Instituto Superior Técnico

Complex Networks

Schelling's Segregation Model

Group #98

Carlos Branco, 71003
Isaac Vargas, 78690

Lisbon, 24th November 2018

Introduction

Segregation is something that we can find in our everyday lives. Some of it is forced due to external pressures (economic for example), some of it is natural due our own nature of being close to those who we look up or aspire to become, some is caused by different languages and it may be one sided or reciprocal. Segregation can arise from different causes and manifest itself as pattern of separation which we will look into in the following report. Furthermore we will bring an updated approach from Schelling' s original work which is almost 50 years old.

Different factors and mechanisms can trigger separation or accentuate willingness for such a thing. Separation mechanism can be viewed as ratios that either are or not met either locally or globally, movement may be restricted by distance or completely free, when moving or during life agents can either try to integrate themselves or further segregate by joining the most likely to them. These same agents can have different demands in the conditions they consider to move into and from and tipping points can show up when combining all these agents.

Methodology

We started by creating a simulation of the model using Python. As explained in the original paper in the Area Distribution section, we have a matrix, where each entry is a "house", that house may be empty or occupied by an agent. If a house is empty ,in the visualization its location in the matrix is colored white (Figure 1), otherwise it's colored according to the agent's race. The agent distribution is generated randomly.

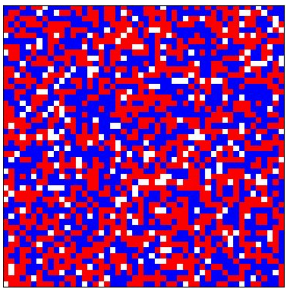


Figure 1: 50x50 size, 10% empty, two races, 50/50 distribution between races, 50% intolerance both races, racial behaviour for both agents. Initial state board.

All the parameters in figure 1 description are changeable. For example, you can have a 20x30 board, 50% empty, three races, 55/30/15% distribution between races, 0.2/0.3/0.4 intolerance levels for the races. Moreover there's an extra parameter, the number of iterations to run the simulation for. As some configurations don't converge to a state where it's possible for every agent to be happy (the end condition of a simulation) we need to have a parameter (default is 500 iterations) so that the simulation doesn't get stuck in an infinite loop.

Now with a initial state set we run the matrix, house by house. When we get to an occupied house we see if the agent is happy by comparing the agent against his neighborhood (according to some criteria), if the agent is unhappy we move him to an empty house. This process is repeated until every agent is happy or we hit the maximum number of iterations allowed.

Here we introduced two more concepts, neighborhood and happiness of the agent. Let's start by analyzing the neighborhood, the simpler concept.

Neighborhood

In our model the neighborhood are the immediate neighbors of an agent (Figure 2), only one level deep, being a 3x3 square around the agent for a total of eight neighbors. Although Schelling's paper mentions bigger neighborhoods, for example two levels deep, a 5x5 square, 24 neighbors a smaller neighbor function was chosen.

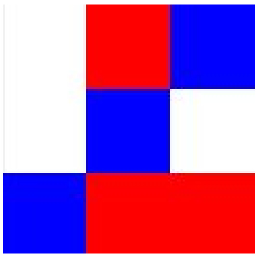


Figure 2: a 3x3 portion of the matrix showing an agent (center cell) and their neighbors (two other agents of the same race, three agents of a different race, and three empty houses).

Happiness (rank_diff function)

$$happiness = happy.with.neighbour / total.neighbours$$

If happiness ratio is greater than intolerance_threshold the agent is happy, same if the agent is isolated (total_neighbours = 0).

Agent happiness is defined as a ratio of neighbours similar to an agent's motivation. The agents motivation (behaviour) can vary among agents although three main behaviour groups were implemented. Firstly, we have racial agents, whose happiness calculation is based on the number of equal race agents in their neighbourhood. If this ratio is smaller than the intolerance threshold de agent assumes it is not happy with the surroundings and decides to move otherwise it stays in the same place. When moving the agents they choose the first available position to jump to.

Similarly two more behaviours were defined, economic based agents that attempt to simulate agents that look at the economic level of the surroundings in order to decide whether to move or not and academic based agents that tend to group with other academic agents not looking at race or wealth.

The aforementioned attributes are binary representing the presence or lack of attribute for that agent.

The model was created in such a way that attributes for agents and agent difference functions can be altered or added for more specific behaviours.

