8. Schelling's Segregation Model

Modelling Social Interaction in Information systems

http://davidhales.com/msiis

David Hales, University of Szeged

dave@davidhales.com

Schelling's segregation model

- Seminal Agent Based Model (ABM)
- ABM is a term used for any model that
  - Specifies agent (micro) behaviour
  - Observes the emergent (macro) outcomes
- All models we have looked at are ABM's
- Non-ABM models specify the relationships between macro properties (generally)
- Schelling never used the term ABM

Schelling's segregation model

- Thomas Schelling (1971) "Dynamic Models of Segregation"
- 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

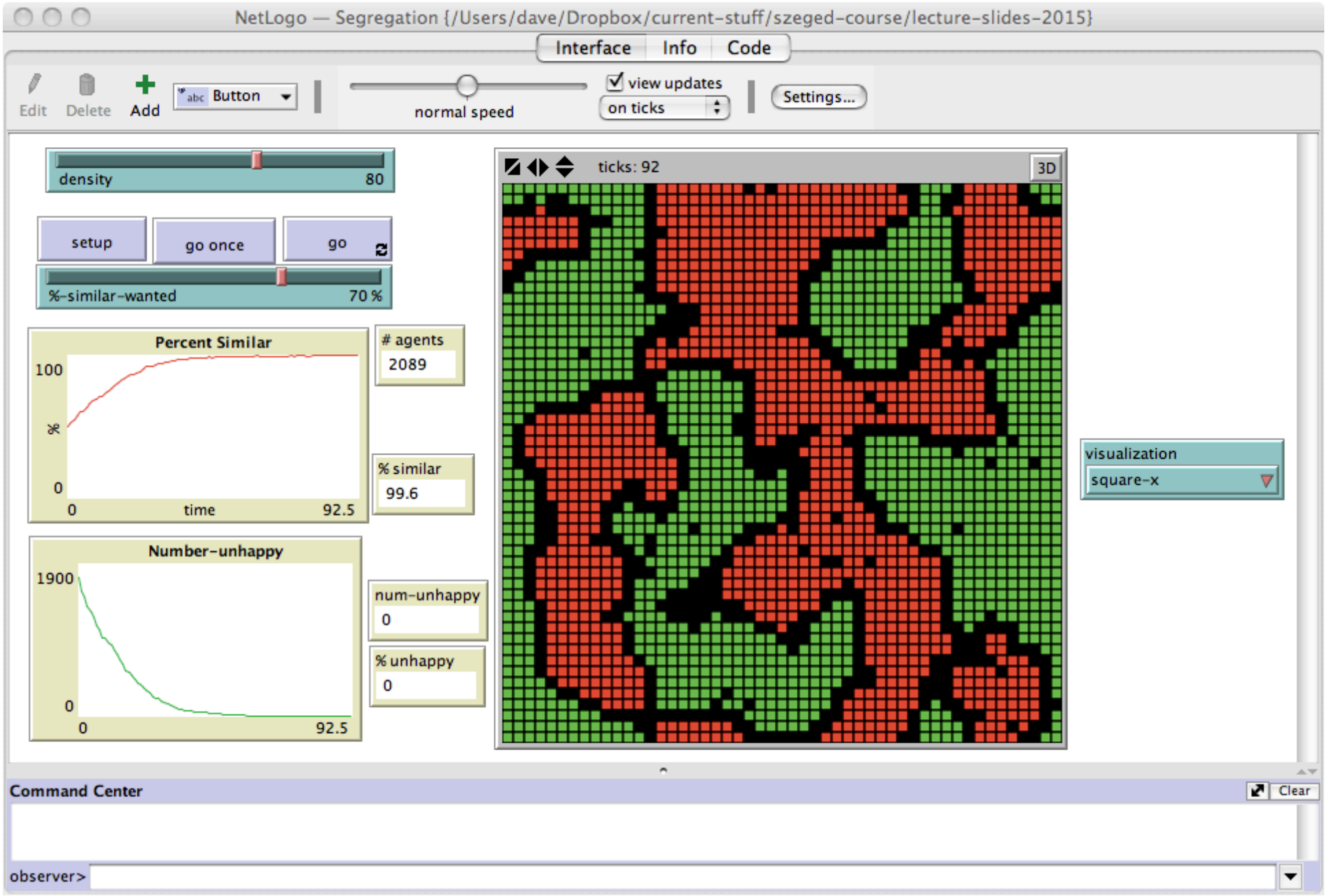
- Schelling observes:
  - Even comparatively "tolerant" agents (say T=35%) can produce high segregation
  - This means that if agents don't want to be in a significant minority => high segregation
  - Playing around with coins on a checkerboard produced counter-intuitive insights
  - Others can reproduce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

