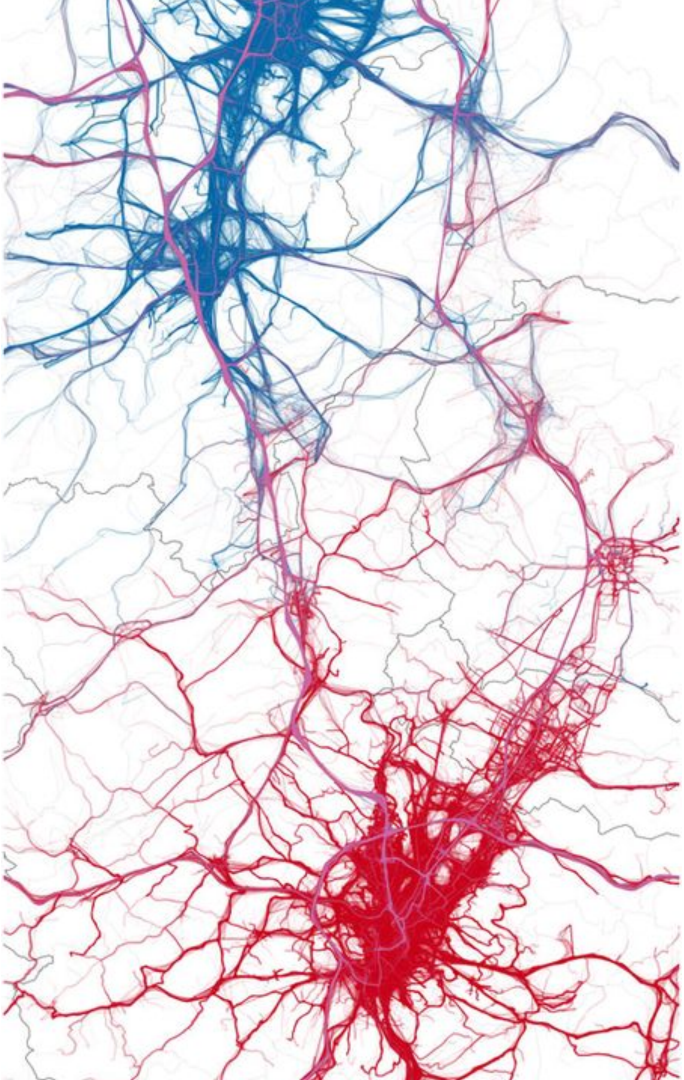


IN SUPREMÆ DIGNITATIS 1343

Consiglio Nazionale delle Ricerche

09 Segregation Models

Cities are Complex Systems

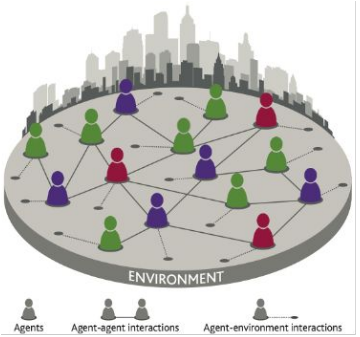
What happens?

- Traffic
- Pollution
- Epidemics
- Inequalities
  - Housing
  - Economic
  - Racial

Cities are Complex Systems

Can we model them?

Agent-Based Models (ABMs)



- **Pro**:
  - Detailed representation
  - Explainable
  - What-if tool
- **Cons**:
  - Curse of dimensionality
  - No predictions
  - Only simulative

Artificial Intelligence Models

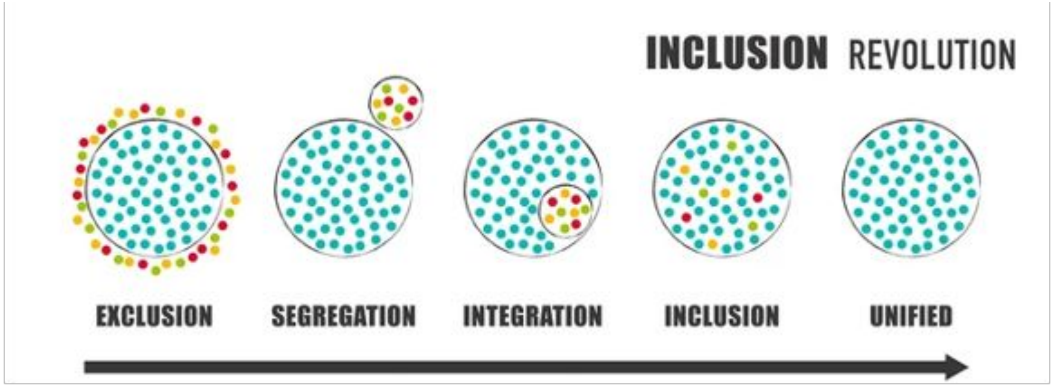
- **Pro**:
  - Accurate algorithms
  - Latent Knowledge
  - Data-driven
- **Cons**:
  - Hard to interpret
  - Hard to control
  - Performance based

Segregation

What's that?

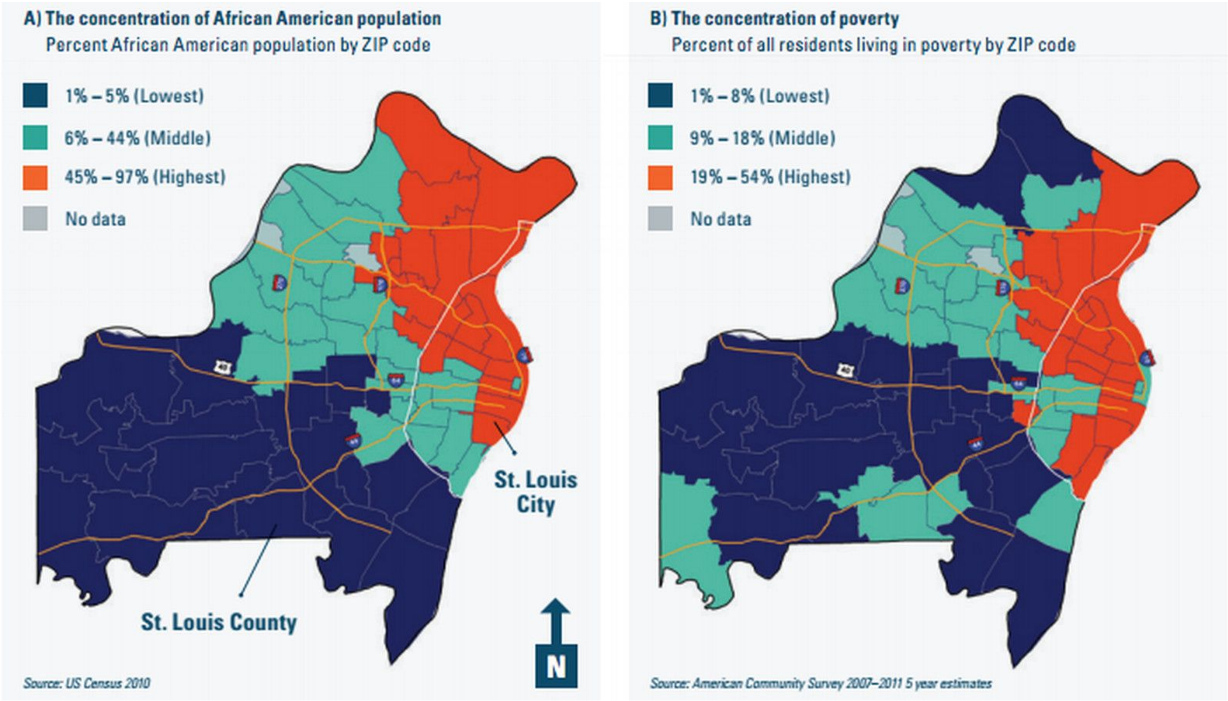
Segregation

- The act by which a (natural or legal) person **separates** other persons on the basis of one of the enumerated grounds **without** an **objective** and reasonable **justification**.
  - *European Commission against Racism and Intolerance*



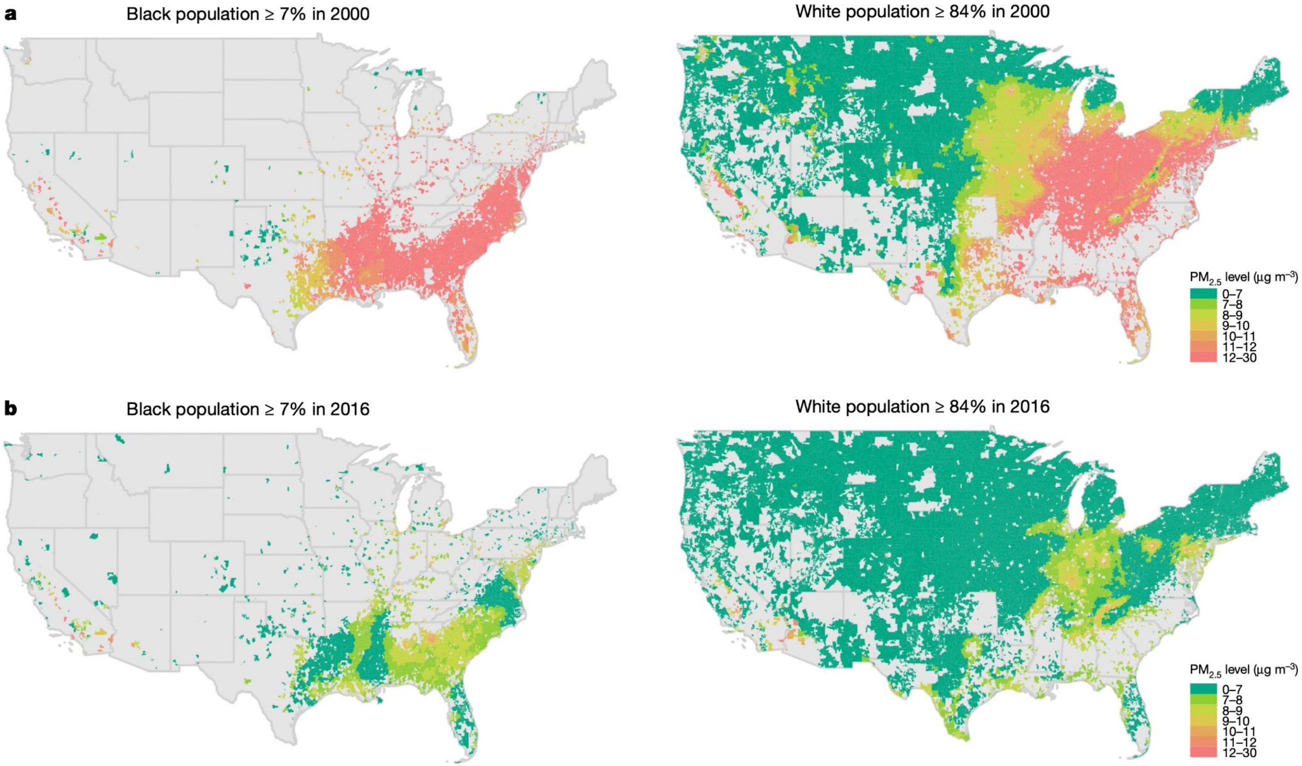
Segregation

C’mon it’s 2023… (I)



Segregation

C’mon it’s 2023… (II)



Segregation

Did you mean Schelling?

