Modelling socially awkward crowd in a subway wagon

FdV Bachelor — Modelling Project Week

Group 3

Nine Collomb, Kiara Mangin,Jules Herrmann
CRI - UNIVERSITY OF PARIS

March 4th, 2022

Final report

Introduction

Context and rationale of the project

We were brought together by our mutual interest in modelling the crowd dynamics in a subway wagon. This is a situation we all experience daily while commuting between our homes and the CRI. And you will see, after reading this report, you will (over)analyze metro users' behaviour.

Our first idea was to focus on the individual behaviour of each subway user and their interactions in a wagon. There are many interesting dynamics : people try to avoid each other while being attracted by comfortable positions, like seats or metal poles. Different parameters could influence these dynamics from the number of available seats to the crowdedness of the wagon.

We had an other idea regarding the flux of people travelling from one station to another and how the creation of a new metro line would impact this flux. However, we were more interested in the individual behaviour of metro users rather than an overall flow over a large network of stations. Our goal was to design a subway wagon where agent were able to enter, move and get out. We wanted to get closer to the reality. In order to have an interesting model, we needed to have a problematic that our model would be able to answer.

Problematic

A subway wagon architecture is something that is not fixed. In Paris, from metro line 1 to metro line 13, seats and metal bars are not always in the same place. We were interested in being able to recreate this diversity. Moreover, we have all experienced a crowed versus non crowed subway trip. This has a very important impact our experience of the subway ride. Thus, geometry of the wagon and number of agents were things that we wanted to make vary. In our problematic we wanted to measure something that would be linked to both, so we could experiment by changing the number of seats and people.

A key element to understand the behaviour of metro users is the presence of attractive objects (i.e. seats) or repulsive objects (other metro users). This made us reflect on a larger concept of the users' comfort. We defined comfort as a value of the cell the agents are on. This comfort depends on whether or nor there are people too close to them and if they managed

to reach a seat. Our goal was to model the dynamics of agents evolving in a subway wagon to study the impact of the geometry of subway (position of seats and metal pole) on the flow and the final distribution of agents as well as the number of users that were able to find a comfortable place.
A clear problematic we came up with was : How does the number of seats and agents impact the comfort of the users over time ?The model we produced was rather the one of a socially awkward crowd, but that is something you'll find out in our results section...

Model

We chose to use a cellular automata based model. After some discussion, we found that it was the best way to implement our idea of the model and enable us to obtain the result we were expecting, in the amount of time we were given. In addition, there is an extensive literature on using cellular automaton for crowd simulation. See for example Sarmady et al., 2014, Klüpfel, 2003, Zhou et al., 2010 or Fu et al., 2014.
This model takes into account the subway wagon design, the attractiveness of seats and metal poles as favourite spots for metro users, the number of metro users in order to compute the location of the users over time and their level of comfort (based on whether or not they reached an attractive point or if they're at sufficient distance from other users). This will help us optimize the wagon's configuration as we experiment.

