SwarmFest 2013

Agent-based models for exploring social complexity,

with a final example on simulating groups' behavior via rules and scenarios

Pietro Terna

Department of Economics, Sociology, Mathematics and Statistics
University of Torino

web.econ.unito.it/terna or goo.gl/y0zbx

the presentation is at goo.gl/UMs4C

Accompanying material:

- P. Terna (2013), A Complex Lens for Economics, or: About Ants and their Anthill, in “Spazio filosofico”, 7, pp. 167-177 http://www.spaziofilosofico.it/wp-content/uploads/2013/01/Terna-English.pdf

- P. Terna (2013), Learning agents and decisions: new perspectives, in "Law and Computational Social Science", 1, http://eco83.econ.unito.it/terna/materiale/terna_def.pdf

Basics

A note: the slides contain several references; you can find them in a draft paper, on line at **http://goo.gl/ryhyF**

Artifacts of social systems

Leibniz (*xi. De scientia universali seu calculo philosophico*): … *quando orientur controversiae, non magis disputatione opus erit inter duos philosophos, quam inter duos computistas. Sufficiet enim calamos in manus sumere sedereque ad abbacos et sibi mutuo (...) dicere, calculemus*

Calculemus or ... ***Simulemus***

... plus complexity, bounded rationality, chaos ...

Anderson 's 1972 paper "More is different" as a manifesto.

(p.393) The reductionist hypothesis may still be a topic for controversy among philosophers, but among the great majority of active scientists I think it is accepted without questions. The workings of our minds and bodies, and of all the animate or inanimate matter of which we have any detailed knowledge, are assumed to be controlled by the same set of fundamental laws, which except under certain extreme conditions we feel we know pretty well.

(…)The main fallacy in this kind of thinking is that the reductionist hypothesis does not by any means imply a "constructionist" one: The ability to reduce everything to simple fundamental laws does not imply the ability to start from those laws and reconstruct the universe.

Anderson 's 1972 paper "More is different" as a manifesto.

The constructionist hypothesis breaks down when confronted with the twin difficulties of scale and complexity. The behavior of large and complex aggregates of elementary particles, it turns out, is not to be understood in terms of a simple extrapolation of the properties of a few particles. Instead, at each level of complexity entirely new properties appear, and the understanding of the new behaviors requires research which I think is as fundamental in its nature as any other.

(p.396) In closing, I offer two examples from economics of what I hope to have said. Marx said that quantitative differences become qualitative ones, but a dialogue in Paris in the 1920's sums it up even more clearly:

FITZGERALD: The rich are different from us.
HEMINGWAY: Yes, they have more money.

Rosenblueth and Wiener's 1945 paper, "The Role of Models in Science" , as a "manual" from the founders of cybernetics.

(p. 317) A distinction has already been made between material and formal or intellectual models. A material model is the representation of a complex system by a system which is assumed simpler and which is also assumed to have some properties similar to those selected for study in the original complex system. A formal model is a symbolic assertion in logical terms of an idealized relatively simple situation sharing the structural properties of the original factual system.

Material models are useful in the following cases. a) They may assist the scientist in replacing a phenomenon in an unfamiliar field by one in a field in which he is more at home.

(…) b) A material model may enable the carrying out of experiments under more favorable conditions than would be available in the original system.

Rosenblueth and Wiener's 1945 paper, "The Role of Models in Science" , as a "manual" from the founders of cybernetics.

(p. 319) It is obvious, therefore, that the difference between open-box and closed-box problems, although significant, is one of degree rather than of kind. All scientific problems begin as closed-box problems, i.e., only a few of the significant variables are recognized. Scientific progress consists in a progressive opening of those boxes. The successive addition of terminals or variables, leads to gradually more elaborate theoretical models: hence to a hierarchy in these models, from relatively simple, highly abstract ones, to more complex, more concrete theoretical structures

A comment: this is the main role of simulation models in the complexity perspective, **building material models as artifacts running into a computer**, having always in mind to go toward "more elaborate theoretical models".

Finally, quoting another paper of the special issue referred above, that of prof.W. Brian Arthur

(…) a second theme that emerged was that of making models based on more realistic cognitive behavior. Neoclassical economic theory treats economic agents as perfectly rational optimizers. This means among other things that agents perfectly understand the choices they have, and perfectly assess the benefits they will receive from these.