Journal of Mathematical Sociology
1971, Vol. 1, pp 143–186

DYNAMIC MODELS OF SEGREGATION†

THOMAS C. SCHELLING
Harvard University

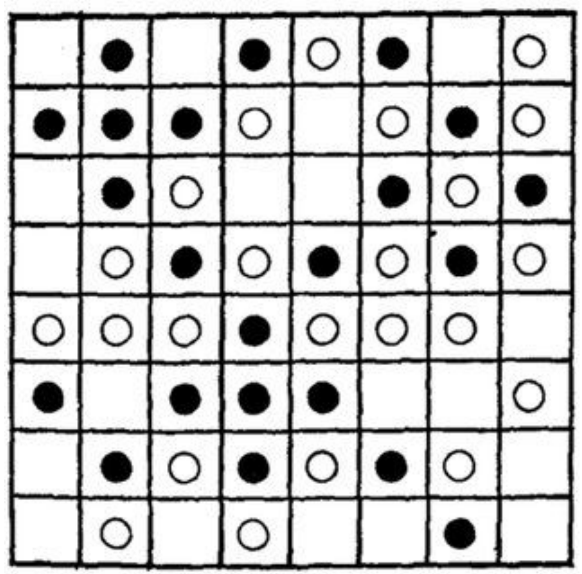
Some segregation results from the practices of organizations, some from specialized communication systems, some from correlation with a variable that is non-random; and some results from the interplay of individual choices. This is an abstract study of the interactive dynamics of discriminatory individual choices. One model is a simulation in which individual members of two recognizable groups distribute themselves in neighborhoods defined by reference to their own locations. A second model is analytic and deals with compartmented space. A final section applies the analytics to 'neighborhood tipping.' The systemic effects are found to be overwhelming: there is no simple correspondence of individual incentive to collective results. Exaggerated separation and patterning result from the dynamics of movement. Inferences about individual motives can usually not be drawn from aggregate patterns. Some unexpected phenomena, like density and vacancy, are generated. A general theory of 'tipping' begins to emerge.

Schelling Model

An Overview (I)

The Model

- **First** ABM of the history
- City as a **chessboard**
- **Two races** of agents (b/w)
    - 1 householder occupy 1 houseunits (cell)
- **Initially** agents at **random**

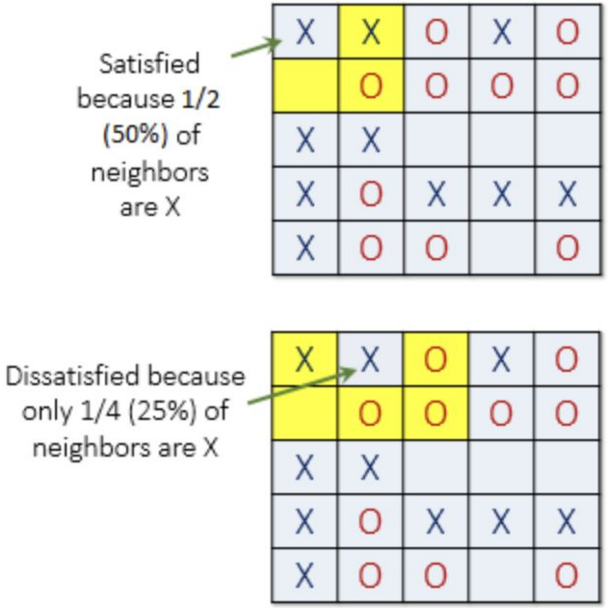


Schelling Model

An Overview (II)

The Dynamics

- Each agent has **8** direct **neighbours**
- At each time step ***dissatisfied*** agents **relocate**
- Agents desire a fraction (**B_a**)of their neighbors (B) to be **like** them for being satisfied
  - the higher B_a the higher the **intolerance**
- If $B < B_a$ the agent relocate to a **free** cell where it is **satisfied**



Schelling Model

Let's start… From the end!

Discovery #1

- Even if all agents tolerate up to ⅔ different neighbours
  - City becomes segregated

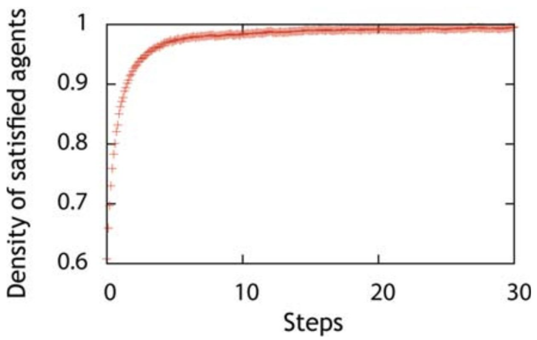
Discovery #2

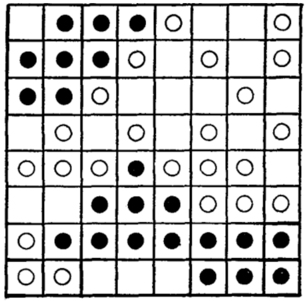
- Segregation is **sudden**
  - **Small** number of **step** is needed

Discovery #3

- At the end city is **more** segregated than necessary
  - **Overwhelming** effect

$$Tolerance \neq Inclusion$$





Simulating Schelling

http://nifty.stanford.edu/2014/mccown-schelling-model-segregation/

https://ncase.me/polygons/

https://web.mit.edu/djwendel/www/biograph/polygons6-9-18-003/

Schelling Model

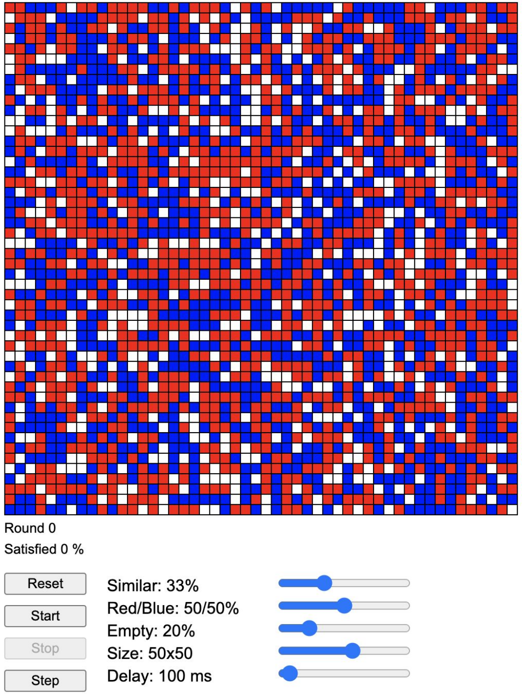
Simulations (I)

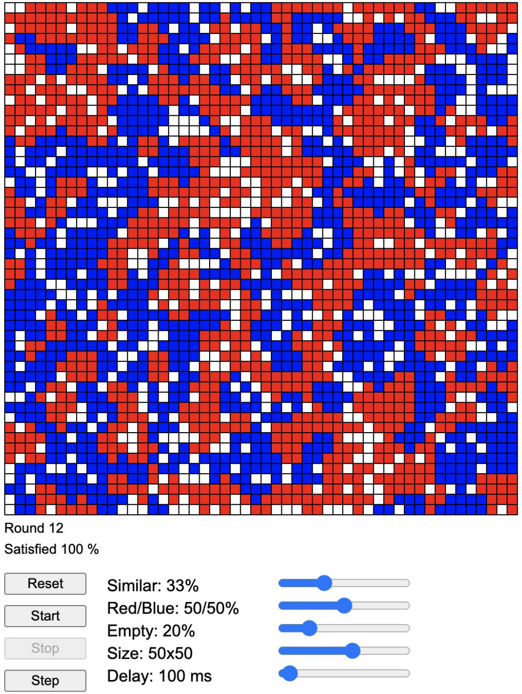
Question #1

Which is (+/-) the **number** of **steps** needed for segregating a 50x50 city? (Schelling default values)

Answer #1

12





Schelling Model

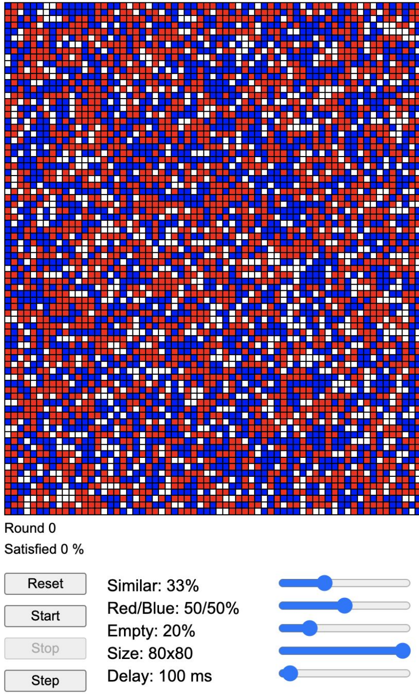
Simulations (II)

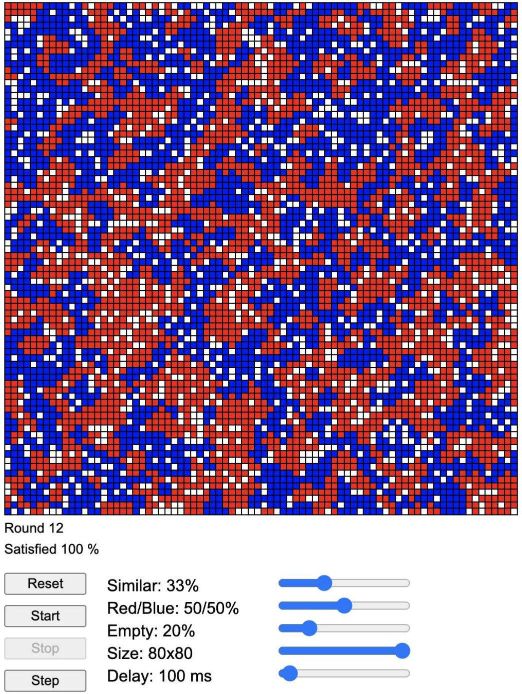
Question #2

What happens to the convergence **time** if we change city dimensions?