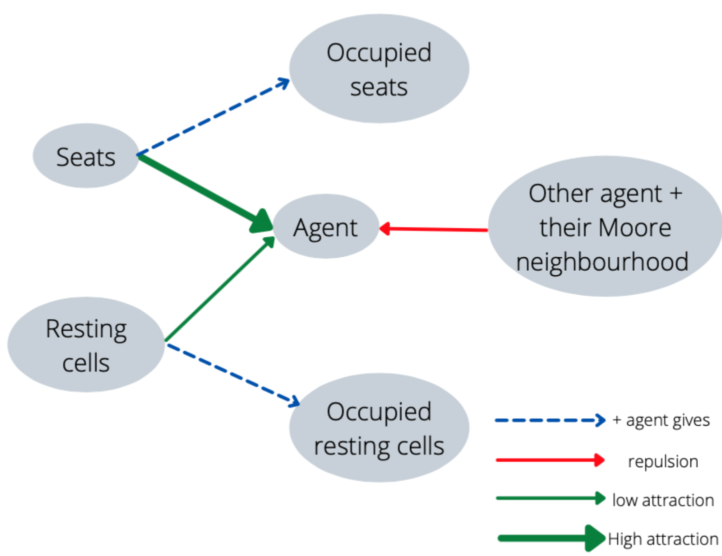


Figure 1: Interactions between the different components of our model

In the final model we used for our results, we decided not to include metal bars with their resting cells (the cells surrounding the metal bar, which are supposed to be more attractive than the rest of the wagon, but less attractive than the seats). Making our model work with only one type of attractive cells, the seats, was already challenging. However, implementing metal bars is a good perspective to get closer to reality.

Our cellular automaton consists of several types of cell.

- Agent (A) cells represent the metro users, they repel each other.
- Wall (W) cells represent obstacles.
- Resting cells (R) are the metro seats, they attract A.
- Cells can also be empty (E).

Dependent variables

- Location of agents over time, for a more qualitative analysis
- Total comfort function. We will define a comfort function that associates to a state of our system a single value corresponding to the total comfort of the agents, according to how their wish to sit and to be away from other agents is fulfilled.

Independent variables / parameters

We can consider multiple parameters to test. We will test them in an exploratory phase of our experiments and then choose which one we want to test systematically.

- Subway wagon disposition (seat and metal bar placement, wagon size).
- Number of people.
- How they enter the wagon.

One important part of our model was the notion of comfort. Indeed, here we defined the comfort of an agent as the amount of space they have around them and if they are on an attractive spot (seats or metal bars). Thus every cell has a comfort value and every agent is lowering cells' comfort in its Moore neighbourhood. Thanks to the matrix we computed in our code, agent were able to spot the most comfortable cell in the wagon and compute their next position in order to find the fastest way to their new goal. Here is an example of a wagon we designed. It shows what we just explained, seats have a higher comfort value compared to resting cells around metal bars. Moreover, the agent has a negative impact on its environment, thus, another agent will not want to get close to him.

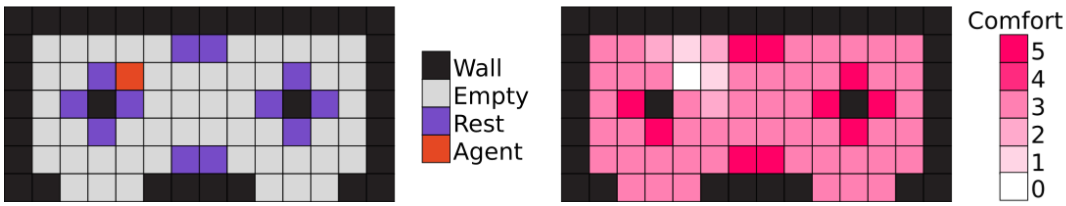
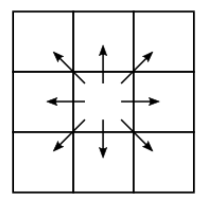


Figure 2: A grid representing a wagon with an agent and its comfort matrix

Dynamics

Let A be the set of agents and A_i be the agent i of position $p_i^t = (x_i, y_i)$ at time t.
Let the matrix G of size $n \times m$ with values in $\{A, W, R, E\}$ represent the state of our model at a time t.

The model is based on agent reaching a position of highest comfort $C_{x,y}$. At each iteration of our model, agents consider the distance to $C_{x,y}$ in the 8 cells of their Moore neighbourhood. Then, they move to the cell that has the smallest distance to the chosen comfortable position.



Moore neighbourhood

At each iteration, 4 steps are necessary :

1. We compute the comfort matrix C of size $n \times m$ such that $C_{x,y}$ is the comfort of the cell (x, y). $C = \sum_i F_i$, where F_i is the comfort field of the cell i, computed as follows: Consider the cell i of coordinates (x_i, y_i).