Computer simulation

- Schelling's model is simple
- Easy to reproduce using any computer language: a 2D array of agents, a loop that keeps moving unsatisfied agents
- Simple is good – "keep it simple stupid" (KISS)
- NetLogo comes with a built-in version of Schelling's segregation model
- We will look at this and do some experiments with it

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


NetLogo — Segregation {/Users/dave/Dropbox/current-stuff/szeged-course/lecture-slides-2015}
Interface
Info
Code
Edit
Delete
Add
Button
normal speed
view updates
on ticks
Settings...
density
80
setup
go once
go
%-similar-wanted
70 %
Percent Similar
100
%
0
0
time
92.5
agents
2089
% similar
99.6
Number-unhappy
1900
0
0
92.5
num-unhappy
0
% unhappy
0
ticks: 92
3D
visualization
square-x
Command Center
Clear
observer>

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 small random distance
  - Until empty cell found

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

Systematic scan of T

- NetLogo has tool called “BehaviourSpace”: automates systematic sets of runs, writing results to a CSV file
- Can load CSV into other statistical applications for visualisation
- Such a scan is often called a “sensitivity analysis” of model since we determine how sensitive outputs are to some input parameter (or set of parameters)

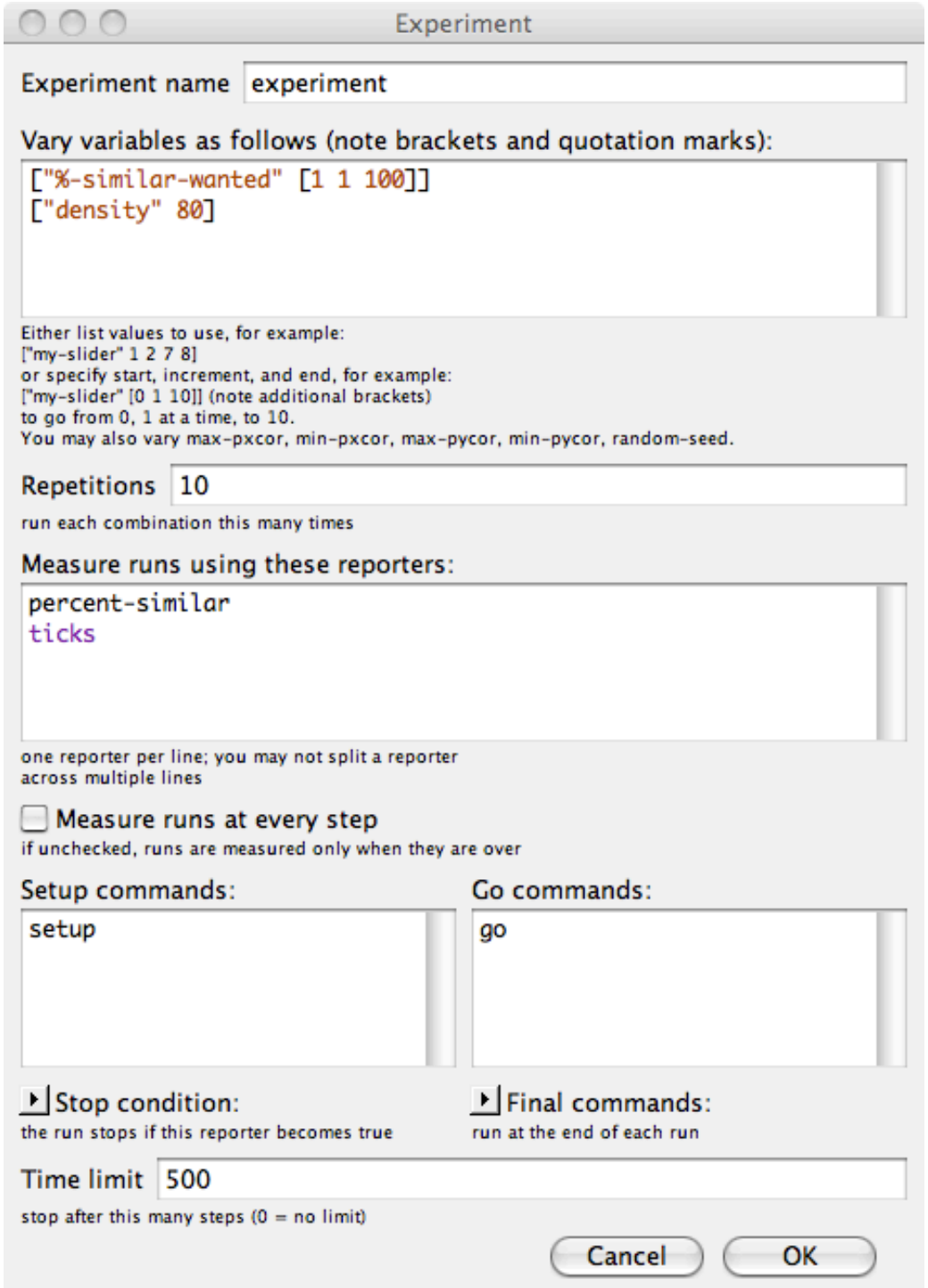
Systematic scan of T

- Since the model is not deterministic (it uses random numbers) it is wise to perform a number of runs for each value of T
- Since runs with high values of T never stop we need to put a cut-off at some number of steps
- We will do a scan of:
  - T values from 1..100 in increments of 1.
  - For each T value do 10 runs
  - Terminate any run at step 500 if it has not already terminated
  - Report %similar, and number of steps for each run

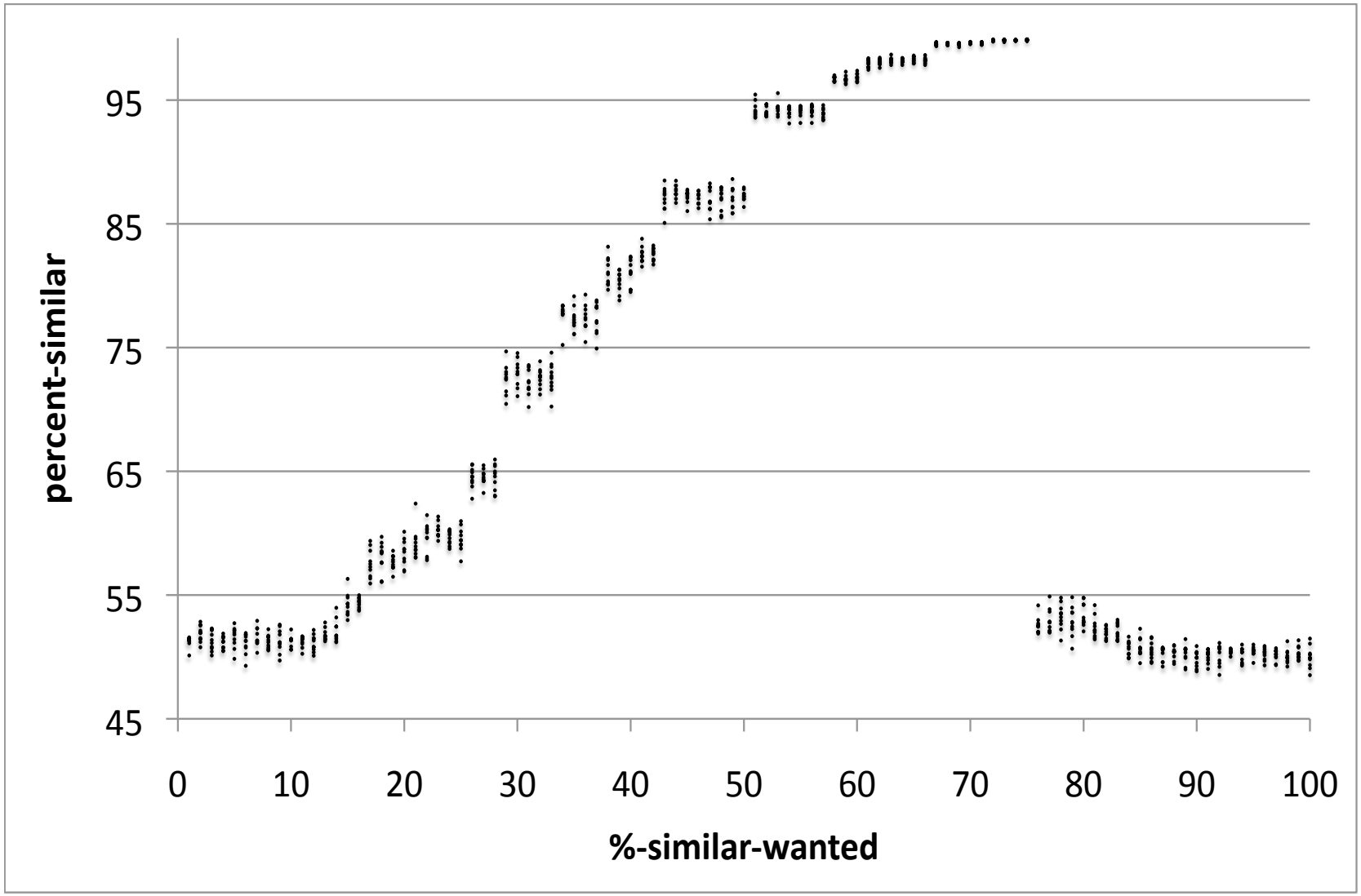
A NetLogo “experiment” defined under Tools>BehaviorSpace for the Segregation model.