(…) Our approach, by contrast, saw agents not as having perfect information about the problems they faced, or as generally knowing enough about other agents' options and payoffs to form probability distributions over these. This meant that agents need to cognitively structure their problems—as having to 'make sense' of their problems, as much as solve them.

A comment: So we need to include learning abilities into our agents.

Moving to models

We can now move to models:
in the traditional way

or in the *new* perspective of

the material models of cybernetics founders

the computational artifacts of the agent-based simulation models.

quite close

Following Ostrom (1988), and to some extent, Gilbert and Terna (2000), in social science, excluding material (analogue) models, we traditionally build models as simplified representations of reality, using:

i. Verbal Argumentation and
ii. Mathematical Equations, typically with Statistics and Econometrics

Now we have computational tools:

- Equilibrium Models
- Game Theory
- System Dynamics

close to ii.

- Serious Gaming
- Agent-Based Simulation

iii.

Computer simulation (mainly agent-based one) can combine the extreme flexibility of a computer code – where we can create agents who act, make choices, and react to the choices of other agents and to modifications of their environment – and its intrinsic computability.

New model army

Economics after the crisis

Efforts are under way to improve macroeconomic models

Jan 19th 2013 | WASHINGTON, DC | From the print edition

The Economist, Jan 19th 2013

http://www.economist.com/node/21569752

Improving DSGE models is the obvious way to take the lessons of the crisis on board. But others exist too. "Agent-based modelling" tries to depict the transactions that might occur in an actual economy. These models are populated by millions of agents that gradually alter the economy as they interact with each other.

However, reality is intrinsically agent-based, not equation-based.

At first glance, this is a strong criticism. Why reproduce social structures in an agent-based way, following (iii), when science applies (ii) to describe, explain, and forecast reality, which is, per se, too complicated to be understood?

The first reply is again that we can, with agent-based models and simulation, produce **artifacts** (the 'material model') of actual systems and “play” with them, i.e., **showing the consequences of perfectly known ex-ante hypotheses and agent behavioral design and interaction**; then we can apply statistics and econometrics to the outcomes of the simulation and compare the results with those obtained by applying the same tests to actual data.

In this view, simulation models act as a sort of magnifying glass that may be used to better understand reality.

The second reply is that, relying again on Anderson (1972), we know that complexity arises when agents or parts of a whole act and interact and the quantity of involved agent is relevant.

Furthermore, following Villani (2006, p.51), “Complex systems are systems whose complete characterization involves more than one level of description.”

To manage complexity, one mainly needs to build models of agents.

As a stylized example, consider ants and their ant-hill: Two levels need to be studied simultaneously to understand the (emergent) dynamic of the ant-hill based on the (simple) behaviors of the ants.

However, agent-based simulation models have severe weaknesses, primarily arising from:

- The difficulty of fully understanding them without studying the program used to run the simulation;
- The necessity of carefully checking computer code to prevent generation of inaccurate results from mere coding errors;
- The difficulty of systematically exploring the entire set of possible hypotheses in order to infer the best explanation. This is mainly due to the inclusion of behavioral rules for the agents within the hypotheses, which produces a space of possibilities that is difficult if not impossible to explore completely.

Some replies:

- Swarm (provisionally http://goo.gl/tAEJL; stable address, temporary out of order: http://www.swarm.org) a project that started within the Santa Fe Institute (first release 1994) and that represents a milestone in simulation;

- Swarm has been highly successful, being its protocol intrinsically the basis of several recent tools; for an application of the Swarm protocol in Python, see my SLAPP, Swarm Like Agent Protocol in Python at http://eco83.econ.unito.it/slapp

- Many other tools have been built upon the Swarm legacy, such as Repast, Ascape, Mason, **JAS** and also by simpler, but important tools, such as NetLogo and StarLogoTNG.

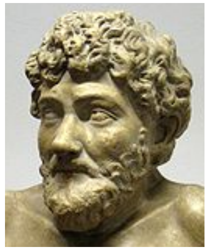
λ

Technicalities:
Why Swarm, Python SLAPP and why NetLogo?

λ

SLAPP, or Swarm-Like Agent Protocol in Python, is a simplified implementation of the original Swarm protocol (http://www.swarm.org), choosing Python as a simultaneously simple and complete object-oriented framework.

