

# 3. Exploring Schelling's Segregation Model

Modelling Social Interaction in  
Information systems (MSIIS)

David Hales, University of Szeged

# Recap from last two lectures

- Nature of course (overview)
- Nature of models (many kinds not only prediction)
- Models are used in social science, models are used in computer science
- Social models can use computer simulation
- Can inform / inspire design of algorithms for engineering
- Even simple algorithms sometimes have to be executed and empirically analysed to understand what they do

# 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: It is interesting to note the political background in which social models come about and we will come back to this later*

# Schelling's 2D segregation model

- Bounded grid of cells
- 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 are the same colour
- If  $<T\%$  of neighbours are same colour then an agent is not satisfied
- Unsatisfied agents try to move to 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
- He considers equal white/black number of agents (69) placed randomly on a 13 x 16 grid
- Initially placing the agents randomly
- By hand he moves the agents until they are all satisfied and an equilibrium is reached
- His movement rule is a little vague but it involves picking up unsatisfied agents and placing them in the nearest empty cell that makes them satisfied
- He shows pictures 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  $T \leq 30\%$  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
- He also explores varying other parameters such as different proportions of colours and different  $T$  values for different colours
- We will not consider these latter aspects



# 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 Schelling's 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 bits, a loop that keeps moving unsatisfied agents
- Simple is good - remember "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
- Two input parameters: number of agents, T%
- Three output windows:
  - percent similar time series (segregation measure)
  - percent 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

Interface Info Code

Edit Delete Add abc Button

normal speed

view updates  
on ticks

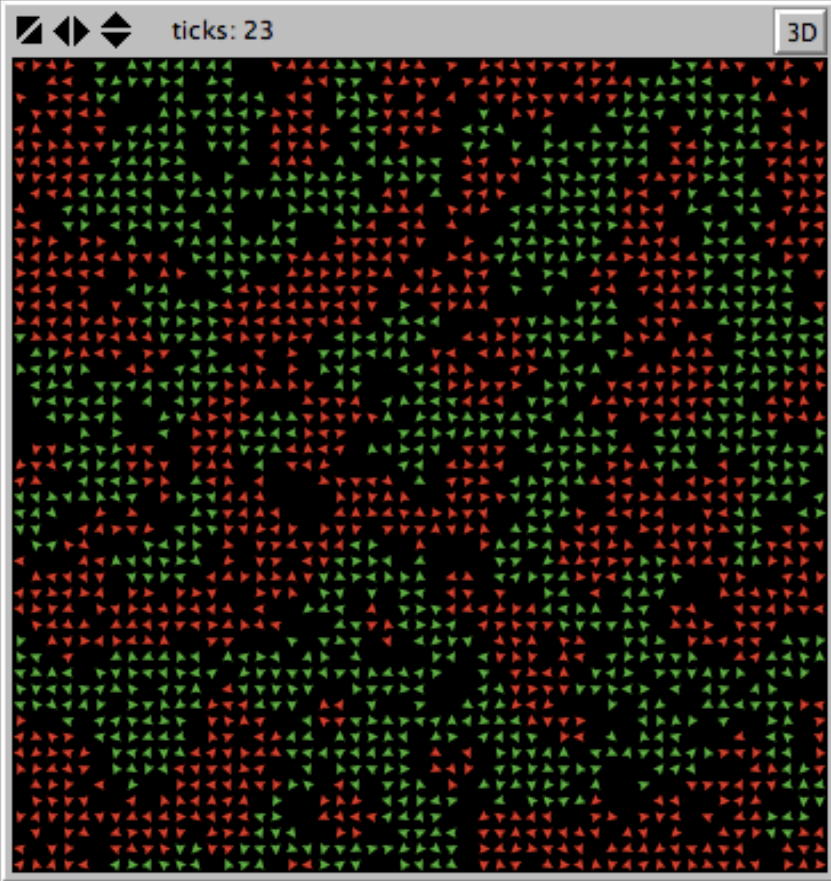
Settings...