It takes quite a while to run on a standard laptop (NetLogo is slow) but can take advantage of multiple cores on bigger machines

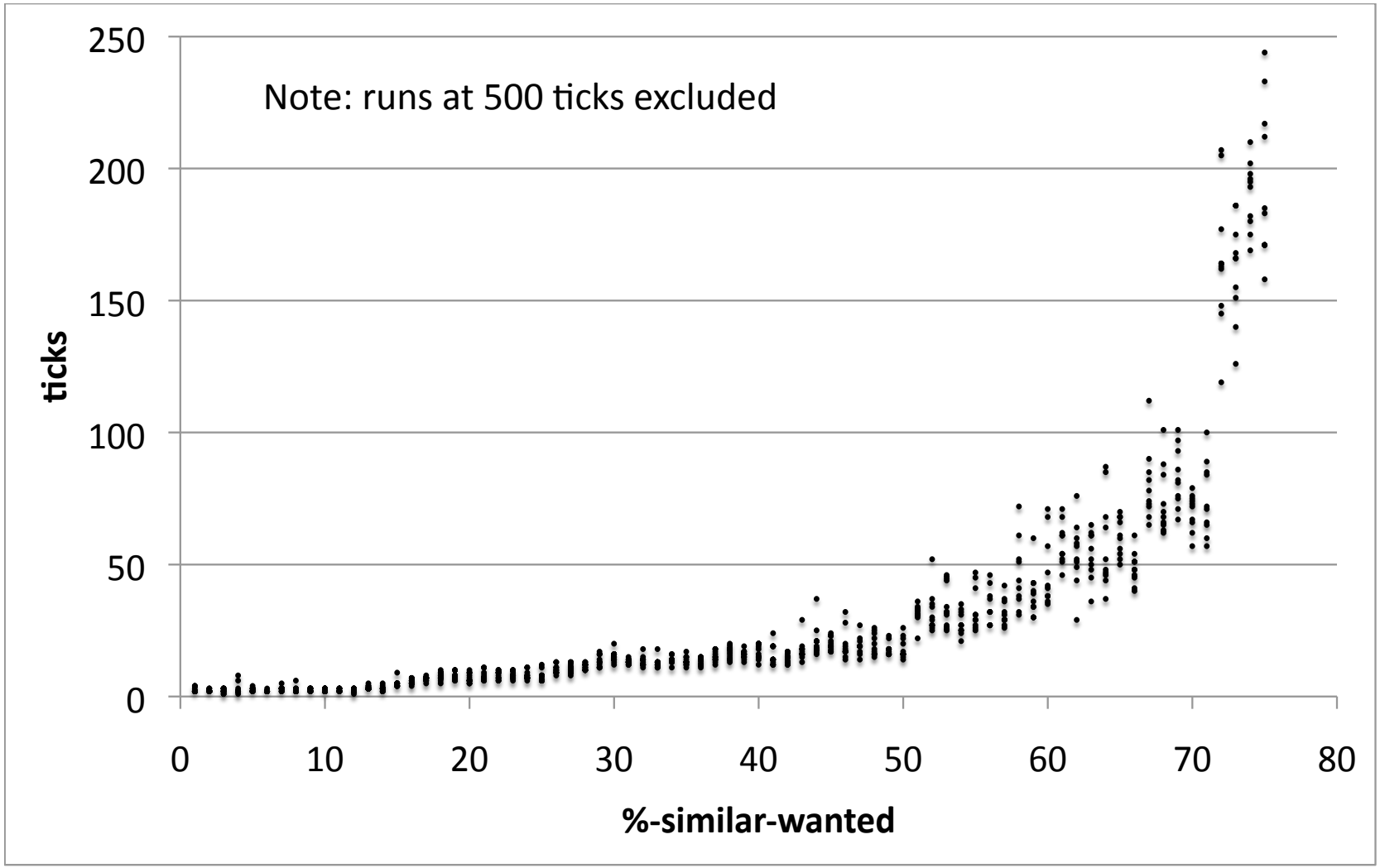
Since NetLogo is written in Java you can easily run experiments on servers without recompiling. “see headless”