SLAPP contains AESOP



AESOP (Agents and Emergencies for Simulating Organizations in Python), written upon SLAPP as a simplified way to describe and generate interaction within artificial agents:

- *bland* agents (simple, unspecific, basic, insipid, …) doing basic actions;
- *tasty* agents (specialized, with given skills, acting in a discretionary way, …), playing specify roles into the simulation scenario.

SLAPP is also useful:

- for didactical reasons, applying a such rigorous and simple object oriented language as Python
- to build models upon transparent code: Python does not have hidden parts or feature coming *from magic*, it has no obscure libraries
- to leverage the openness of Python
- to apply easily the SWARM protocol

The SWARM protocol

SLAPP is **a demonstration that we can easily implement the Swarm protocol** [Minar, N., R. Burkhart, C. Langton, and M. Askenazi (1996), *The Swarm simulation system: A toolkit for building multi-agent simulations*. Working Paper 96-06-042, Santa Fe Institute, Santa Fe (*)] **in Python**

(*) http://www.santafe.edu/media/workingpapers/96-06-042.pdf

Key points (quoting from that paper):

- *Swarm defines a structure for simulations, a framework within which models are built.*
- *The core commitment is to create a discrete-event simulation of multiple agents using an object-oriented representation.*
- *To these basic choices Swarm adds the concept of the "**swarm**," a **collection of agents** with a **schedule of activity**.*