Answer #2

Nothing





Schelling Model

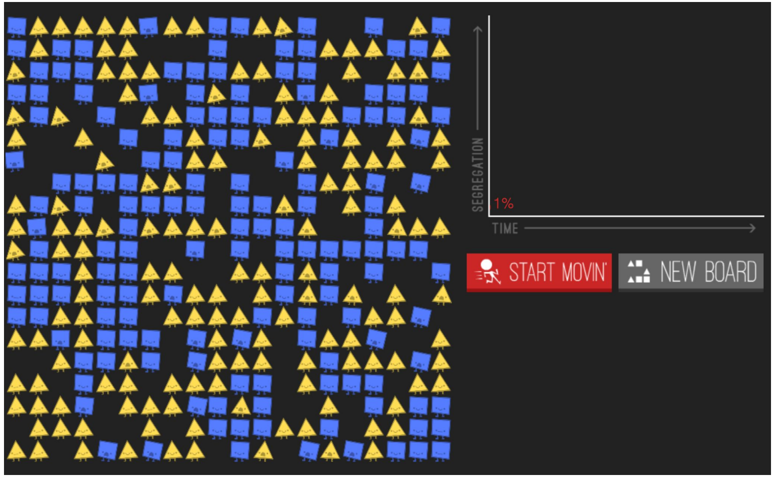
Simulations (III)

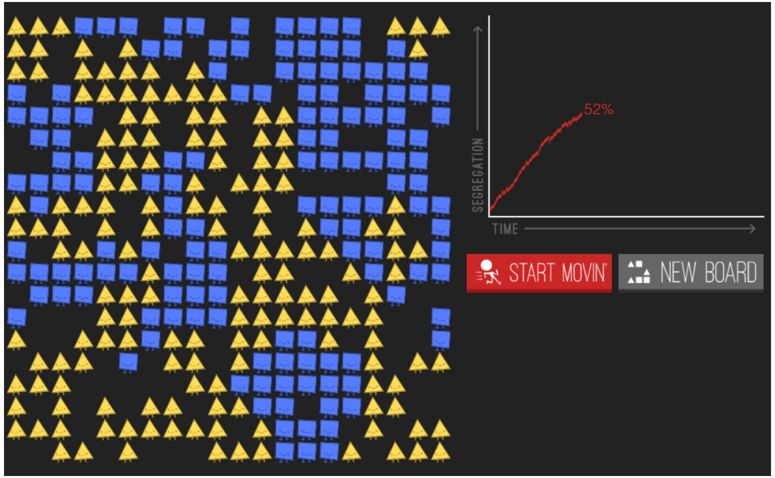
Question #3

Which **final** segregation **level** do you expect with a **intolerance** level of 1/3? (Classical Schelling)

Answer #3

>50%





Schelling Model

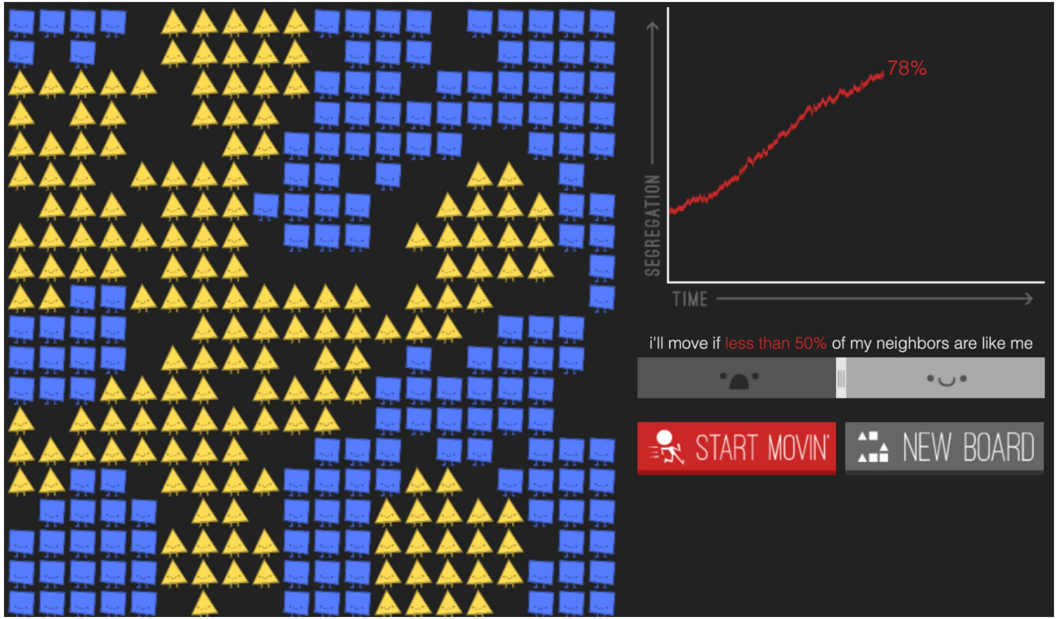
Simulations (IV)

Question #4

What happens to the convergence **time** if we increase the **intolerance**? And to the final segregation level?

Answer #4

Increase, especially the second



Schelling's take-home messages

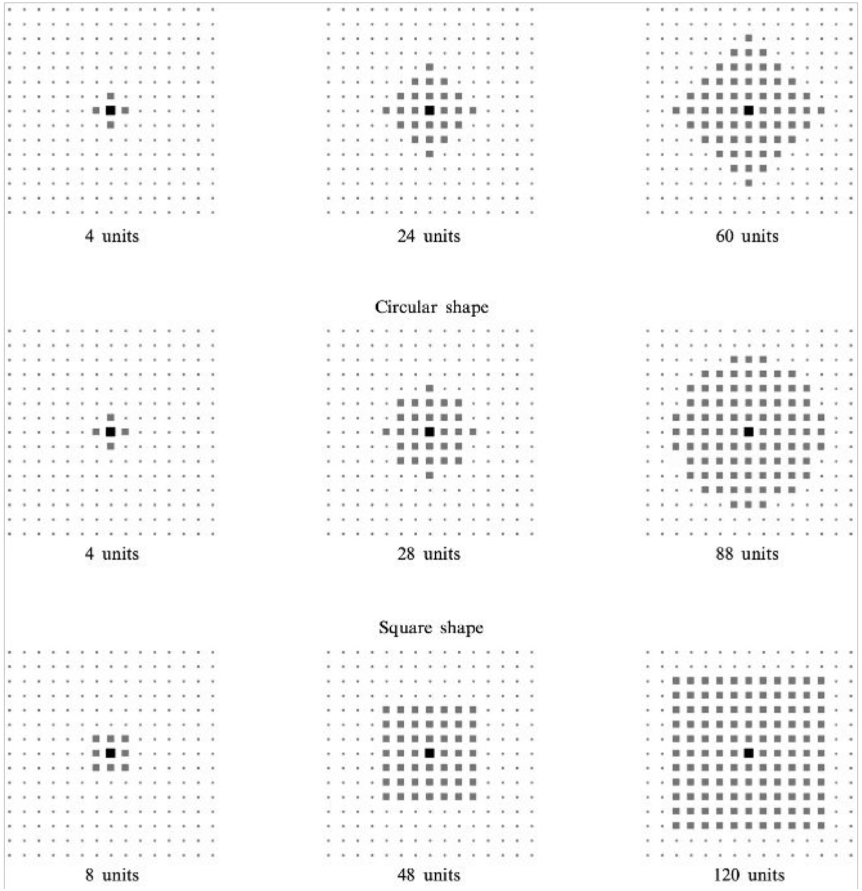
- A crowd of tolerants is not a tolerant crowd
- Tipping point can, theoretically, exist
    - Less racist agents will follow the herd too:
        - If everyone of my race is leaving… I'm leaving
- "People get separated along many lines, and in many ways"

Schelling Model

Classical Variants

- Model change
  - Agent **vision** and behavior[1]
  - **Environment** setting[2]
  - **Population** proportions
- Role of
  - **Venues**[3]
    - Mitigate segregation
  - **Vision**[4]
    - Neighborhood

[1] M. Fossett and D. R. Dietrich, "Effects of city size, shape, and form, and neighborhood size and shape in abm of residential segregation: Are schelling-style preference effects robust?," 2009
[2] T. Rogers and A. J. McKane, "A unified framework for schelling's model of segregation," 2011
[3] D. Silver, U. Byrne, and P. Adler, "Venues and segregation: A revised schelling model," 2021.
[4] A. J. Laurie and N. K. Jaggi, "Role of 'vision' in neighbourhood racial segregation," ,2003.

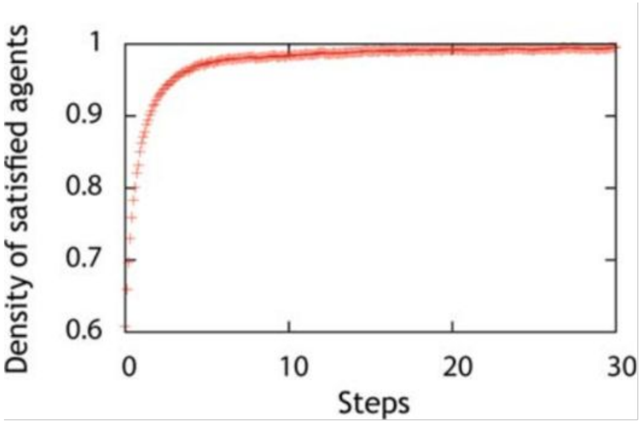


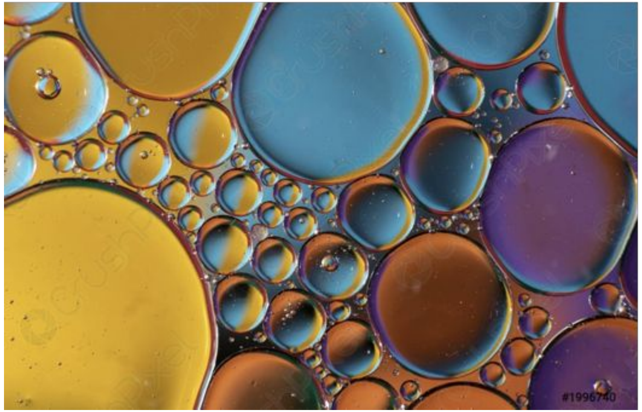
Schelling Model

A physics perspective

- Segregation is a **phase transition**[1]
- It happens like a clustering of **droplets**[2]
    - Same law of **Ising** Model[3]
- **Tipping** point is hard to find[4]
- **Attractors** exist[5]

[1] L. Gauvin, J. Vannimenus, and J.-P. Nadal, “Phase diagram of a schelling segregation model,” 2009.
[2] D. Vinkovi´c and A. Kirman, “A physical analogue of the schelling model,” 2006.
[3] D. Stauffer, and S. Solomon Ising, “ Schelling and self-organising segregation “ 2007
[4] N. G. Domic, E. Goles, and S. Rica, “Dynamics and complexity of the schelling segregation model,” 2011
[5] V. Cortez, P. Medina, E. Goles, R. Zarama, and S. Rica, “Attractors, statistics and fluctuations of the dynamics of the schelling’s model for social segregation,” 2015

