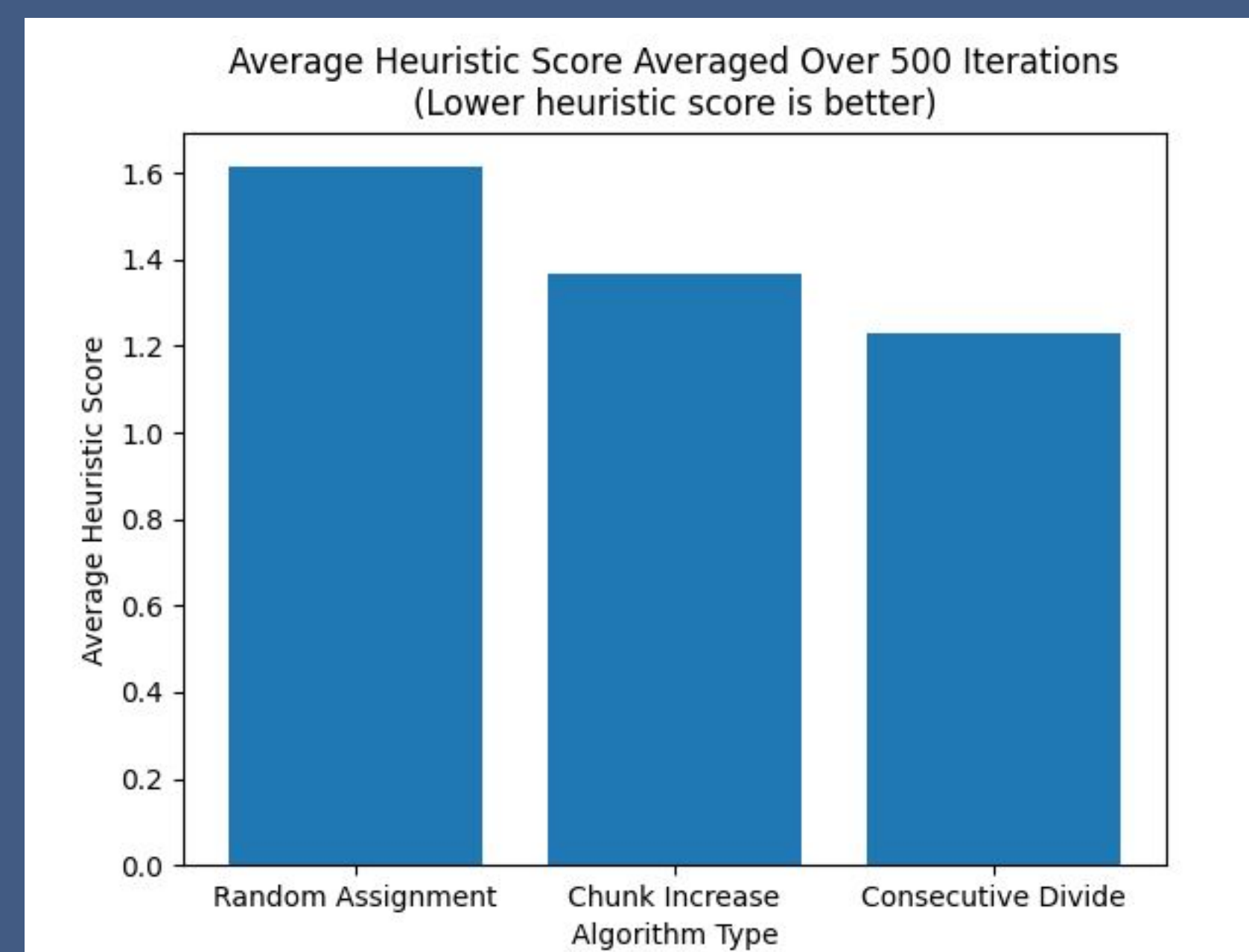
Graphical Interface

The screenshot shows the LupSeat GUI with the following sections:

- Input Settings:** Fields for "Set student file" and "Set room layout file".
- Output Settings:** "Set output directory" field.
- CSV chart name:** "chart.csv".
- Graphic chart name/size:** "chart.pdf" and "a4".
- Graphic room name/size:** "room.jpg" and "a4 flip".
- Advanced Settings:** "Format String" field with "(sid)".
- Sort Output By:** Radio button for "seat".
- Seed:** "1613858956" and "Generate new seed" button.
- Choose the algorithm:** Radio button for "consecdivide".
- Run LupSeat** button.

LupSeat only requires the user to specify the list of students and the layout of the room. The graphical interface, built with Tkinter, is an easy way to run **LupSeat** without having to navigate the command line. Both the GUI and command line interface provide various customizable options to tune the output exactly as needed.

Algorithms and Methodology



While developing **LupSeat**, two fundamentally different algorithms were compared, which I will call chunk increase algorithm and consecutive divide algorithm. The former is a bottom up algorithm, with the latter is a top down algorithm. The goal of the algorithms employed are essentially to minimize the size of the chunk of students so that the average number of students sitting next to each other is minimized.

The intuition with the chunk increase algorithm is to first start off with a chunk size of 1, that is, all students would have two empty seats around them. If all the students do not fit in the room with this method, the chunk size is increased by 1, and continues until all students fit in the room.

The intuition behind the consecutive divide algorithm is that the room initially starts out with the largest chunks possible, and for each empty seat, the current biggest chunk is split into smaller chunks. A backtracking algorithm was used where if the chunk needs to be split, it would first be patched together into the original large chunk, then divided again evenly.

By creating a heuristic score (average size of chunk in a room) to rank a room layout and running a script to run each algorithm many times, the consecutive divide algorithm was shown to consistently outperform the others.

Output Seating Chart

LupSeat is very customizable in its output. Several files are created including a graphical room layout image, a csv file with all the student/seat information, and the graphical student to seat assignment image. The image sizes are able to modified according to standard paper sizes, and the order of the students can be sorted based on their name, ID, or by their seat position. A full description of **LupSeat**'s options are available in the documentation at <https://gitlab.com/luplab/lupseat>.

Seating Chart (Seed: 1613858957)

A12 76275	C5 16164	E19 41074	F6 37602	H12 93510
A13 77162	C7 5052	E2 26262	F7 50646	H13 821
B10 18918	C8 62973	E21 3345	F9 94672	H15 79461
B12 89254	D10 16590	E22 75158	G1 50053	H17 88348
B14 41064	D11 9360	E4 92810	G10 4829	H18 92322
B18 54451	D13 78438	E5 58288	G12 6266	H19 44727
B19 27813	D15 46993	E7 75580	G13 30904	H2 53465
B2 98593	D18 87633	E8 40633	G15 26456	H21 41810
B21 97936	D19 72406	F1 2218	G17 15007	H22 87017
B3 62410	D2 60454	F10 19886	G18 51585	H4 27612
B5 81579	D21 82205	F12 87738	G19 63833	H5 9780
B8 13220	D3 17007	F13 80865	G2 79483	H6 49478
C10 94353	D5 95308	F15 81193	G21 57383	H7 12761
C11 11892	D7 58853	F16 88614	G22 70509	H8 21286
C13 378	D8 90628	F18 85611	G4 9675	J1 63441
C15 82559	E1 78257	F19 2794	G5 64579	J10 65995
C18 99383	E10 9736	F2 16490	G6 92291	J12 6871
C19 10917	E12 53614	F21 78624	G7 85689	J13 12247
C2 79738	E14 94962	F22 3072	G9 66500	J15 67684
C21 29246	E16 5849	F4 88089	H1 94546	J17 35357
C3 95756	E18 92844	F5 10013	H10 73432	J18 13946

Seating Chart