   - If $i \in A$ is an agent of position x_i, y_i.
     Let $\Omega_i = \{(x_i + a, y_i + b) \mid (a, b) \in [\![-1; 1]\!]\}$ be the Moore Neighborhood of i. If $(x, y) \in \Omega_i$, F_{xy} takes the value of the following matrix M_A centred on (x_i, y_i).

$$M_A = \begin{bmatrix} K_{A3} & K_{A2} & K_{A3} \\ K_{A2} & K_{A1} & K_{A2} \\ K_{A3} & K_{A2} & K_{A3} \end{bmatrix} \text{ With } K_{A1} < K_{A2} < K_{A3} < 0$$

Else, $F_{xy} = 0$

- If $i \in R$ is a rest cell of position (x_i, y_i).

$$F_{xy} = \begin{cases} K_r & \text{if } x, y = x_i, y_i \\ 0 & \text{else} \end{cases} \text{ with } K_r < 0$$

2. Each agent i find its goal g_i as the cells of maximal comfort, using distance between the agent and the cell as a tie-breaker. $g_i = \underset{x,y}{argmax}(C_{xy}, d(x, y))$

3. Each agent i find its wanted new position $p_i^?$.
We have Ω_i be the Moore Neighborhood of i and M the set of cell agents can step on.
At each step, we define the set of possible positions as $P = \Omega_i \cap M$.
We then find the wanted new position of the agent A_i : $p_i^? = \underset{p \in P}{argmin}\, d(p, i)$, with $d(p, i)$ the Manhattan distance between p and i.

4. In case of conflict (ie. $\exists\, A_i, A_j$ such that $p_i^? = p_j^?$), we choose randomly and with equiprobability which agent is able to have its wanted position as its next position.

Total comfort

For the purpose of experiment, we define the total comfort $C_{tot} = \sum_i^A C_{x_i,y_i}$.
This value has a flaw in the sense that you can't compare it for two situation with a different number of agents : no matter the comfort of the system, if there are more agents, the value will get bigger (not necessarily in the positive direction).
To deal with this situation, we should have normalised this value to an average comfort $\hat{C} = \frac{1}{n}\sum_i^A C_{x_i,y_i}$, with n the number of agent.

Expected dynamics

Those dynamics should prevent the subway user to fall into local optimums, because it has an overview of the entire wagon. We should be able to define rest cells with a lower constant K_r and agents should only consider them when all the seats are used or at least next to another agent that decrease its comfort.

In the example below of expected dynamics, the rest cells grouped by two represent seats and have a high constant K_r whereas the two rest cells in the centre of the wagon are metal pole with a lowered constant K_r.
Two agents enter alternately the wagon, the first one will choose the closest seat. As a result of the negative comfort field of this agent, the comfort of the seat just next to it decreases, causing the second agent to enter a further away seat.

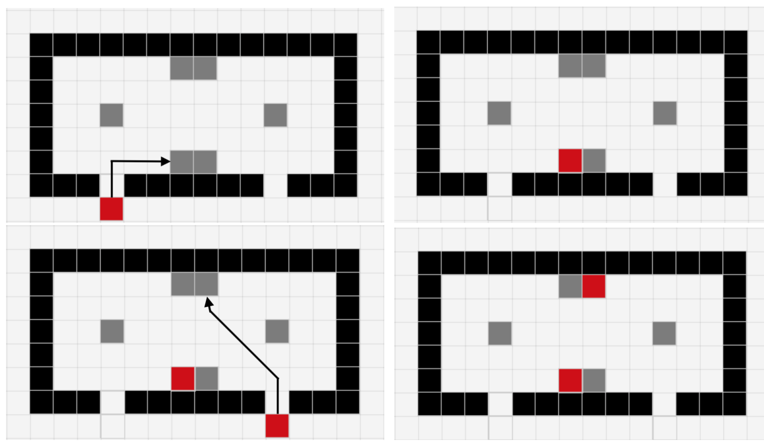


Figure 3: Expected dynamics when two agents enter the wagon. Agents prefer seats than closer metal pole. Reads from left to right then up to down.