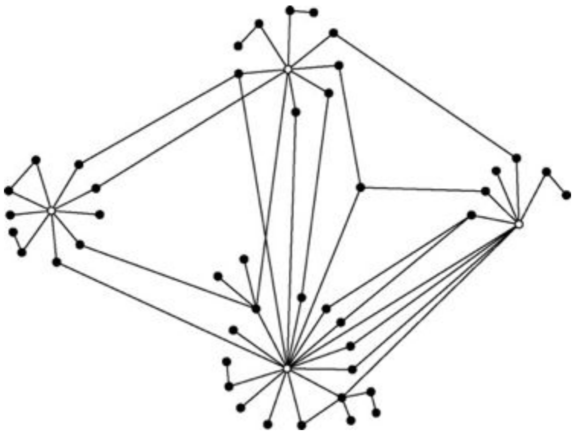


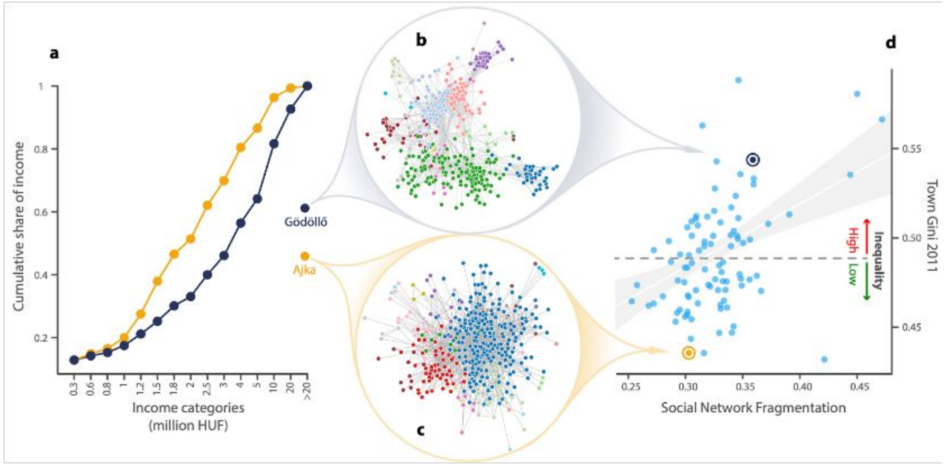


Schelling Model

A networks perspective

- Different **network** structure of the city are not crucial[1]:
  - Several **indexes** developed[2,3]
- The more the **heavy** infrastructures in the **road** network
  - The more the segregation levels[4]



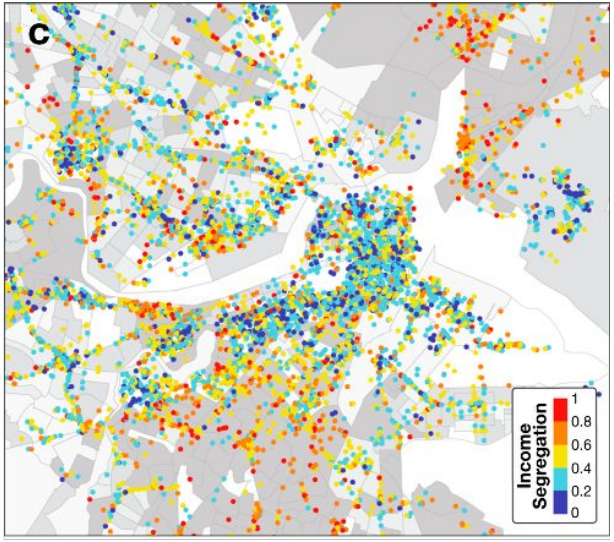


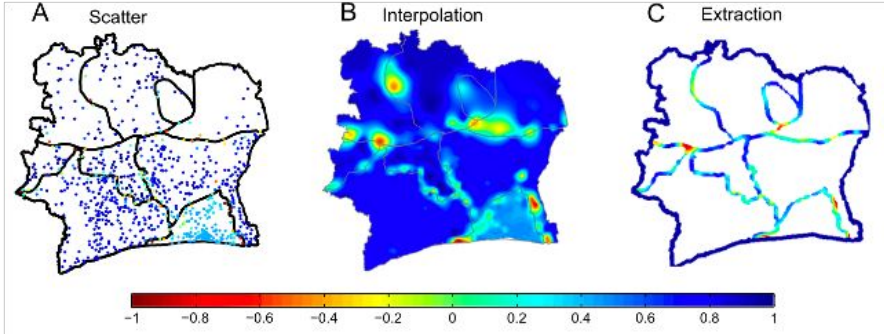
[1] G. Fagiolo, M. Valente, and N. Vriend, "Segregation in networks," 2007.
[2] Freeman and L. C, "Segregation in social networks," 1978
[3] Echenique, F. and Fryer Jr, R. G. " A measure of segregation based on social interactions ". 2007
[4] G. Toth, J. Wachs, R. Di Clemente, A. Jakobi, B. Sagvari, J. Kertesz, and ´ B. Lengyel, "Inequality is rising where social network segregation interacts with urban topology," 2021

Schelling Model

Mobility and AI

- Place and Social exploration explain experienced **income** segregation[1]
  - Tested on large American cities
- Assessing the quality of a political **partition** studying the **strength** of the **border**[2]
- First use of **Reinforcement** Learning[3]





[1] E. Moro, D. Calacci, X. Dong, and A. Pentland, "Mobility patterns are associated with experienced income segregation in large us cities," 2021.
[2] A. Amini, K. Kung, C. Kang, S. Sobolevsky, and C. Ratti, "The impact of social segregation on human mobility in developing and industrialized regions," 2014
[3] E. Sert, Y. Bar-Yam, and A. J. Morales, "Segregation dynamics with reinforcement learning and agent based modeling," 2020

www.nature.com/scientificreports

scientific reports

Check for updates

OPEN

Mobility constraints in segregation models

Daniele Gambetta[1,2], Giovanni Mauro[1,2,3] & Luca Pappalardo[1]

Since the development of the original Schelling model of urban segregation, several enhancements have been proposed, but none have considered the impact of mobility constraints on model dynamics. Recent studies have shown that human mobility follows specific patterns, such as a preference for short distances and dense locations. This paper proposes a segregation model incorporating mobility constraints to make agents select their location based on distance and location relevance. Our findings indicate that the mobility-constrained model produces lower segregation levels but takes longer to converge than the original Schelling model. We identified a few persistently unhappy agents from the minority group who cause this prolonged convergence time and lower segregation level as they move around the grid centre. Our study presents a more realistic representation of how agents move in urban areas and provides a novel and insightful approach to analyzing the impact of mobility constraints on segregation models. We highlight the significance of incorporating mobility constraints when policymakers design interventions to address urban segregation.

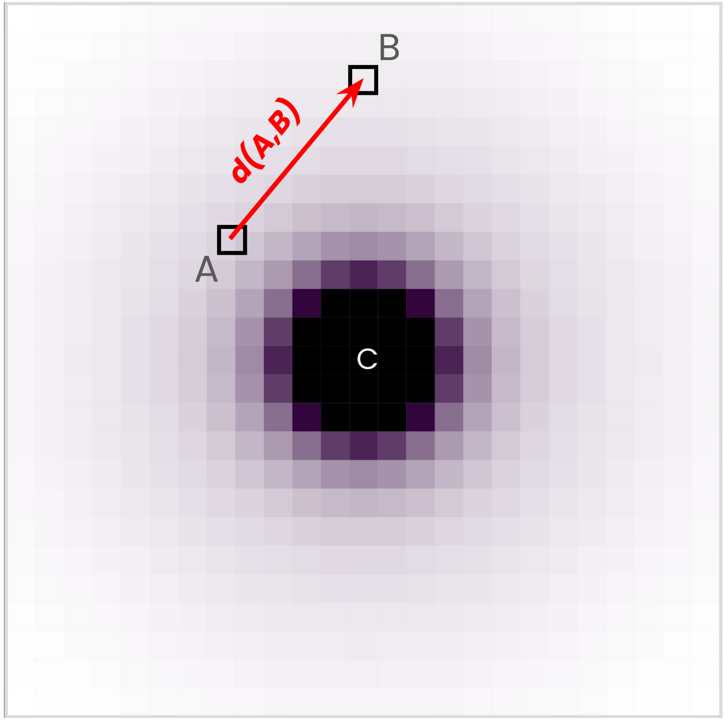
A class of mobility constrained models

An agent in cell *A* **relocates** to a cell *B* with a **probability**:

$$p(B) \propto r(B)^{\alpha} d(A,B)^{\beta}$$

with

- $r(B) \propto \frac{1}{d(B,C)^k}$ as relevance
- $d(A,B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$ as distance

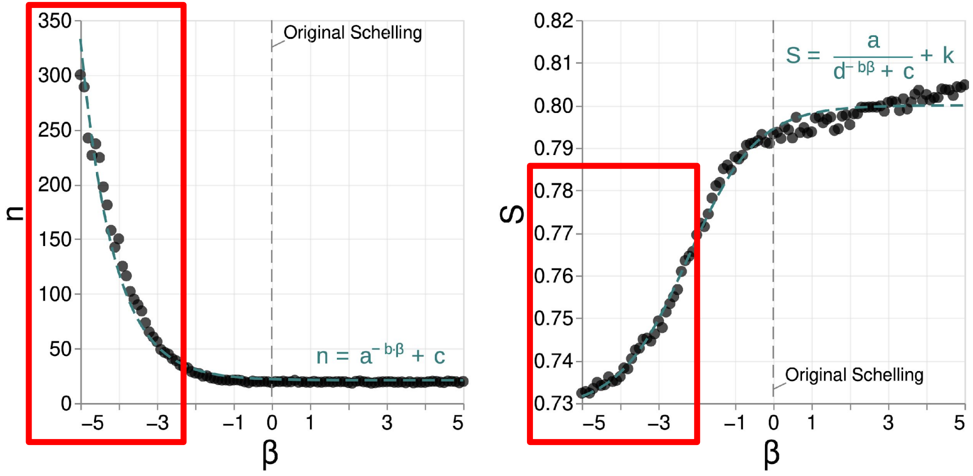


Results

Distance models (α=0)

The more the **distance constrains** the dynamics (β<0):

- The higher ***n***
- The lower ***S***

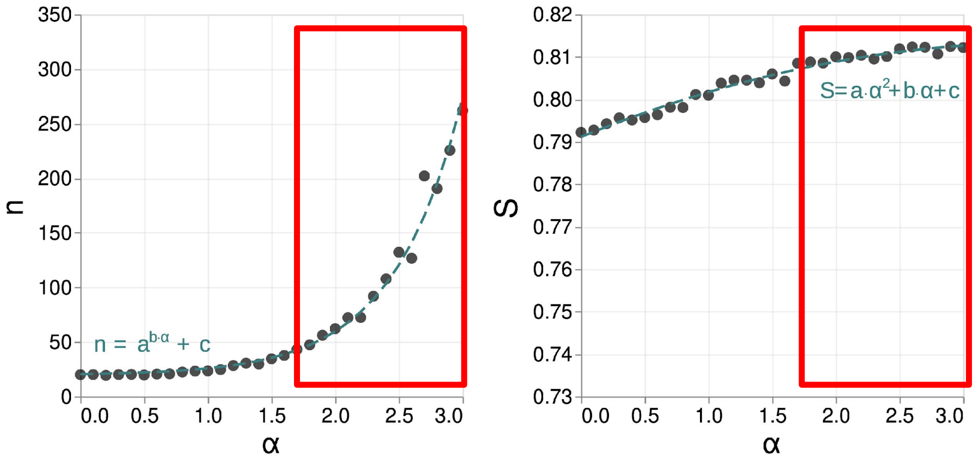


Results

Relevance models (β=0)

