Agent-Based Modelling in NetLogo

David Hales

www.davidhales.com/abm-netlogo
Resources

• NetLogo runs on most platforms (Windows, Linux, Mac). It can be downloaded from: https://ccl.northwestern.edu/netlogo

• I ran a lecture course last year which looked at several ABM in the scientific literature: http://davidhales.com/msiis
Schelling’s segregation model

- Could communities become segregated by race, sex, social class, profession etc.
  - if no explicit barriers prevent integration
  - if individuals are tolerant of others
- Explores effects of individual movement (micro interaction) decisions on segregation (emergent macro) outcomes
Schelling’s segregation model

- In his paper Schelling describes several variants of his model:
  - 1D version (agents ordered on a line)
  - 2D versions (agents placed on a checkerboard)
  - Generalised group (agents entering or leaving a large group)
- We will focus only on the 2D version here
- Schelling did not use a computer but a checkerboard with coins and did it by hand
- He called it a game of “solitaire”
Schelling’s segregation model

• Schelling makes it clear he is talking about segregation *in general* based on any recognisable attribute and interaction structure

• However he often makes a racial / residential neighbourhood *interpretation*

• This may have something to do with the political and social background of late sixties USA

• *Aside: In is interesting to note the political background in which social models come about*
Schelling’s 2D segregation model

• Bounded grid of cells (checkerboard)
• Each cell may contain an agent or be empty
• Each agent is one of two colours (say, black or white)
• Neighbourhood of each cell are surrounding 8 cells (the Moore neighbourhood)
• An agent is
  – “satisfied” if at least T% of its neighbours the same colour
  – If <T% of neighbours same colour then it is not satisfied
• Unsatisfied agents try to move to nearby empty locations that satisfy them. Satisfied agents stay where they are
Schelling’s 2D segregation model

- Schelling notes that about 25-30% empty cells allows for enough space for movement
- Places equal white/black number of agents randomly on a 13 x 16 grid
- By hand he moves the agents until they are all satisfied and an equilibrium is reached
- His movement involves picking up unsatisfied agents and placing them in the nearest empty cell that makes them satisfied
- He shows diagrams of some example start and end configurations and discusses them
Schelling’s 2D segregation model

• He finds that with T between 35% to 50% an equilibrium is reached producing high segregation
• With T <= 30% much less segregation is found
• He measures segregation by calculating ave% of agents neighbours that are same colour
• He states he can not do enough simulations by hand to generalise but uses experiments to inform hypotheses
Schelling’s 2D segregation model

• Schelling observes:
  – Even comparatively “tolerant” agents (say T=35%) can produce high segregation
  – This means that if agents don’t want to be in a significant minority => high segregation
  – Playing around with coins on a checkerboard produced counter-intuitive insights
  – Others can reproduce his results (in about 10 minutes with paper and coins)
NetLogo segregation model

• File>models library/social science/segregation
• 2 input parameters: density, %similar-wanted (T)
• Three output windows:
  – percent similar time series (segregation measure)
  – number unhappy (not satisfied) time series
  – 2D grid showing red & green agents
• To run first click “setup” button then “go” button
• Simulation stops when all agents satisfied or go button is pressed again
Playing with the model

• Playing with the model:
  – The T value (%-similar-wanted) slider can be moved during a simulation run
  – However to change density of agents the setup button needs to be pressed to re-initialise the population
  – Commenting out the stop condition in the code means the simulation keeps running making it easier to play with T value while running
Rough observations from playing

• With density value at 80%:
  – T < 20% tends to produce %similar < 60%
  – T > 30% tends to produce %similar > 70%
  – T > 80% things never seem to stabilise
  – T < 70% things seem to stabilise quickly

• To get an idea of how T affects %similar (segregation) and how long it takes we need to do a systematic set of simulation runs.

• Note: a NetLogo tool called BehaviorSpace can do this automatically (we will look at in future labs)
NetLogo implementation

• 51 x 51 grid (wrapped) = 2601 cells (called patches)
• Agents (called turtles) placed on random patches. Divided between colours randomly, placed randomly
• For each cycle:
  – If all turtles are happy then stop simulation
  – Else move all unhappy turtles
• Movement rule (a random walk):
  – Repeat
    • Point turtle in random direction
    • Move forward a random distance
  – Until empty cell found
Code

• See exercise sheet for description of code
Tasks

• Modify model so agents take one of four colours (red, green, blue or yellow)

• Add a “Chooser” dropdown to the Interface (like the visualization one) that lets user select between two or four colours.

• Modify the find-new-spot procedure so instead of recursively moving it moves directly to a random empty patch. Hint: this can be done in one line by asking patches

• Create button called “find-clusters” that when clicked counts number of connected clusters of the same colour. Display the result in a monitor. Hint: this can be done with a recursive procedure
model after tasks
new code

to setup
  clear-all
  ;; create turtles on random patches.
  ask patches [  
    if random 100 < density [ ;; set the occupancy density
      sprout 1 [  
        if num-colors = "four" [  
          set color one-of [red green blue yellow]
        ]
        if num-colors = "two" [  
          set color one-of [red green]
        ]
      ]
    ]
  ]
update-variables
reset-ticks
end
new code

to find-new-spot2
    move-to one-of patches with [count turtles-here = 0]
end

to find-color-clusters
    set num-clusters 0
    ask turtles [ set visited? false ]
    ask turtles [
        if not visited? [
            set num-clusters num-clusters + 1
            visit-all-color-neighbours
        ]
    ]
end

to visit-all-color-neighbours
    set visited? true
    let same-color-neighbours (turtles-on neighbors) with [
        (not visited?) and
        (color = [color] of myself)
    ]
    ask same-color-neighbours [ visit-all-color-neighbours ]
end
Thomas Schelling

- American political economist
- “Nobel prize” in economics (2005)
- Involved in post WWII Marshall Plan
- Cold war strategist, US govt. RAND
- Not a “game theorist”, much more than that
- Helped inspire director Stanley Kubrick (who did movie 2001) to do movie “Dr Strangelove” (1964) This can be viewed as a satire on game theory – worth watching
- Rumours say that the character Dr Strangelove in the movie was partially inspired by John von Neumann
- Invented term “collateral damage” (1961) ?

Video interview: https://youtu.be/fujQaAgqgxQ