Swarm = a library of functions and a **protocol**

modelSwarm

create agents

create actions

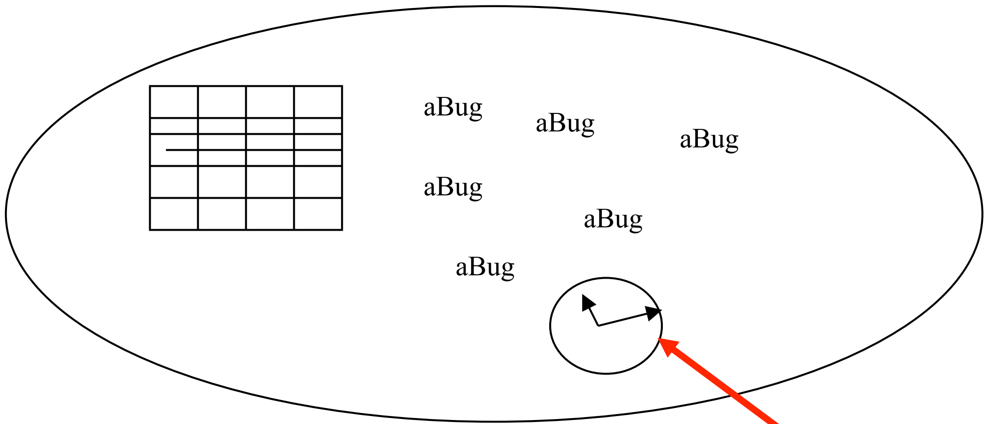
run modelSwarm

Bug

aBug

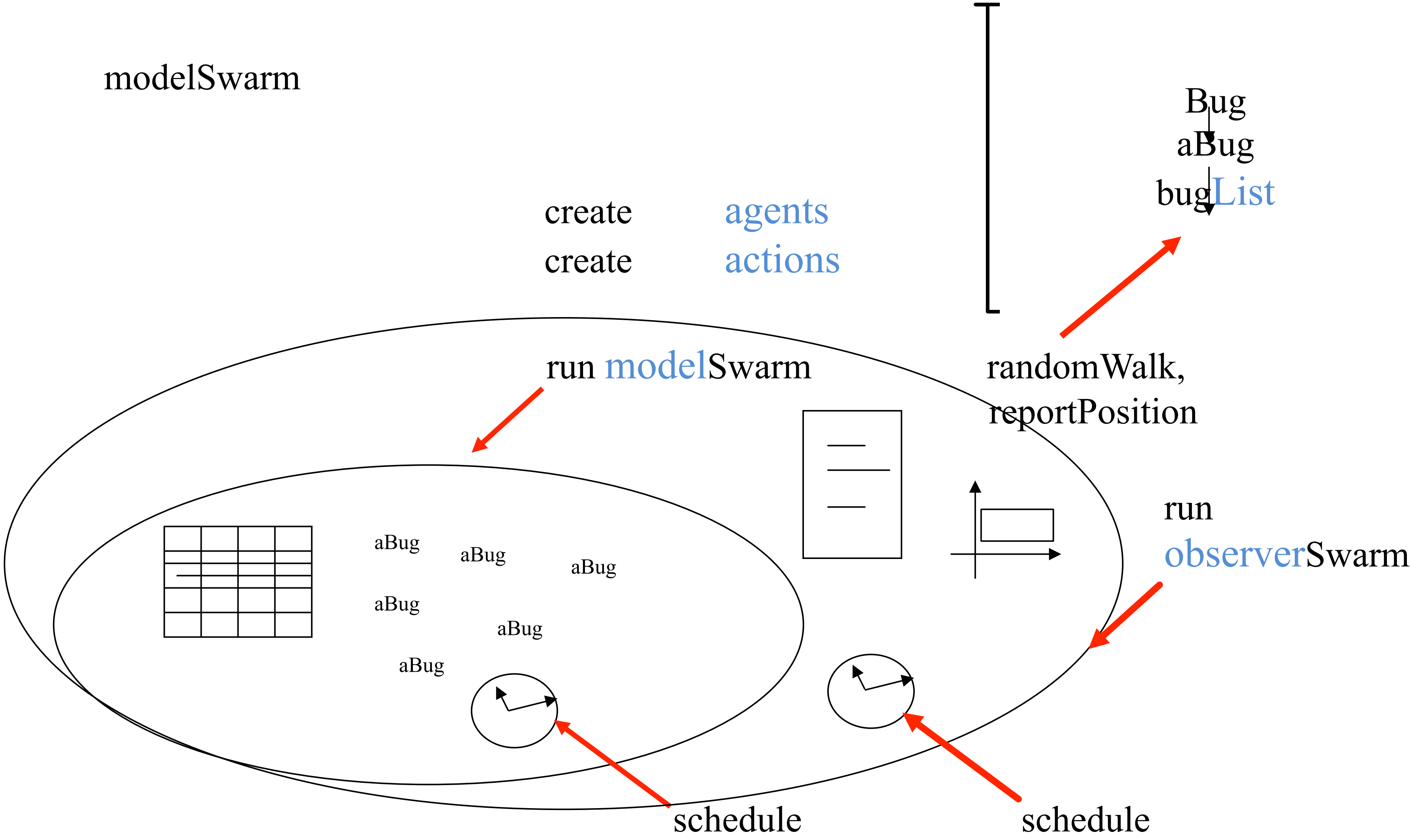
bugList

randomWalk,
reportPosition

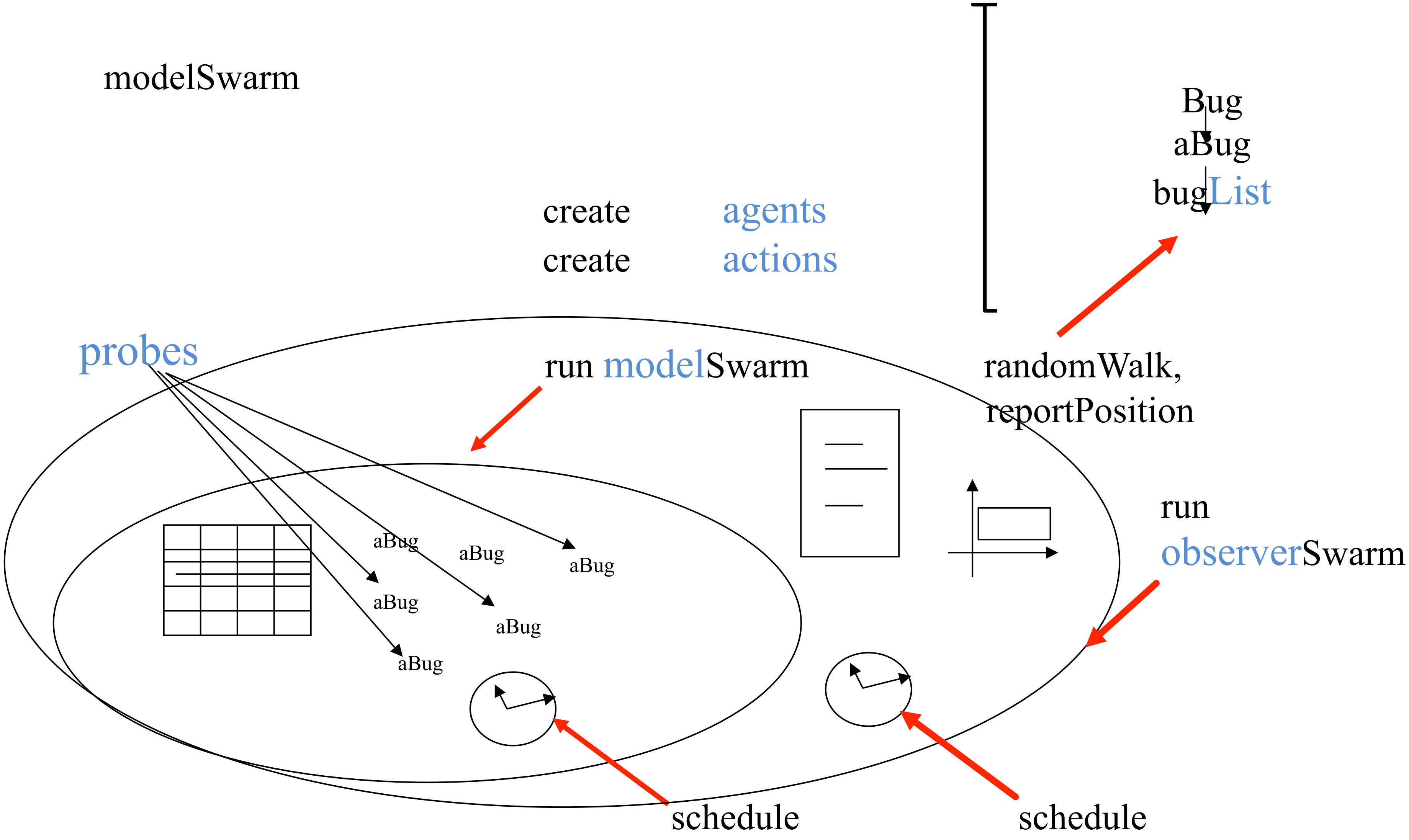


schedule