The more the relevance **impacts** the dynamics (α>0):

- The higher ***n***
- Slightly higher ***S***



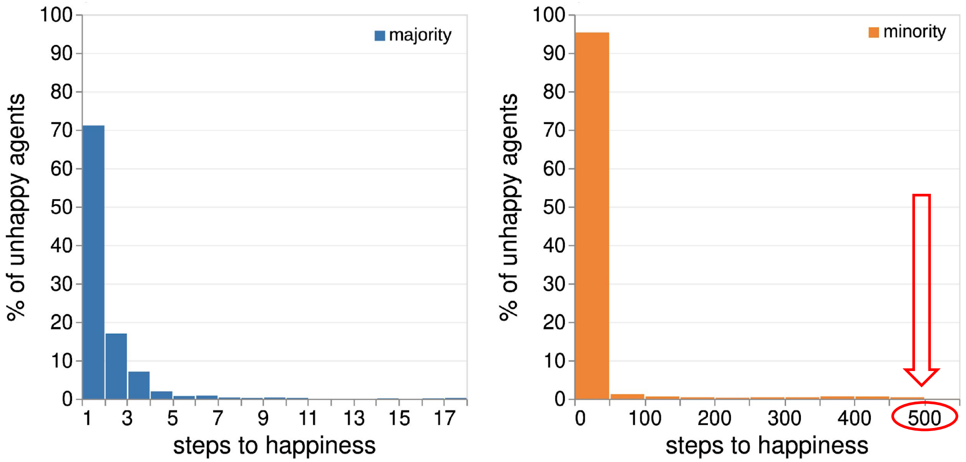
Why elongation?

- A few **minority** agents stay unhappy for long time

    persistently unhappy agents (p.u. agents):

    unhappiness time > 95th percentile

- How can we characterize p.u. agents?



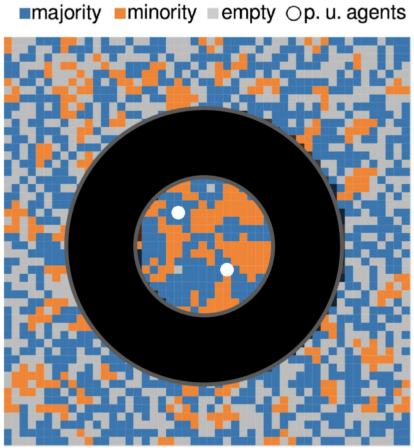
Majority
max **17** steps
(**1.57** avg)
to be happy

Minority
max **500** steps
(**14.78** in avg)
to be happy

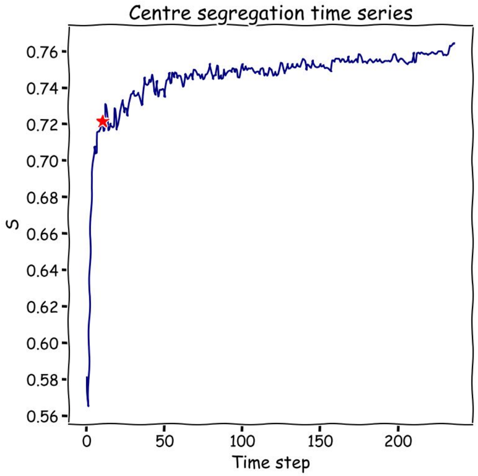
Gravity focus

Characterization of Persistently Unhappy Agents (p.u.agents)

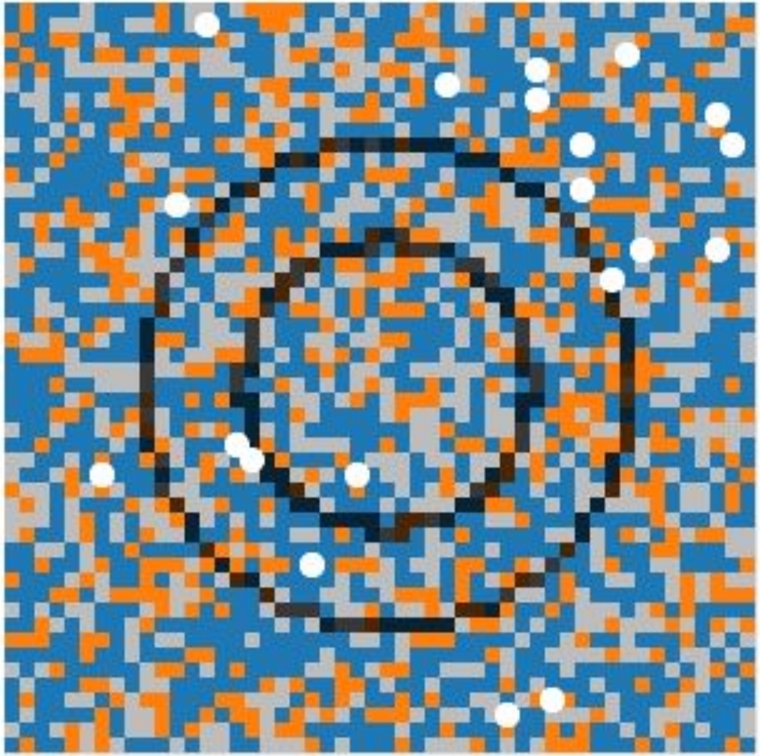
- Suburbia: circular band surrounding the centre
- At ★ step the centre-zone will be segregated:
    - **minority** agents in **suburbia** at ★ are the p.u.agents