Segregation

And lastly the goal of this project, the segregation of the population.

Segregation is a measure that is calculated as the mean neighbors similarity of all agents. First we calculate the similarity of the each agent with their neighborhood. To calculate that we take an agent's neighborhood and sum the amount of people that the agent shares a similarity with (e.g. race, for the ones that have the racial behaviour) and then divide this sum by the size of their neighborhood. To obtain the population segregation level, we then sum all the agents' similarities and then divide by the total amount of agents to get the overall segregation of the population.

Results

The configuration used was what we defined as default values which are the following:

Parameter	Value
Width	100
Height	100
Empty Ratio	0.10
Number of Races	2
Agents Probability	[0.5, 1]
Intolerance Threshold	[0,5, 0.5]
Max. Number of Iterations	500

Table 1: Default parameters for the model.

Most parameters are obvious in what they mean but two may need further explanation.
Agent probability is a list with the cumulative probability of spawning an agent of that race. There are two races in this example, so a list of size two. Index 0 of the list is race *"0"*, that has 0.5 (value in the index position) representing 50% chance of spawning, index 1 is race *"1"*, that has 1 (value in the index position) - 0.5 (value at the previous index) = 0.5, 50% of spawning. The agents are split 50/50 by race.
The intolerance threshold follows the same principle of the agents probability, but it's not cumulative, each value is the intolerance threshold for the race with that index.

Running the model with these parameters we obtain the following two plots (Figure 3, 4).

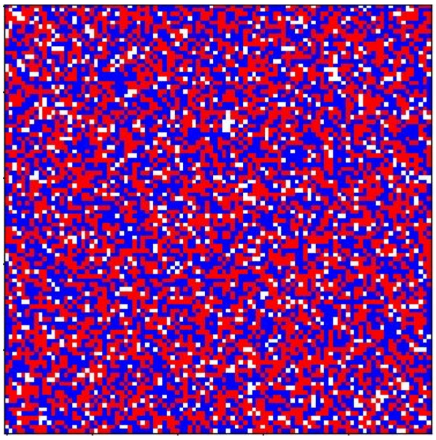


Figure 3: Initial state, 50.09% segregation.

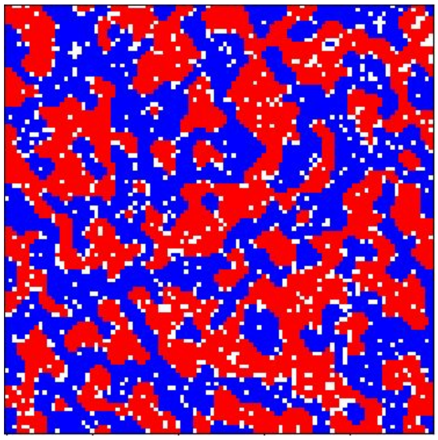


Figure 4: Final state, 87.04% segregation.

From the initial state, generated randomly, we compute that 58,63% of the agents were happy and there was a 50.09% of segregation on the board, at the end we got 100.0% happiness, which was the goal, but to achieve that goal the segregation rose to 87.04%. To achieve this it took 10 iterations from beginning to end. The initialization time was 0.428 seconds, the iterations took 0.093 seconds, for an average of 0.009 seconds per iteration. In total it was 1.906 seconds, this extra time comes from calculating the segregation, happiness, generating the two plots, etc.

In table 2 we can see the average time of running different sized boards for five times measured in seconds.

Board Sizes	50x50	100x100	150x150	200x200	250x250	300x300	350x350	400x400
Initialization Time	0.034	0.420	2.050	6.444	15.571	32.180	59.834	86.218
Iterations Time	0.260	1.273	3.179	5.421	9.229	14.366	21.712	22.972
Avg. Iteration Time	0.023	0.090	0.206	0.375	0.592	0.855	1.202	1.397
Total Time	0.690	2.251	6.049	13.014	26.430	48.692	85.189	112.457

Table 2: Average times, in seconds, for boards of different size with the rest of parameters as defaults.