Swarm = a library of functions and a **protocol**



Swarm = a library of functions and a **protocol**



Always in technicalities … why NetLogo?

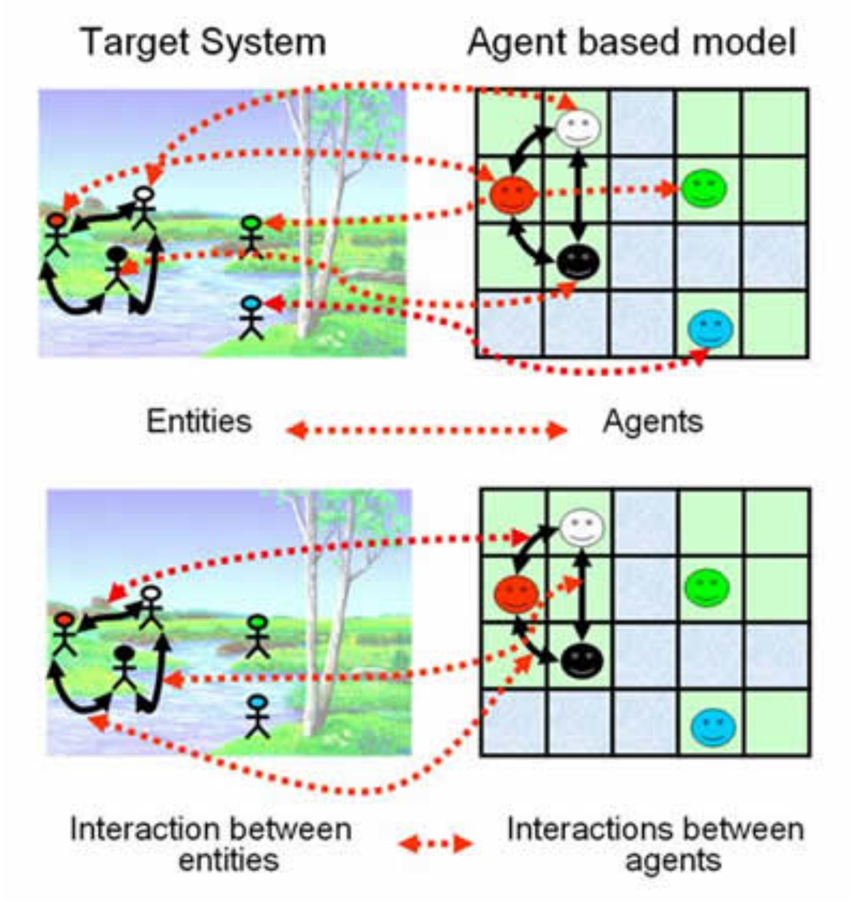
NetLogo is highly diffusing as a rigorous and easy tool, especially useful for prototyping and when we need advanced graphical capabilities