number 2000

setup go

%similar-wanted 50%

ticks: 23 3D



Percent Similar

100

%

0

0 time 28.8

% similar 87.5

Percent Unhappy

100

%

0

0 time 28.8

% unhappy 0

Command Center

observer>

Clear

Detailed description: The image shows a NetLogo window titled "NetLogo — Segregation". At the top, there are three tabs: "Interface" (selected), "Info", and "Code". Below the tabs is a toolbar with "Edit", "Delete", and "Add" buttons, followed by a dropdown menu showing "abc Button". A speed slider is set to "normal speed". To the right, there is a checkbox for "view updates" which is checked, and a dropdown menu for "on ticks". A "Settings..." button is also present. The main workspace contains two sliders: "number" set to 2000 and "%similar-wanted" set to 50%. Below these are "setup" and "go" buttons. Two monitors are visible: "Percent Similar" showing a red line graph that rises from approximately 75% to 100% over time, with a current value of 87.5; and "Percent Unhappy" showing a green line graph that falls from 100% to 0% over time, with a current value of 0. To the right of the monitors is a large 3D view of a square world filled with green and red triangles, representing the spatial distribution of agents. The world is currently mixed. At the bottom, there is a "Command Center" with a text input field containing "observer>" and a "Clear" button.

# NetLogo implementation

- 51 x 51 grid (wrapped) = 2601 cells (called patches)
- Agents (called turtles) placed on random patches. Divided between colours 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 number 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

# Observations

- With default value  $N=2000$ :
  - $T < 20\%$  tends to produce %similar  $< 60\%$
  - $T > 30\%$  tends to produce %similar  $> 70\%$
  - $T > 80\%$  things never seem to stabilise
  - $T < 75\%$  things seem to stabilise quickly
- To get an idea of how  $T$  affects %similar (segregation) and %unhappy (stability) we need to do a systematic set of simulation runs

# Systematic scan of T

- Usefully, NetLogo has a built-in tool called “BehaviourSpace” that automates systematic sets of runs, writing results to a CSV file
- We can then take the results file and visualise it using some statistical application
- If the sim. was in some other language we would do this by having a loop that changed the T parameter and ran the sim. Outputting results to a file
- 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 0..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 %unhappy, %similar 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”

Experiment

Experiment name

Vary variables as follows (note brackets and quotation marks):

```
[\"number\" 2000]  
[%-similar-wanted\" [0 1 100]]
```

Either list values to use, for example:  
[\"my-slider\" 1 2 7 8]  
or specify start, increment, and end, for example:  
[\"my-slider\" [0 1 10]] (note additional brackets)  
to go from 0, 1 at a time, to 10.  
You may also vary max-pxcor, min-pxcor, max-pycor, min-pycor, random-seed.

Repetitions   
run each combination this many times

Measure runs using these reporters:

```
percent-similar  
percent-unhappy
```

one reporter per line; you may not split a reporter across multiple lines

Measure runs at every step  
if unchecked, runs are measured only when they are over