Systematic scan percent-similar - result



Systematic scan ticks results



Results – what have we found?

- Schelling's observations about the model appear basically sound (not bad for a guy with a bunch of coins and checkerboard in 1969!)
- Non-linear relationship between T and segregation
- With T > 75% no stability emerges which actually leads to lower segregation due to very low satisfaction levels and constant moving => random
- We could of course explore other parameters such as number of agents, neighbourhood size, proportion of agent colours, distributions of satisfaction functions, more than 2 agent colours etc.

Aside: What do you think?

- Is this model too simple to be useful?
- Does it tell you anything you didn't already know?
- Can it help to understand the real world?
- Do you find it surprising that such a simple model is so highly cited?
- Can you think of a way of using the model in distributed systems design?

Schelling's model in P2P

- Singh and Haahr (2007) use Schelling's model as inspiration for a P2P clustering algorithm
- The paper is a little unclear, sketchy and not so easy to follow in parts
- However it does demonstrate how a social model has been applied to develop a P2P algorithm (I don't think it's been deployed)

- Singh, A. and Haahr, M. (2007) Decentralized clustering in pure p2p overlay networks using Schelling's model. In Communications, ICC'07. IEEE International Conference on, pages 1860–1866. IEEE, 2007.

Could a more rigorous and cleaner job be done building on this work?

Schelling's model in P2P

"Abstract—Clustering involves arranging a P2P overlay network's topology so that peers having certain characteristics are grouped together as neighbors. Clustering can be used to organize a P2P overlay network so that requests are routed more efficiently. The peers lack of a global awareness of the overlay network's topology in a P2P network makes it difficult to develop algorithms for clustering peers. This paper presents two decentralized algorithms for clustering peers. The algorithms are concrete realizations of of an algorithm called the abstract Schelling's algorithm (based on a model from sociology by Thomas Schelling) that can be used to create a family of self-* topology adaptation algorithms for P2P overlay networks. The proposed clustering algorithms are easy to implement, are not designed for clustering on a specific criteria and do not require separate algorithms to handle the flux of peers on the overlay network. The paper presents simulation results for applying the algorithm on random small-world topologies."