Limits are in coping with the design of complex experiments (and with huge numbers of agents)

Moving to computation

From Bruce Edmonds, Agent-based social simulation

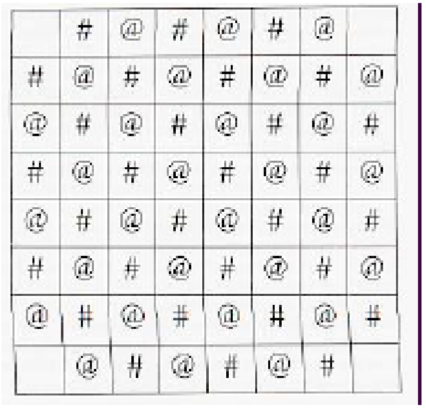
www.methods.manchester.ac.uk/methods/abss/

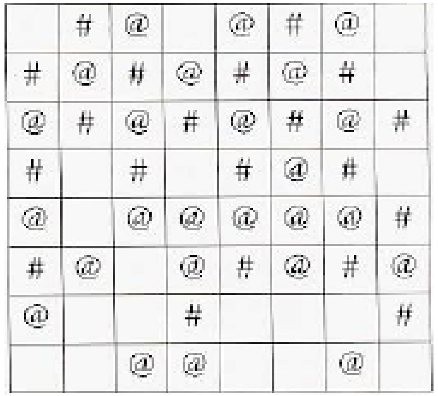


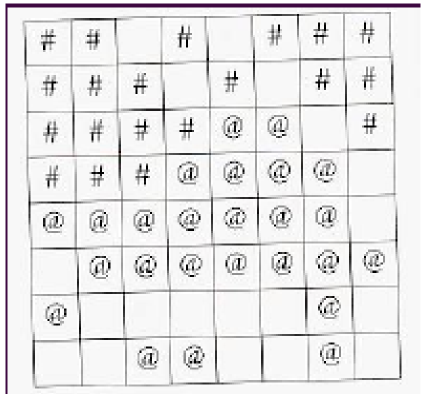
Finally, the importance of calculating: our complex system models live mainly in their computational phase and require calculating facilities more and more powerful.

Schelling model and random mutations

The well known segregation model from prof.Schelling has been initially solved moving dimes and pennies on a board.