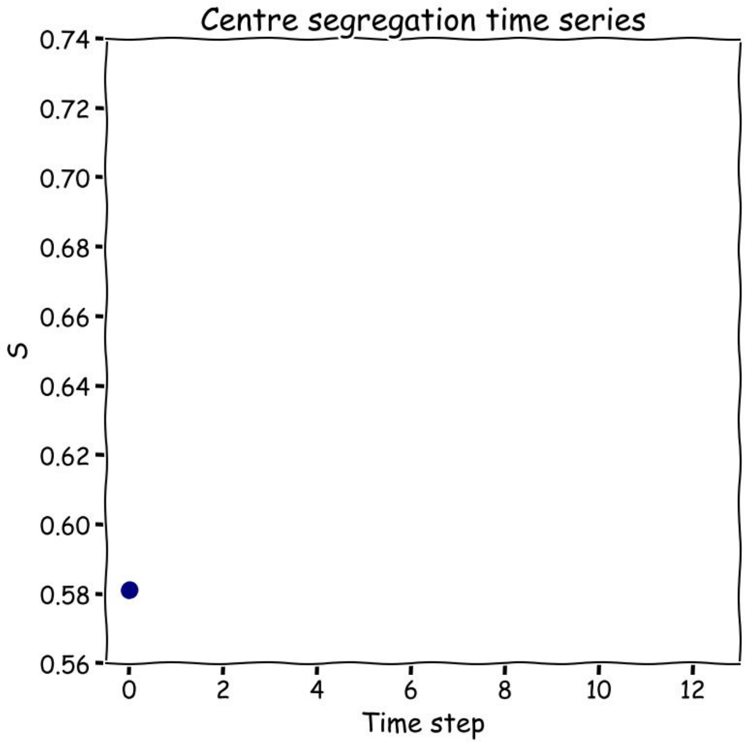
positions of p.u.a. at ★



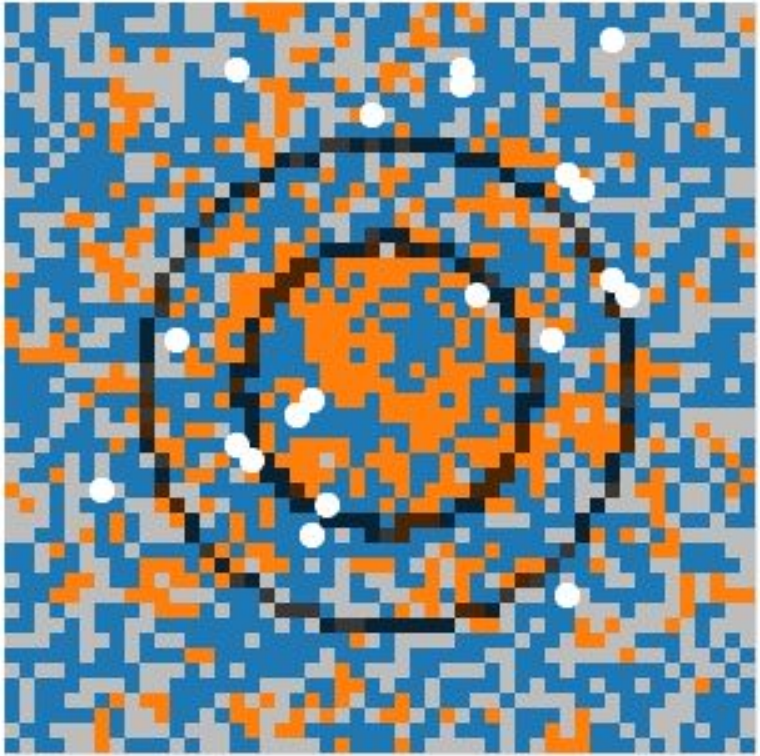
Step 0

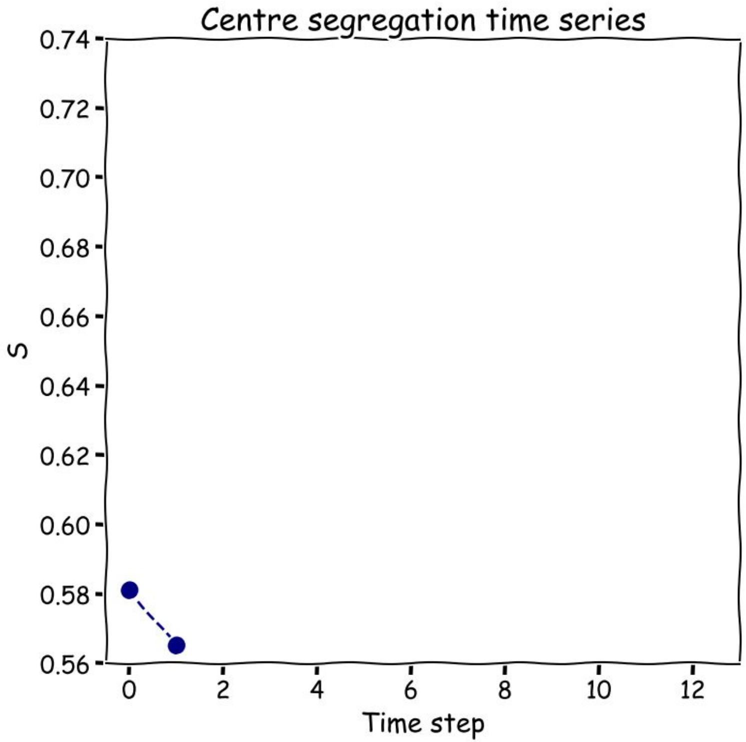


■majority ■minority ■empty ○p. u. agents



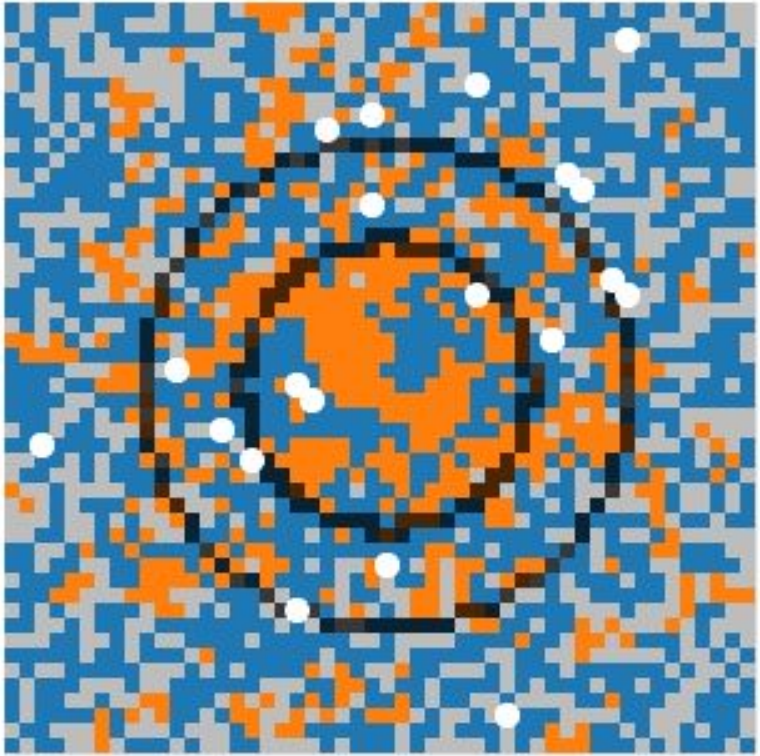
Step 1



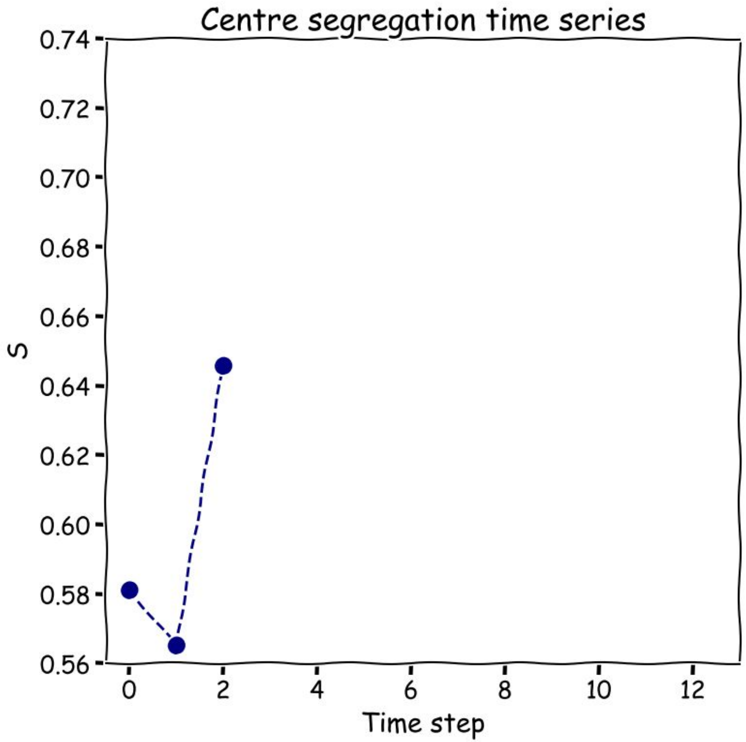


■majority ■minority ■empty ○p. u. agents

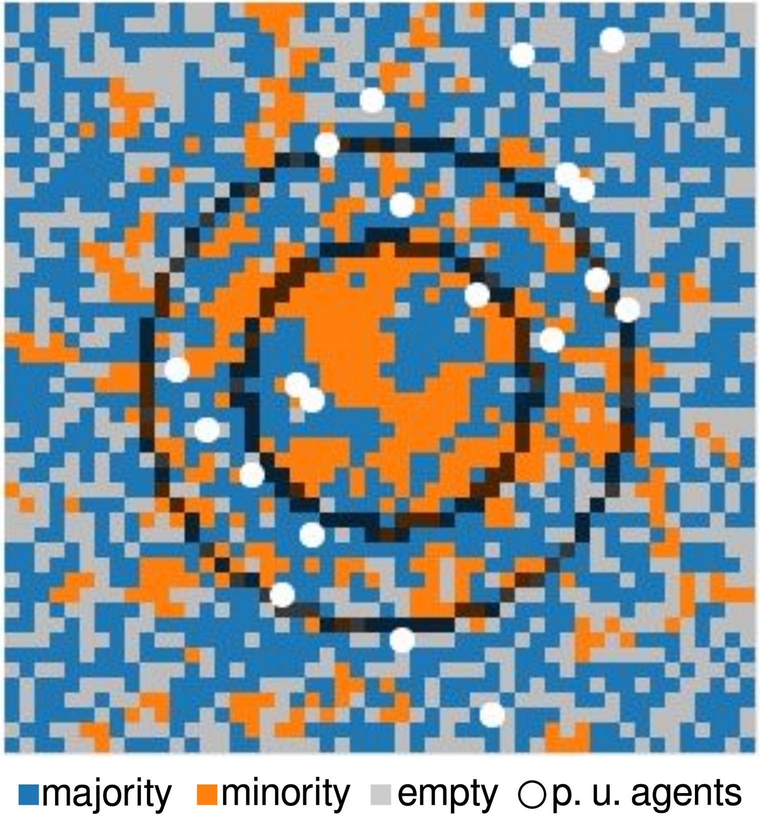
Step 2



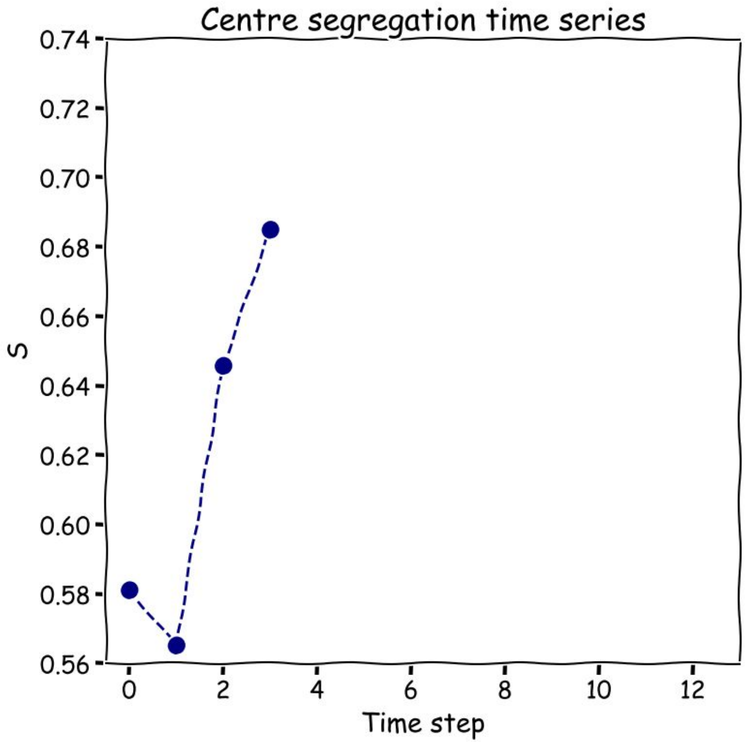
■majority ■minority ■empty ○p. u. agents



Step 3

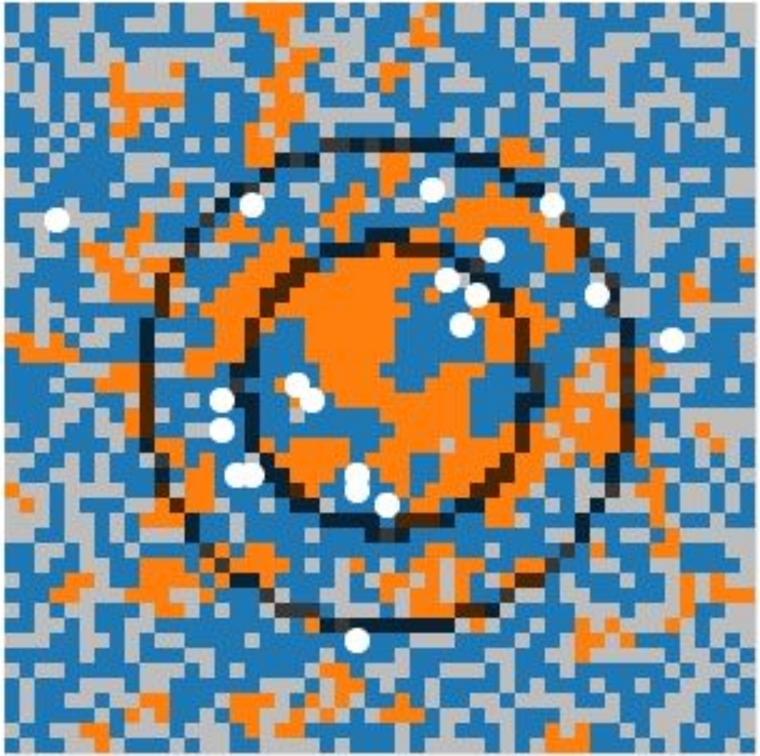


majority
minority
empty
p. u. agents

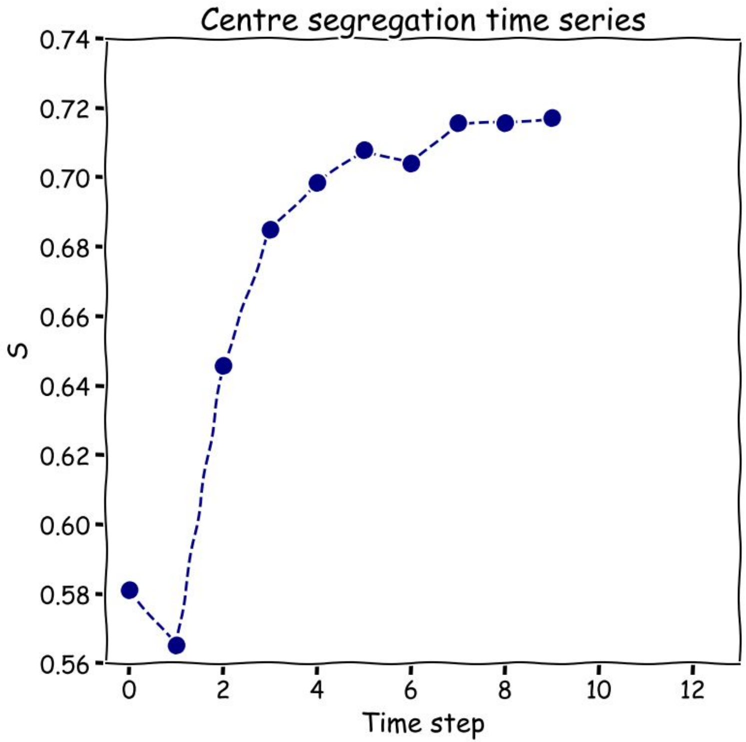


Centre segregation time series
0.74
0.72
0.70
0.68
0.66
0.64
0.62
0.60
0.58
0.56
S
0
2
4
6
8
10
12
Time step