Implementation

Architecture

In comparison to other projects during this week, ours was less centered around equations but particularly coding heavy.
To make everything more tidy and ease collaborative coding, we chose to create a base object-oriented architecture for our project (see figure 4).

Data structure

One challenge we had was the data structure to store the state of our system.
On one hand, we sometimes want to know in an efficient way if there is a block at a specific position, which can be done with a 2d array of the size of the wagon. On the other hand, we

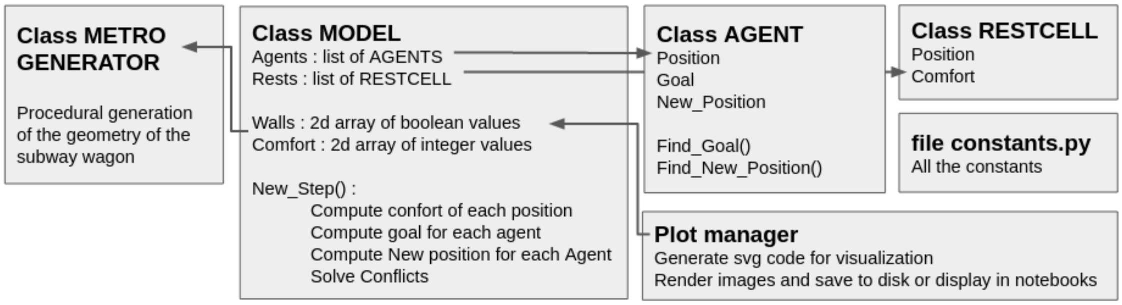


Figure 4: Architecture of our implementation of our model.

often want to iterate over a specific set of cell (for example all agents), which is much faster if we keep a list of all agents and their position.
We chose to do a mix of those two possibilities : we keep a lists containing all agents, which are objects that store their position by themselves. We store the rest cells in a similar way. However, the comfort is defined for each cell, which is why we need to store it in an array. Walls are also stored in an array, because if we never need to iterate over all walls, we often need to know if there's a wall at a specific position.

Visualisation

Having some sort of graphical representation of the state of our system was really important for debugging. On one hand, we really wanted our code to integrate neatly in python notebooks, because those environment are perfect to explore and discuss the results of our model, one the other hand, we needed the ability to render the visualisation into a video to share our results at the end of the week.
Our class PLOT MANAGER generates code for vector image in SVG format which can then either be rendered by the browser in a jupyter notebook or rendered to an image thanks to the package HTML 2 IMAGE (*html2image* 2022).

Results

Constants

For the following simulation, we set :

- $M_A = \begin{bmatrix} -2 & -3 & -2 \\ -3 & -5 & -3 \\ -2 & -3 & -2 \end{bmatrix}$
- $K_r = 5$

M_A being the negative impact of an agent on his Moore neighbourhood and K_r the comfort value associated to our seats to make them attractive.
We were able to simulate our model reliably for a wide variety of initial conditions, and to record the total comfort of the agents as the sum of each agent's cell comfort over time.

Simulations

Simulation A : Video 1
Simulation B : Video 2

We decided on a wagon's design and kept it for the entire length of our experimenting. Its a 6×12 grid. As for the seats, we decided to study two difference scenarios. Our problematic was to see how the number of seats and the number of users impacted the overall comfort of the agents over time. To do so, in one wagon configuration we had 10 seats for 10 agents and in the other 4 seats for 10 agents. Our comfort function collected the values of the total comfort of the users over time. The total comfort is the sum of the comfort value of each user depending on whether they are not too close to others and if they are on a seat. We plotted this function for each simulation and came to the obvious conclusion that when there are less seats than agents, the total comfort value decreases.