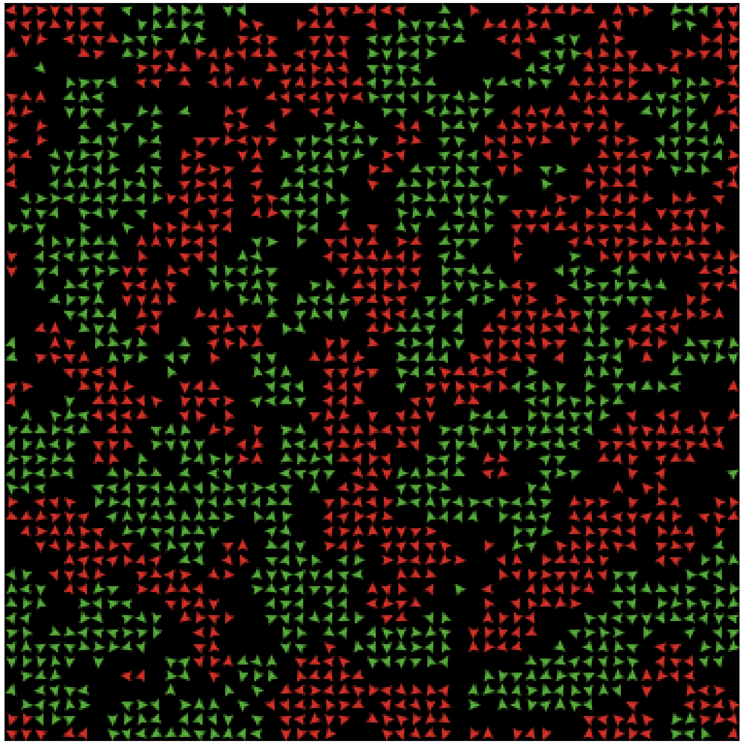


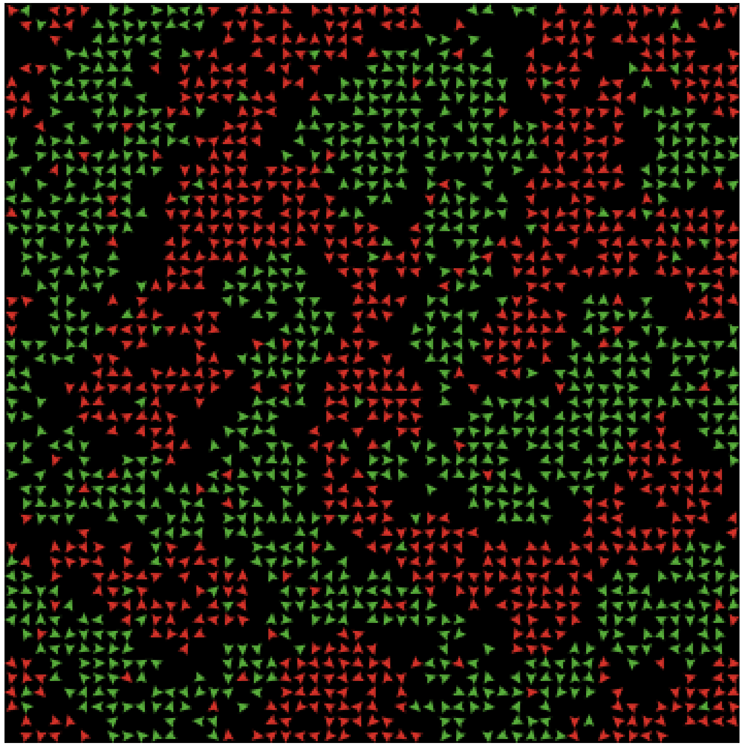




These pictures are from a presentation of Eileen Kraemer, http://www.cs.uga.edu/~eileen/fres1010/Notes/fres1010L4v2.ppt

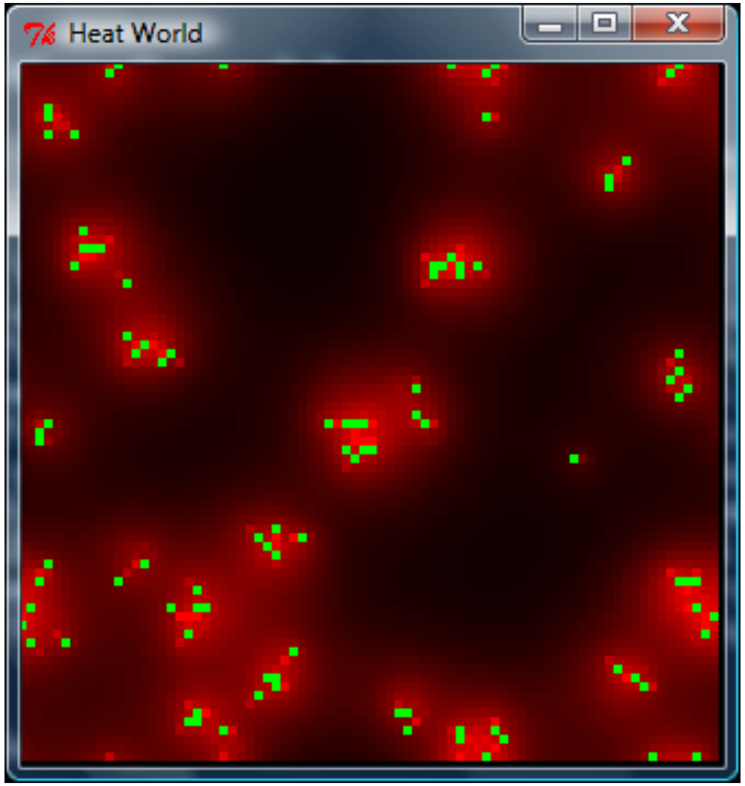
However, if you want to check the survival of the color islands in the presence of random mutations in agents (from an idea of prof.Nigel Gilbert), you need to use a computer and a simulation tool (NetLogo, in this case).