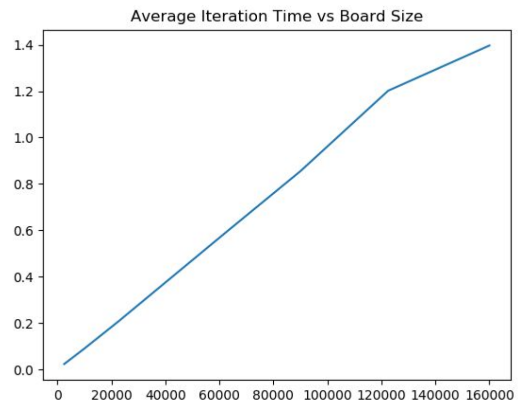


Figure 5: The average iteration time vs board size.

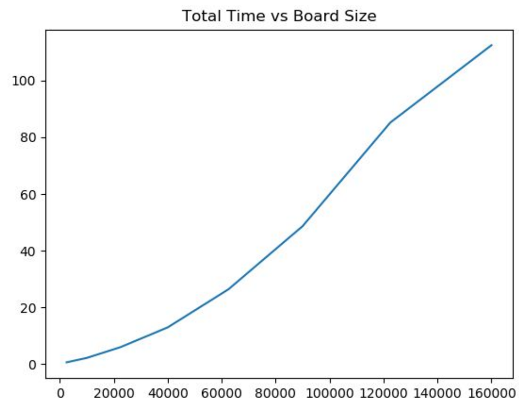


Figure 6: The total time vs board size.

After running all different sized boards while maintaining all the other configurations as default we can see that the time to execute our model seems to run close to linear time.

Due to this being a report about segregation and not performance of algorithms, let’s now vary some parameters that influence the segregation of the agents. What happens if we change the intolerance of both races from 0.5 to 0.3? Both to 0.7? One at 0.5, the other to 0.6? Let's see.

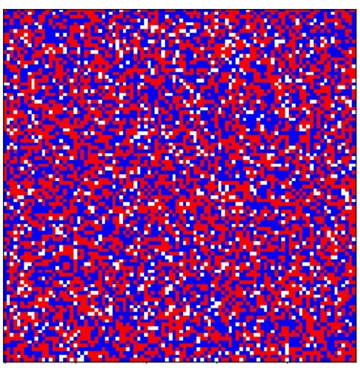


Figure 7: 49.76% segregation, 82.28% happiness.

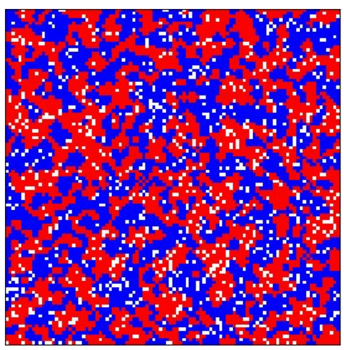


Figure 8: 74.58% segregation, 100% happiness.

Figure 7 and 8 are the initial and final states of a simulation with intolerance at 0.3 for both. With a 0.3 intolerance threshold we can already see the segregation effect beginning. Each agent doesn't need to have much intolerance for it to propagate from a micro to a macro level.

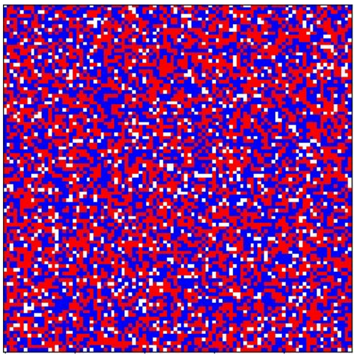


Figure 9: 50.36% segregation, 18.01% happiness.

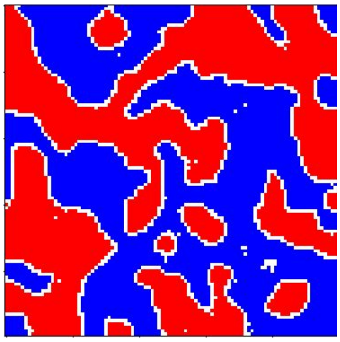


Figure 10: 99.16% segregation, 100% happiness.

Figure 9 and 10 have 2 races, with 50% chance each of spawning, with intolerance threshold at 0.7. At 0.7 threshold segregation is over 99%, society is effectively divided into two and there's negligible contact between races.

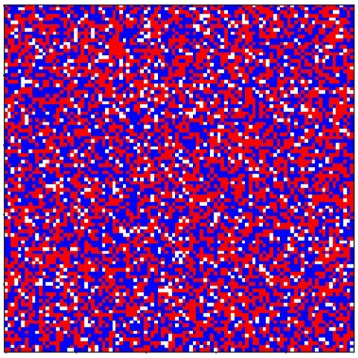


Figure 11: 49.97% segregation, 45.32% happiness.