Schelling's model in P2P

Operation	Details
count(*property*)	The number of neighbors of a given node matching the given property.
add(*peers*)	Add the given peer or peers as a neighbor
drop(*peer*)	Drop the given peer as a neighbor
neighbor(*property*)	Returns a neighbor with the given property.
search(*property*)	Search for peers on the overlay network with the given property.

Specified required peer operations

Algorithm (two variants)

Algorithm 1 SelflessClustering Algorithm

$PNSP_{desired} \leftarrow$ % of neighbors with similar property desired
while *true* **do**
  $PNSP_{actual} \leftarrow \frac{\textbf{count}(\textit{same property})*100}{\textbf{count}(\textit{all})}$
  if $PNSP_{actual} < PNSP_{desired}$ **then**
    if count(*all*) > 1 **then**
      drop(**neighbor**(*different property* and **count**(*all*) > 1))
    end if
    add(**search**(*same property*))
  end if
  sleep(*delay*)
end while

Algorithm 2 SelfishClustering Algorithm

$PNSP_{desired} \leftarrow$ % of neighbors with similar property desired
while *true* **do**
  $PNSP_{actual} \leftarrow \frac{\textbf{count}(\textit{same property})*100}{\textbf{count}(\textit{all})}$
  if $PNSP_{actual} < PNSP_{desired}$ **then**
    drop(**neighbor**(*different property*))
  end if
  sleep(*delay*)
end while

PNSP = Percentage of Neighbours with Same Property.
They use small number of max links per peer (I think 5)

Main insight

From paper:

- "For both the clustering algorithms a low value of PNSPdesired (e.g., 10 to 20) is sufficient to achieve a substantial decrease in the number of clusters."
- "Disconnected Topology: A critical value of PNSPdesired (called PNSPcritical) was observed above which the overlay network's topology was disconnected. The value of PNSPcritical is different for different networks. The authors were not able to find any correlation between the network and the PNSPcritical value. A typical value of PNSPcritical is 40 for SelflessClustering algorithm and 20 for SelfishClustering algorithm."

A take home message?

- Suppose you want to do some simple self-organised clustering in a P2P network
- You don’t need to make the threshold of desired like neighbours high to get good results
- If you make it too high (too aggressive, greedy) then you will have > overheads and might get disconnected topologies
- You might select a low value (10%) to start with and if it seems to work try decreasing, if not try increasing
- In your final code you can put a comment saying you used Schelling’s model to guide you (which sounds better than “I just guessed”)

Schelling in “user models”

- Generate simulated social behaviour on which mobile P2P protocols can be evaluated
- Model stands in for real users, that are expected to exhibit clustering phenomena, for testing purposes
- Assume target population evidence “Shelling like” dynamics which protocols can exploit
  - L Vu, K Nahrstedt, M Hollick (2008) Exploiting Schelling behavior for improving data accessibility in mobile peer-to-peer networks. Proc. of the 5th Int. Conf. on Mobile and Ubiquitous Systems.

I have not studied this paper in detail – how good a use does it make of the Schelling model? Are you persuaded that this is a good way of evaluating systems?

Schelling in social networks

- Henry et al (2011) Analytic model of evolving social networks based on node “similarity”
- Shows *ANY* bias towards similarity will lead to highly segregated networks
- Based on their model and assumptions of course
- *Does this suggest we are all doomed to be trapped in “bubbles” of similar types?*

Henry AD, Prałat P, Zhang CQ (2011) Emergence of segregation in evolving social networks. Proc Natl Acad Sci USA 108(21):8605–8610

Thomas Schelling

- American political economist
- “Nobel prize” in economics (2005)
- Involved in post WWII Marshall Plan
- Major book: The Strategy of Conflict (1960)
- 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

Readings and Questions

- Gilbert et al (2005) Chapter 7 on CA's discusses the Schelling model and some other social models
- Schelling's (1971) paper is interesting to read
- Some questions:
  - What would happen if agents in Schelling's model optimise around their T value?
  - How sensitive are results based density (i.e. amount of empty space)?
  - What would happen if T values were assigned randomly to agents from a range between (0..100)?
  - How can we know a run will *never* stabilise?
  - The Netlogo version is synchronous (all agents decide to move at same time) what if agents moved one at a time (asynchronous)?