When there are more seats, one can also notice that it takes more iterations to find an equilibrium. This is due to the fact that as long as there are free seats left, seated agents while still change their goals because they have become more attracted to those seats. It takes more time for everyone to be happy with their position, because there are much more

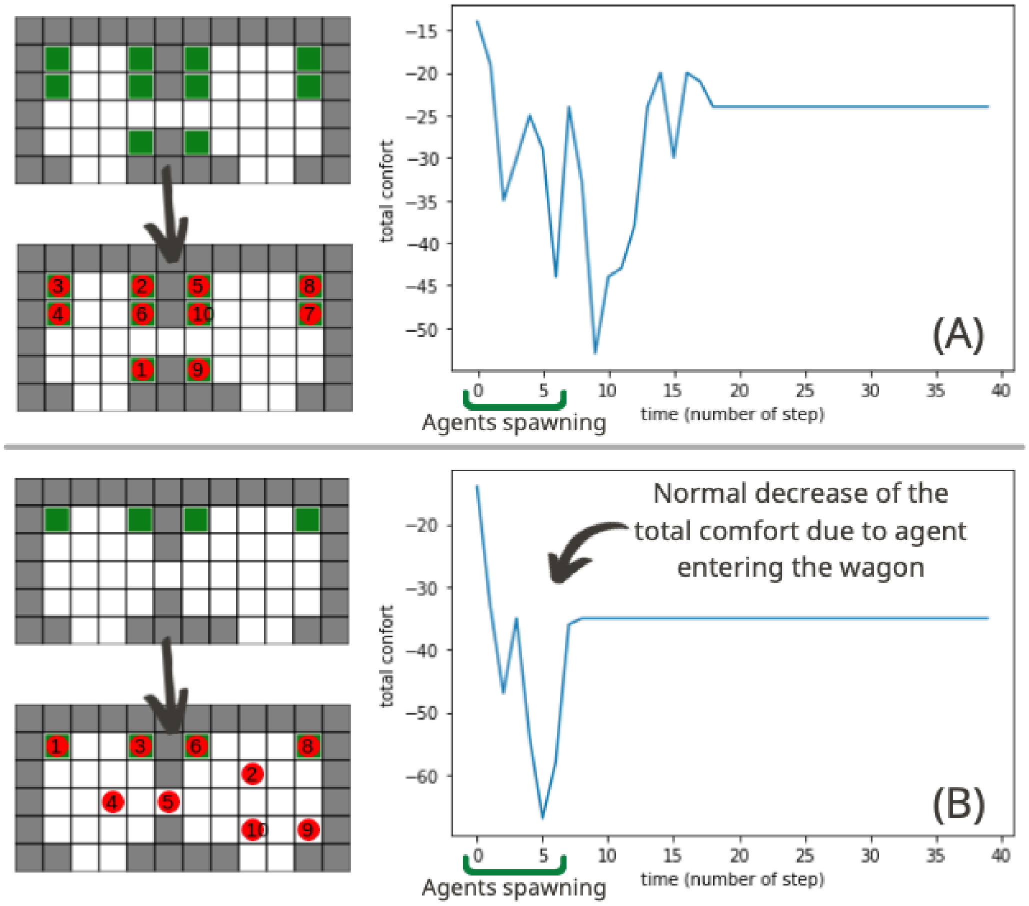


Figure 5: Total comfort of the 10 agents over time for a metro wagon with (A) 10 seats and (B) 4 seats. Agents enter the wagon gradually during the 6 first iterations

attractive options and they are very sensitive to the presence of other agents around them.

Figure 6 is an example of a simple simulation. Two agents are entering through the left door. They will have the same goal until frame 4. Then agent 2 reduces the attractiveness of the seat for agent 1. He finds a new goal : the closest free seat. But now (frame 5) he is very close to agent 2, so the attractiveness of the seat he's on decreases. This is why in frame 6 both agent 2 and 1 move towards the closest seat with the highest comfort value. In frame 7, now that agent 1 is far away, agent 2 will prefer going back to his first seat.