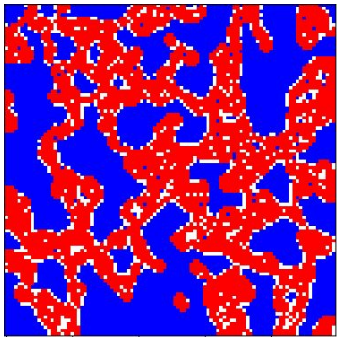


Figure 12: 91.76% segregation, 98.36% happiness.

In Figure 11 and 12 one race has their intolerance level at 0.5 and the other at 0.6.

Note the happiness level on figure 12, it's not 100% because we are running all these examples with the default parameters except for the intolerance thresholds and it exhausted the 500 iterations allowed. It ran 500 iterations with a 0.5/0.6 intolerance threshold and didn't achieve 100% happiness, while with both thresholds at 0.7, which is higher value than 0.5/0.6, it finishes in around 100 iterations achieving 100% happiness.

Let's see how segregation evolves according to the intolerance, all the values on the next two tables are an average of simulating each scenario five times. In all of them the end condition was 100% happiness, but for that to happen the last two scenarios were modified.

Intolerance	**0.05**	**0.10**	**0.15**	**0.20**	**0.25**	**0.30**	**0.35**	**0.40**	**0.45**
Segregation (%)	51.24	51.19	55.48	56.83	58.02	75.25	77.24	84.15	88.00

Table 3: Segregation by different levels of intolerance (0.05 to 0.45) (equal for both races).

Intolerance	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85*	0.90*
Segregation (%)	88.12	93.36	97.22	97.71	99.20	99.50	99.78	99.70	99.62

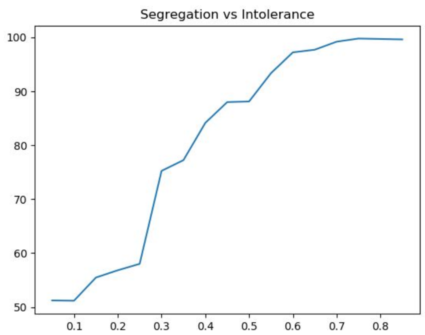
Table 4: Segregation by different levels of intolerance (0.50 to 0.90) (equal for both races).

From intolerance threshold 0.05 to 0.80, it was simulated with the default parameters. But in order to get good results and in a reasonable time the maximum iterations was modified.

For 0.85 intolerance it took 926 iterations (in comparison 0.80 took 110) and 101 seconds.

For 0.90 the empty ratio was changed from 0.10 to 0.20. It took 4139 iterations and 430 seconds. By changing the empty ratio, for 0.90 intolerance, from 0.10 to 0.20 results in a lot more empty houses and that influences the segregation calculation.

Because in our model we only have a total of 8 neighbors, intolerance levels above 0.875 (⅞) are all equal, the intolerance level of 0.90 is the same as 1.00. Since values over 0.80 are costly to simulate, we assumed that 0.90, 0.95 and 1.00 would be equal.



By plotting segregation levels against intolerance we can see two plateaus, under 0.1 and over 0.6 intolerance thresholds. The middle seems jagged because the data was obtained with 0.05 steps. If steps were to be obtained with a finer granularity the curve should be smoother.

Figure 13: Segregation percentage vs shared intolerance threshold by both races.

Conclusion

Even though segregation is still an issue we have to deal with in our everyday life. What we noticed throughout this project is that patterns for integration seem to be much more complex than patterns of separation. Moreover integrationists will face problems in order to meet the demands of the already present population. Furthermore, these integration/movement patterns need to be 'efficient' in such a way that members of the community can share minority neighbours and maintain a satisfiability equilibrium.

The process of movement and neighborhood compartmentalisation might also be further studied with fixed size and shape neighbourhoods and the impact of such constraints on the board and on the overall behaviours exhibited.

References

Schelling, T.C., *Dynamic models of segregation*. Journal of Mathematical Sociology, 1971. **1**(2): p. 143-186 Available from: http://norsemathology.org/longa/classes/stuff/DynamicModelsOfSegregation.pdf.

Hart, V. and N. Case. *Parable of the polygons: A playable post on the shape of society*. Available from: http://ncase.me/polygons/.