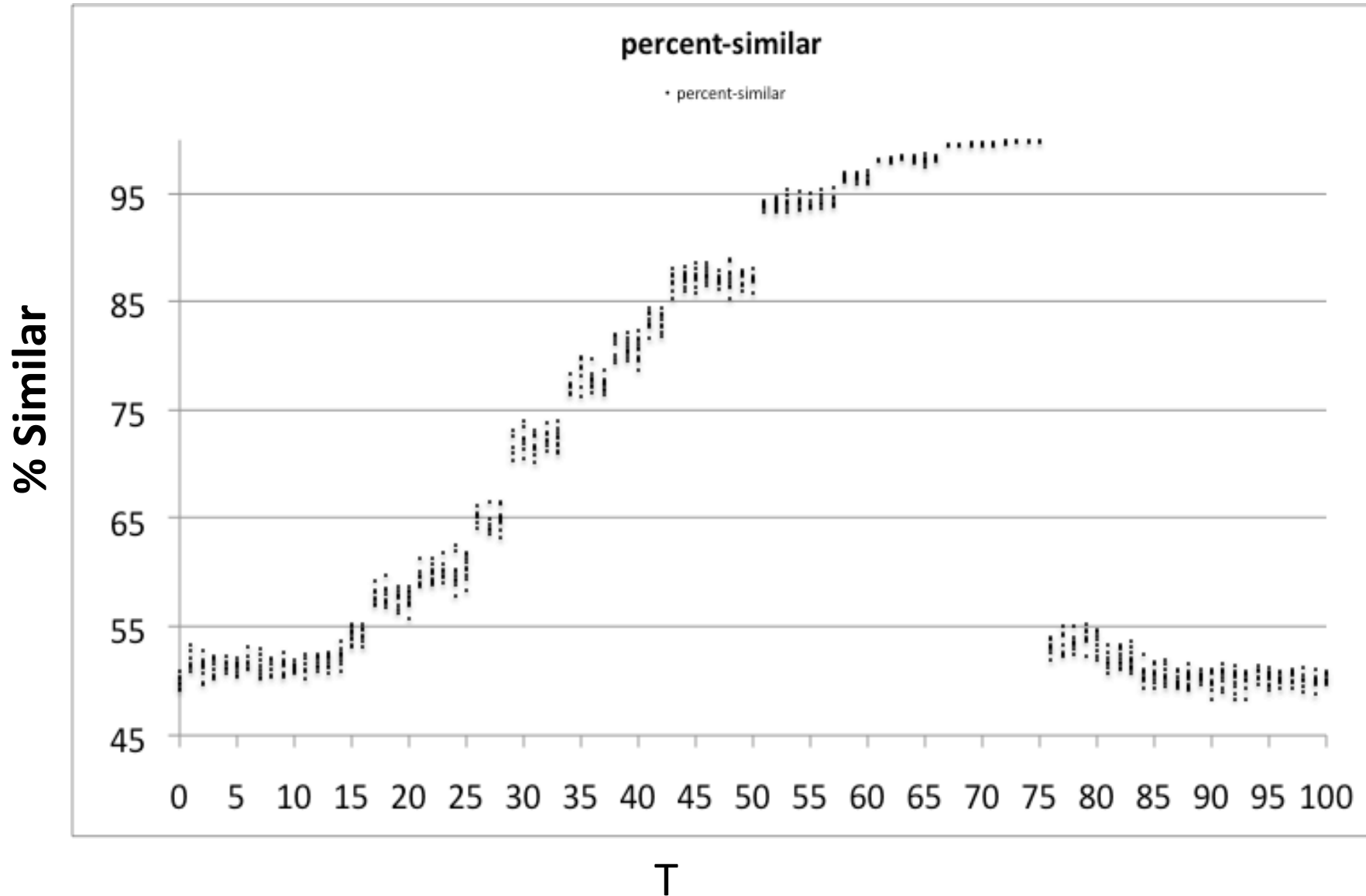
Setup commands:  Go commands:

Stop condition:  Final commands:   
the run stops if this reporter becomes true run at the end of each run

Time limit   
stop after this many steps (0 = no limit)

Cancel OK

# Systematic scan of T - result



# 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.

# What does it mean?

- The results from the simulations are logical deductions from the assumptions of the model (they are facts)
- Interpretations of the model related to the real world are speculative, hypotheses, hunches
- The model is a thought experiment giving a qualitative rather than quantitative result:
  - Even tolerant individuals, given freedom of movement, can lead to highly segregated outcomes
  - Individuals may get a macro outcome that is not what they appear to desire at a micro level

# What does it mean?

- The model does not “prove” anything about real world segregation
- It demonstrates an abstract mechanism that produces a counter-intuitive result
- It challenges the assumption that high segregation *must* be because individuals are highly intolerant
- By making a computer model we formally specify the assumptions and allow for rigorous comparison of different variants
- We also communicate it very clearly and formally. There is no vagueness allowed in a computer model

# Caveats and dangers (my opinion)

- Schelling's model is very widely cited
- I've seen statements like:
  - Schelling's model explains racial segregation
  - Schelling proved that segregation emerges from individual behaviour
- But this is over claiming what the model teaches us
- Some try to map the model onto actual segregation patterns – which is interesting but potentially misguided
- The danger is that if some “fit” can be found then people will believe that they have explained the actual segregation
- See: Hatna and Benenson (2012) The Schelling Model of Ethnic Residential Dynamics <http://jasss.soc.surrey.ac.uk/15/1/6.html> (empirical - Jew and Arab residential in Israel)
- Interesting paper - read intro / conclusion.