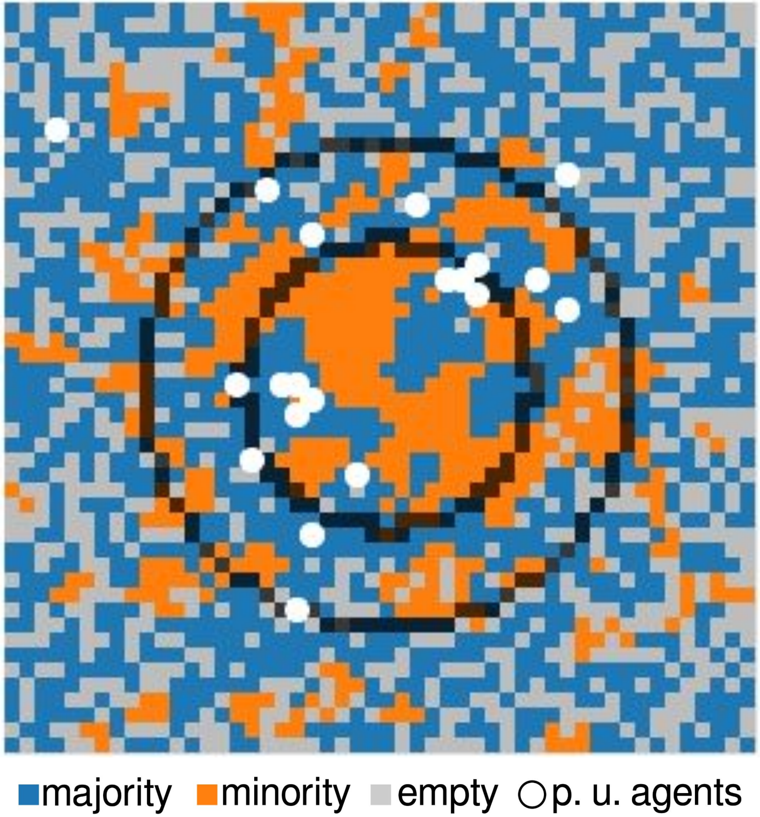
Step 9

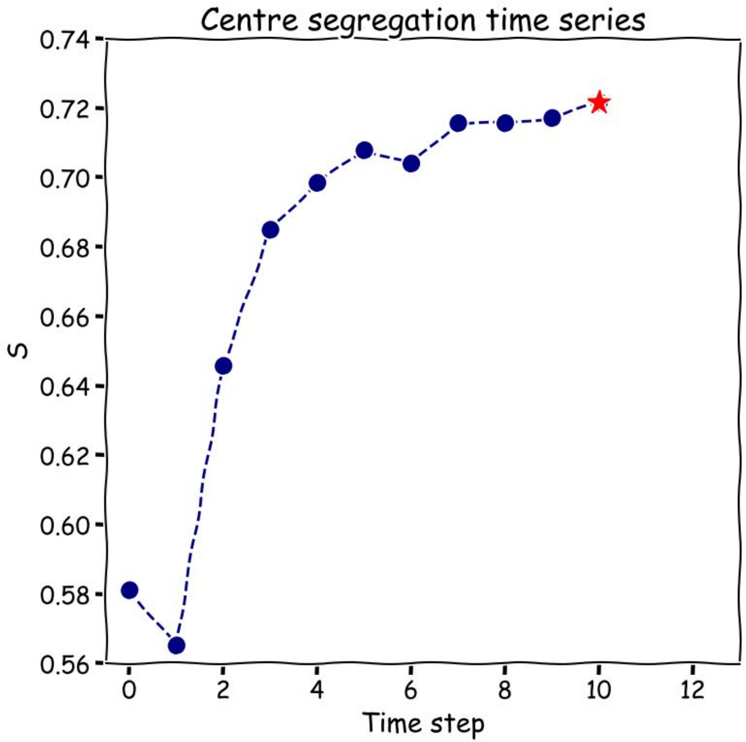


■majority ■minority ■empty ○p. u. agents

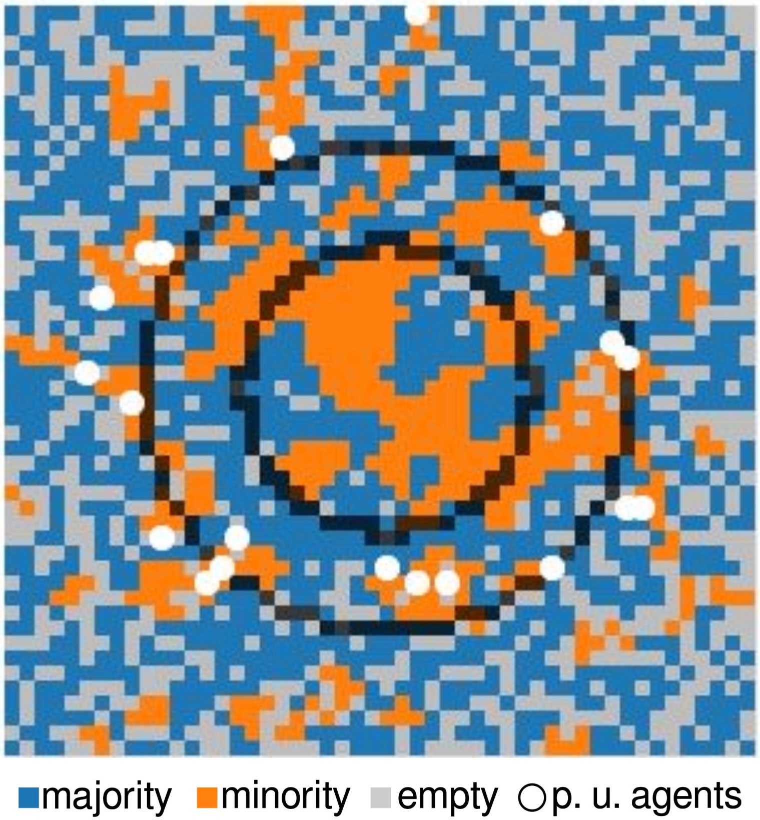


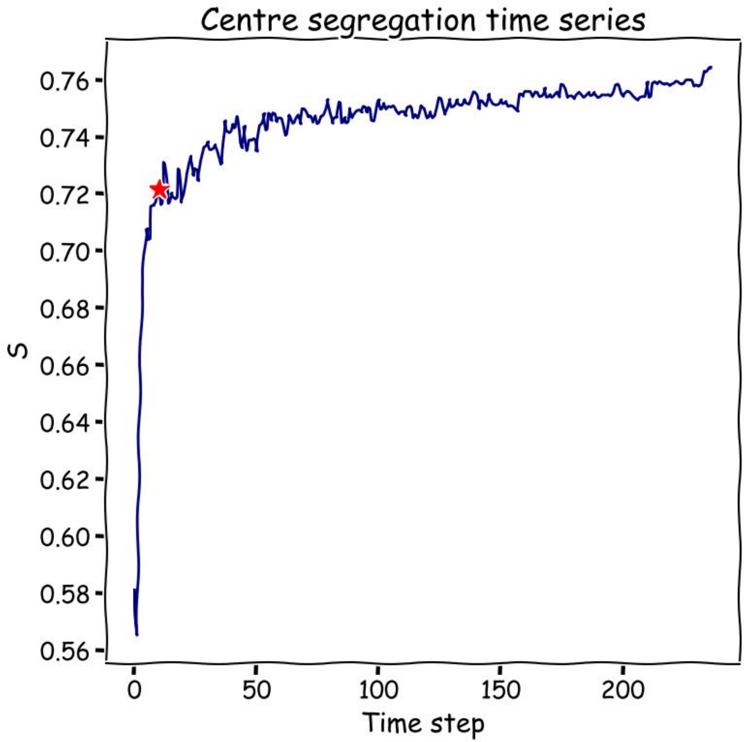
Step 10: Tipping point (stable)





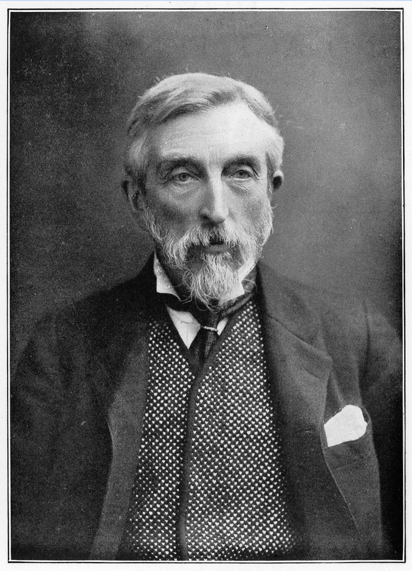
Step 236: Final





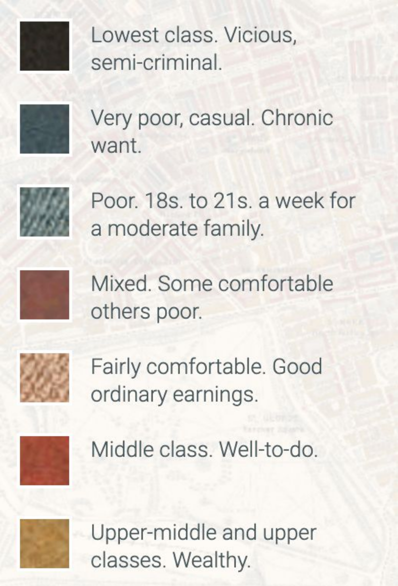
INTERVALLO

Charles Booth and London’s map of poverty



Between 1886 and 1903, he created the first map of poverty and criminality in London.
→ https://booth.lse.ac.uk/learn-more/what-was-the-inquiry

Charles Booth’s London Poverty maps and police notebooks

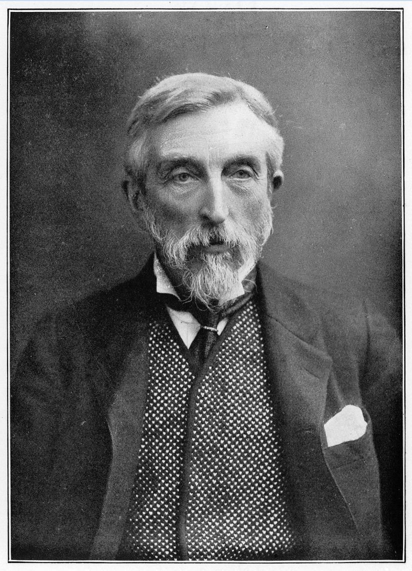


INTERVALLO

- Lowest class. Vicious, semi-criminal.
- Very poor, casual. Chronic want.
- Poor. 18s. to 21s. a week for a moderate family.
- Mixed. Some comfortable others poor.
- Fairly comfortable. Good ordinary earnings.
- Middle class. Well-to-do.
- Upper-middle and upper classes. Wealthy.