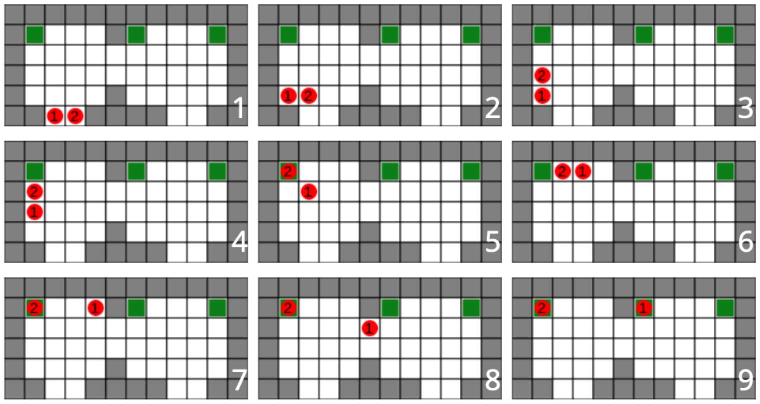


Figure 6: Example of dynamics with two agents for 3 seats.

Discussion and conclusion

We succeeded in modelling a subway wagon with seats and agents moving around. In some simulations we manage to obtain the stable state that we wanted where nobody is moving anymore.

We introduced randomness to solve conflict and it has an important impact on our simulation. In fact, the same initial condition will never lead to the same final distribution of agent. This is problematic in case of reproducibility. However, in reality, no wagon dynamic is the same, so it might not be such an issue. One just needs to run the simulation many times to obtain an average behaviour.

This model has a lot of perspectives due to numerous bug we couldn't fixed within the time limit.
Indeed, when an agent is seated, he still is influenced by the negative comfort value of the Moore neighbourhood of other users. When another agent is too close, the agent will leave his seat because another free seat has become more attractive. This can sometimes lead to infinite oscillations. We could fix his position once he reaches a seat, but that goes against a dynamical system.
As you can observe in the video 3 the same initial condition as the simulation A lead to an infinite oscillation :Video 1.

Moreover, agents can't spot if another agent has the same goal as them. Thus, they are wasting time on unreachable goals. They also are not aware of their close neighbourhood. They only focus on their goal and the nearest cell to the goal. But they are not intelligent pathfinders. They would get stuck behind a wall whose position is close to the goal and would not know how to move around it.

Overall, this project has been a great learning opportunity and the freedom we were given motivated us to produce something fun and nice.

Links

Our code can be found on our github page.

More information as well as our daily reports and poster can be found on our CRI project page.

References

Fu Y.-W., Liang J.-H., Liu Q.-P., and Hu X.-Q. (2014). "Crowd Simulation for Evacuation Behaviors Based on Multi-agent System and Cellular Automaton". *2014 International Conference on Virtual Reality and Visualization*. Shenyang: IEEE, pp. 103–109. DOI: 10.1109/ICVRV.2014.52. URL: https://ieeexplore.ieee.org/document/7281051/.

html2image (2022). *html2image: Package acting as a wrapper around the headless mode of existing web browsers to generate images from URLs and from HTML+CSS strings or files.* URL: https://github.com/vgalin/html2image.

Klüpfel H. (2003). "A Cellular automaton model for crowd movement and egress simulation".

Sarmady S., Haron F., and Talib A. Z. (2014). "Simulation of Pedestrian Movements Using Fine Grid Cellular Automata Model", p. 17.

Zhou S., Chen D., Cai W., Luo L., Low M. Y. H., Tian F., Tay V., Ong D., and Hamilton B. (2010). "Crowd modeling and simulation technologies". *ACM Transactions on Modeling and Computer Simulation* 20, p. 20. DOI: 10.1145/1842722.1842725.