# Caveats and dangers (my opinion)

- Some believe that using formal models means that social science becomes “scientific” and not “ideological” or “political” (see “Scientism”)
- But any model contains assumptions that are ideological and political *even* if those working with the model are not aware of it
- For example, the Schelling model:
  - focuses only on individual preferences and behaviour
  - Ignores history, wealth, social norms, culture etc.
- Consider the specific cultural and ideological norms related to race in late 60’s USA in which Schelling was working.



# 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

**From introduction of paper (caveats!):**

“In 1969, Thomas Schelling, an economist, proposed a model to explain the existence of segregated neighborhoods in America. He observed that the appearance of such segregated neighborhoods is caused neither by a central authority, nor by the desire of people to stay away from dissimilar people; instead, it is the cumulative effect of simple actions (moves) by individuals who want at least a certain proportion of their neighbors to be similar to themselves. Schelling's model is decentralized and self-maintaining in nature. This makes it attractive for topology adaptation in dynamic environments such as pure P2P networks, which lack a central authority.”

# Schelling's model in P2P

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

Specified required peer operations

# Algorithm (two variants)

---

## Algorithm 1 SelflessClustering Algorithm

---

```
PNSPdesired ← % of neighbors with similar property
desired
while true do
  PNSPactual ←  $\frac{\text{count}(\text{same property}) * 100}{\text{count}(\text{all})}$ 
  if PNSPactual < PNSPdesired 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

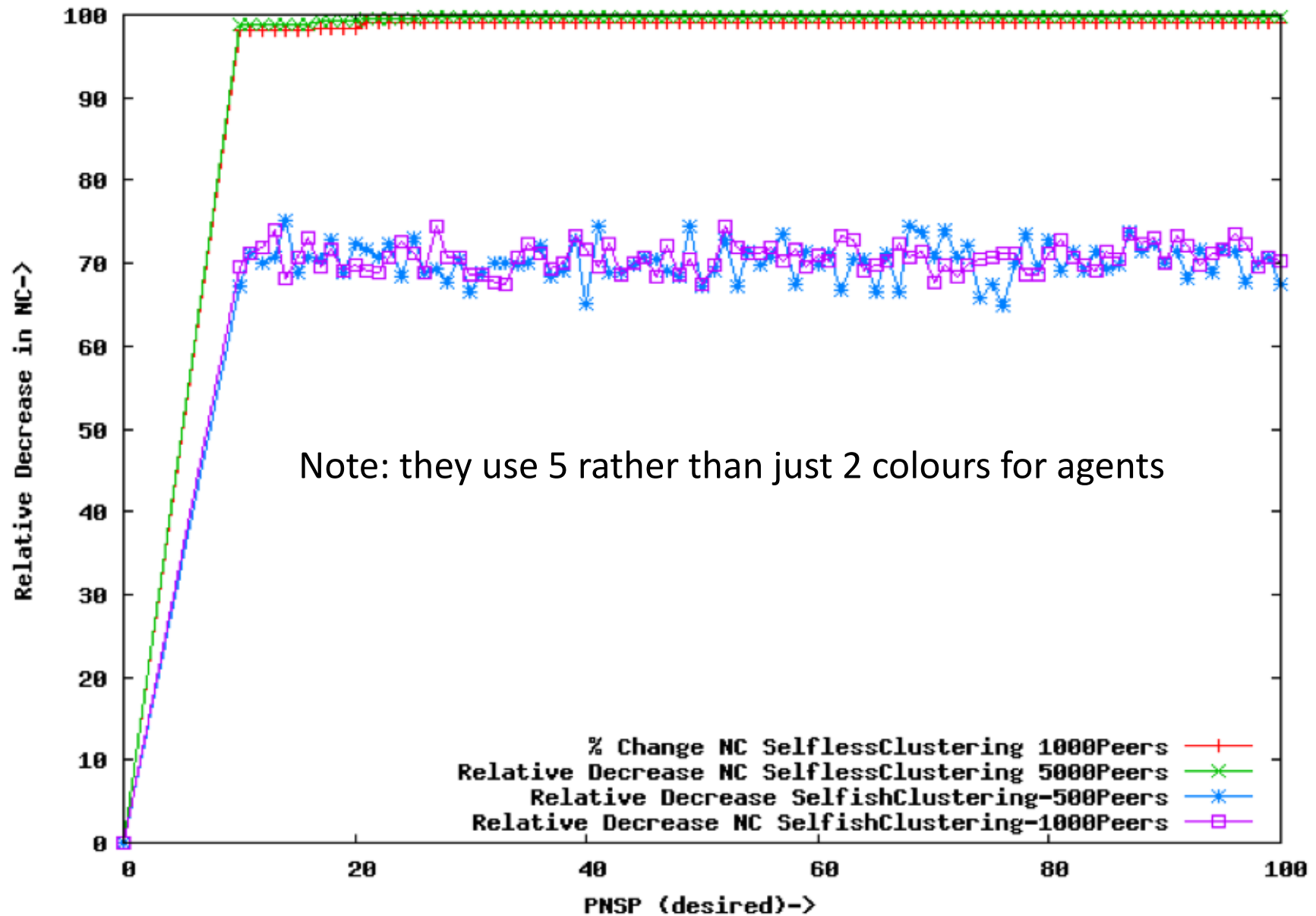
---