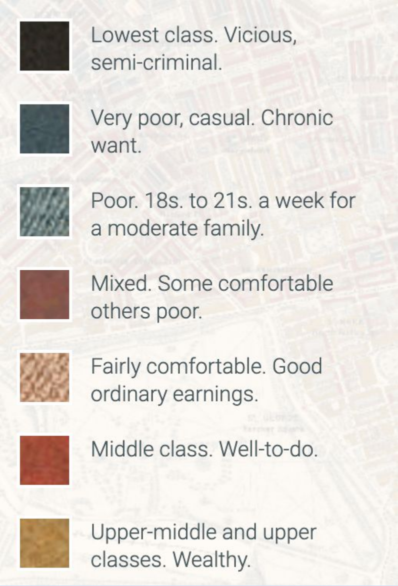
INTERVALLO

Charles Booth and London's map of poverty



Booth also talked about gentrification and anti-gentrification:

"the red and yellow classes leave, and the streets they once occupied turn pink, while the streets that were previously pink turn purple, and from purple turn blue."



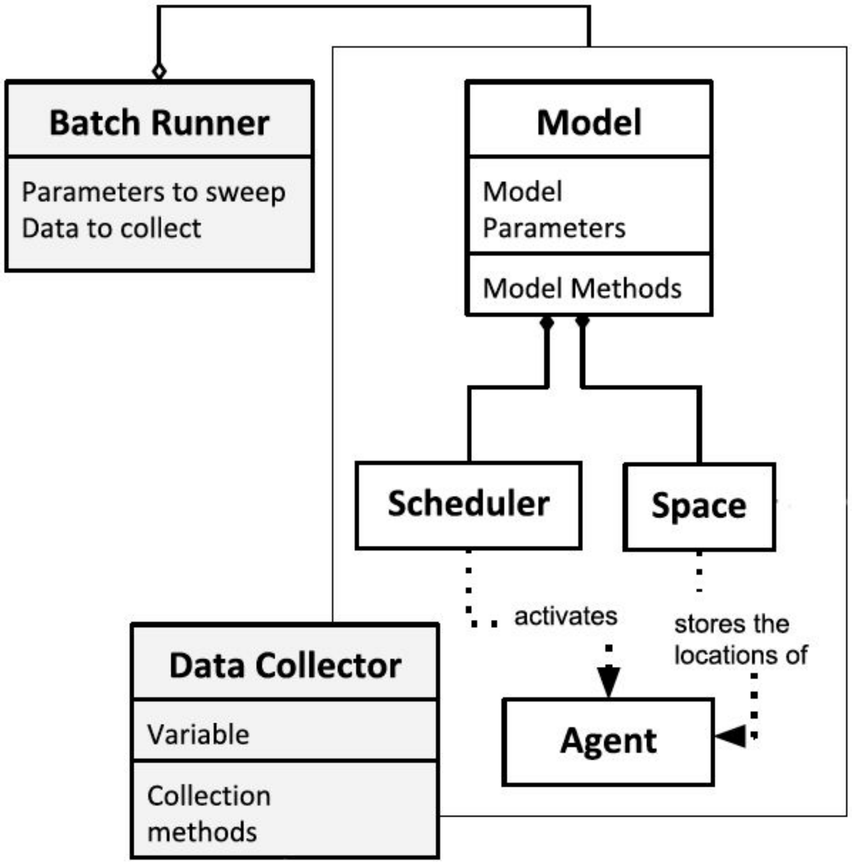
Modelling Agent-Based models

- Hard:
  - Need to account for multilinearity, time management, simultaneous choices etc.
- Need for a common framework
  - Several have been used in the past (NetLogo, JADE etc.)
- … and for Data Science
  - MESA

MESA

A Python Framework

- Common **programming** language suite for ABM
- Two main classes:
  - **Model**
    - Stores variables and methods concerning the environment and the policies
  - **Agent**
    - Defines agents' behaviour. Each agent has a unique ID.
  - Each agent is linked to a model (accessible with `Agent.model`)
- Automatize handling of the **schedule** of an underlying **concurrent** structure



First toy ABM

- The best way to explain an ABM is to make an example of it
- ...so let’s see an example of a toy ABM
  - There are N agents
    - All of them start with 1 unit of money
    - Placed at random on a grid (can share a cell)
  - At every step an agent:
    - Moves at random within its neighborhood
    - Gives 1 unit of money (if they have it) to some other agent in the same cell

X		O	OX
XX	XX	O	
OX		XO	
	X		O

Model

```python
def compute_gini(model):
    agent_wealths = [agent.wealth for agent in
model.schedule.agents]
    x = sorted(agent_wealths)
    N = model.num_agents
    B = sum(xi * (N - i) for i, xi in enumerate(x)) / (N *
sum(x))
    return 1 + (1 / N) - 2 * B


class MoneyModel(mesa.Model):

    def __init__(self, N, width, height):
        self.num_agents = N
        self.grid = mesa.space.MultiGrid(width, height, True)
        self.schedule = mesa.time.RandomActivation(self)
        self.running = True

        # Create agents
        for i in range(self.num_agents):
            a = MoneyAgent(i, self)
            self.schedule.add(a)
            # Add the agent to a random grid cell
            x = self.random.randrange(self.grid.width)
            y = self.random.randrange(self.grid.height)
            self.grid.place_agent(a, (x, y))

        self.datacollector = mesa.DataCollector(
            model_reporters={"Gini": compute_gini},
           agent_reporters={"Wealth": "wealth"})

    def step(self):
        self.datacollector.collect(self)
        self.schedule.step()
```

- Our MoneyModel extend the MESA's Model class
- Initialize the object Model with parameters
- Our space is a MultiGrid: we specify a width, height and if we want it *toroidal*
- Mesa manage the Schedule: How do I wake up my agents? Who check and move first? `running` flag means simulation started
- For loop generation: create an agent, add it to the schedule policy and place it on a random place (`grid.place_agent method`)
- `DataCollector:` This MESA's structure allows to store, at each step, some internal variables or some function-calculated values (`compute_gini`)
- Most important part: at each step of the model we want to collect the data and to make the agent behave (make steps). How? According to the scheduling policy

Agent

```
class MoneyAgent(mesa.Agent):
    """An agent with fixed initial wealth."""

    def __init__(self, unique_id, model):
        super().__init__(unique_id, model)
        self.wealth = 1

    def move(self):
        possible_steps = self.model.grid.get_neighborhood(
            self.pos, moore=True, include_center=False)
        new_position = self.random.choice(possible_steps)
        self.model.grid.move_agent(self, new_position)

    def give_money(self):
        cellmates =
self.model.grid.get_cell_list_contents([self.pos])
        if len(cellmates) > 1:
            other = self.random.choice(cellmates)
            other.wealth += 1
            self.wealth -= 1

    def step(self):
        self.move()
        if self.wealth > 0:
            self.give_money()
```

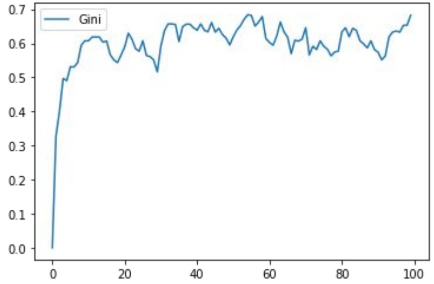
- Our MoneyAgent extend the MESA's Agent class. It is **linked** to the model. The starting wealth value is 1
- Auxiliary function that define my movement . An agent retrieve its neighbors CELLS calling `model.grid.get_neighborhood` and select one position at random. Then it uses `model.grid.move_agent` for moving
- Function for giving money. An agent access the list of co-located agents (`model.grid.get_cell_list_contents`). If it has other agent in its cell it pick one at random and give a unit of money
- Define behaviour at each step. I move and then I call the function for giving money. CALLED BY `model.step()` (which in turn calls `model.schedule.step()` that respects the policy)

Running: Single Step

```
model = MoneyModel(50, 10, 10)
for i in range(100):
      model.step()
```

- Initialize a model with 50 agents on a 10x10 grid and make 100 steps of simulation.

```
gini = model.datacollector.get_model_vars_dataframe()
gini.plot()
```




```
agent_wealth = model.datacollector.get_agent_vars_dataframe()
agent_wealth.head()
```

		Wealth
Step	AgentID	
0	0	1
	1	1
	2	1
	3	1
	4	1

Running: Batch Run

```
params = {"width": 10, "height": 10, "N": range(10, 500, 10)}
results = mesa.batch_run(
    model_cls = MoneyModel,
    parameters=params,
    iterations=5,
    max_steps=100,                Halt condition
    number_processes=1,           Multithreading for speed-up
    data_collection_period=1,     Step of collection (1 means after each step)
    display_progress=True)
```

- How many run of the model?
  - `range(10,500,10)` = 49 elements
  - 2 fixed parameters (w and h)
  - 5 iterations
  - … 49*5 = 245

```
import pandas as pd

results_df = pd.DataFrame(results)
print(results_df.keys())

> Index(['RunId', 'iteration', 'Step', 'width', 'height', 'N',
         'Gini', 'AgentID','Wealth'], dtype='object')
```

- Results can be stored in a df that contains the state of each run, agents etc.
  - Avg. 250 agents per simulation
  - 245 runs
    - Max 100 step per run
  - The df can be long 245*250*101 = 6186250

to study for the exam

Material

- [paper] Dynamic models of segregation, Schelling, Thomas C. Journal of mathematical sociology 1.2 (1971): 143-186.
- [paper] Mobility constraints in segregation models, Gambetta et al. Scientific Reports, 2023
- [tutorial] Introductory Tutorial to Mesa (with the MoneyModel example)

Homework

Literature propose 4 metrics for quantifying the segregation levels from data: (i) exposure: the extent to which different populations share the same residential areas; (ii) the evenness (and clustering): to which extent populations are evenly spread in the metropolitan area; (iii) concentration: to which extent populations concentrate in the areal units they occupy; and (iv) centralization: to which extent populations concentrate in the center of the city.

- Propose 4 simple mathematical formulas for calculate these metrics
- Implement them in Python/Mesa
- Create 4 plots that shows the trend of these 4 metrics during the Schelling dynamics