```
PNSPdesired ← % of neighbors with similar property
desired
while true do
  PNSPactual ←  $\frac{\text{count}(\text{same property}) * 100}{\text{count}(\text{all})}$ 
  if PNSPactual < PNSPdesired then
    drop(neighbor(different property))
  end if
  sleep(delay)
end while
```

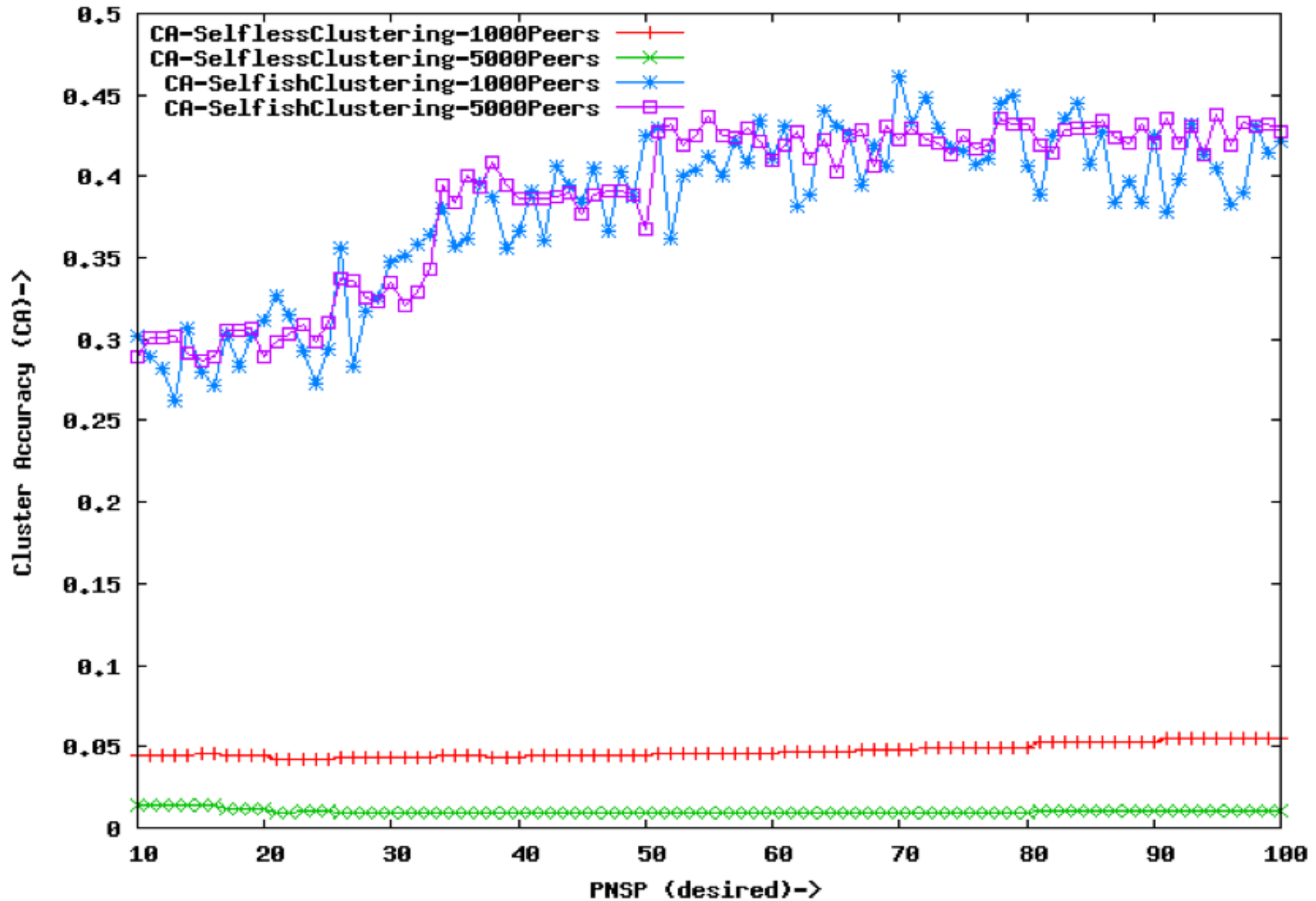
---

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

# Some results (NC = num. clusters)



# Some results (not so clear)



# 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. The SelflessClustering algorithm rearranges the overlay network topology to have approximately five clusters, one for each type of peers. The SelfishClustering algorithm that provides approximately a 70 % decrease in the number of clusters. However the cluster accuracy is far less for SelflessClustering clustering algorithm. So if the cluster accuracy is a concern then SelfishClustering algorithm is a good choice whereas if the decrease in the number of clusters is a concern then SelfishClustering algorithm is a good choice.”



# Interesting observation

From paper:

- “Disconnected Topology: A critical value of  $PNSP_{desired}$  (called  $PNSP_{critical}$ ) was observed above which the overlay network’s topology was disconnected. The value of  $PNSP_{critical}$  is different for different networks. The authors were not able to find any correlation between the network and the  $PNSP_{critical}$  value. A typical value of  $PNSP_{critical}$  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
- If you were hacking something like this 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!)

# Using Schelling for “user models”

- Another way that Schelling has been applied is in generating simulated social behaviour on which mobile P2P protocols can be evaluated
- Hence the model stands in for real users, that are expected to exhibit clustering phenomena, for testing purposes
- The assumption here is that the target population will evidence “Shelling like” dynamics which protocols can be designed to exploit
- L Vu, K Nahrstedt, M Hollick (2008) Exploiting Schelling behavior for improving data accessibility in mobile peer-to-peer networks. Proceedings of the 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services.

**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?**

# Is Schelling's model a Cellular Automata?

- Since agents move about and search on the grid it is not a strict CA (which are fixed cells)
- Often people prefer to call it a simple Agent-Based Model (ABM)
- ABM are a generalisation of a CA where agents can interact in more complex ways in a shared environment
- We will revisit some of these distinctions later (Individual-based model, ABM, artificial societies, system dynamics, Monte Carlo simulations, micro model, macro model etc.)

# 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) ?



# 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 (for fun):
  - What would happen if you made the agents in Schelling's model optimise around their T value?
  - How sensitive are results based on the number of agents (i.e. amount of empty space)?
  - What would happen if T values were assigned randomly to agents between (0..100)?
  - Why is the model so highly cited (in social science)?