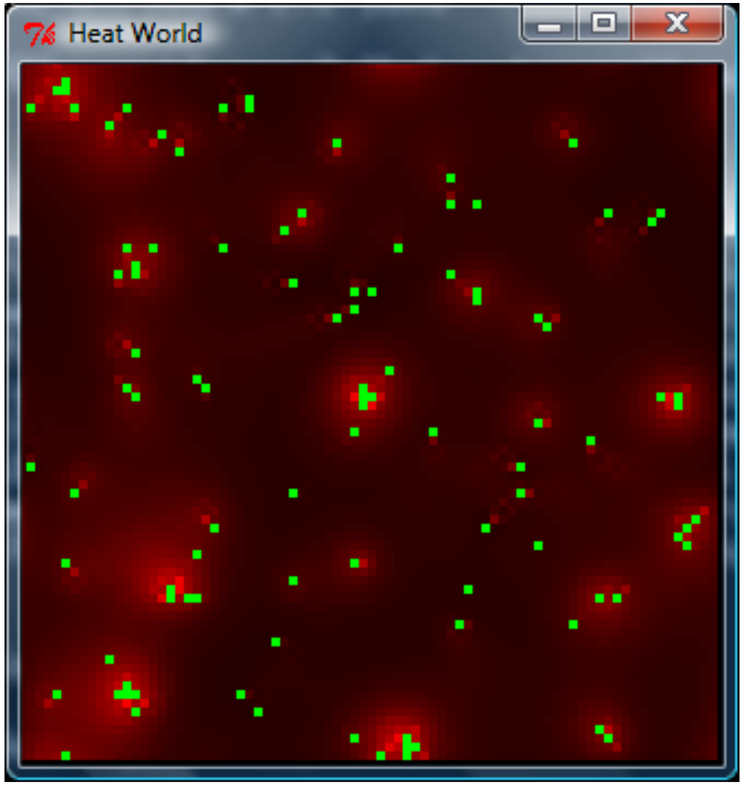


In the case of the test model of Swarm, the so called heatBugs model, you can have agents (i) with a preference with high temperature or with a part of them being adverse to it.; they generate warmth moving; when they are comfortable, they reduce movement; you have to make a lot of computations to obtain the first and the second emergent results below.

high temperature preference

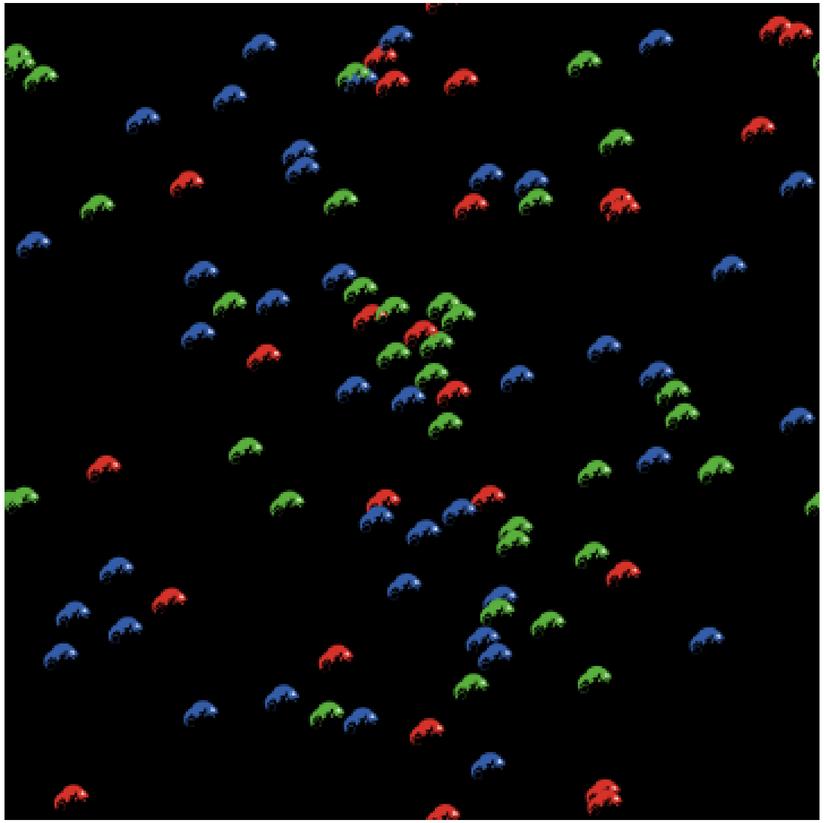


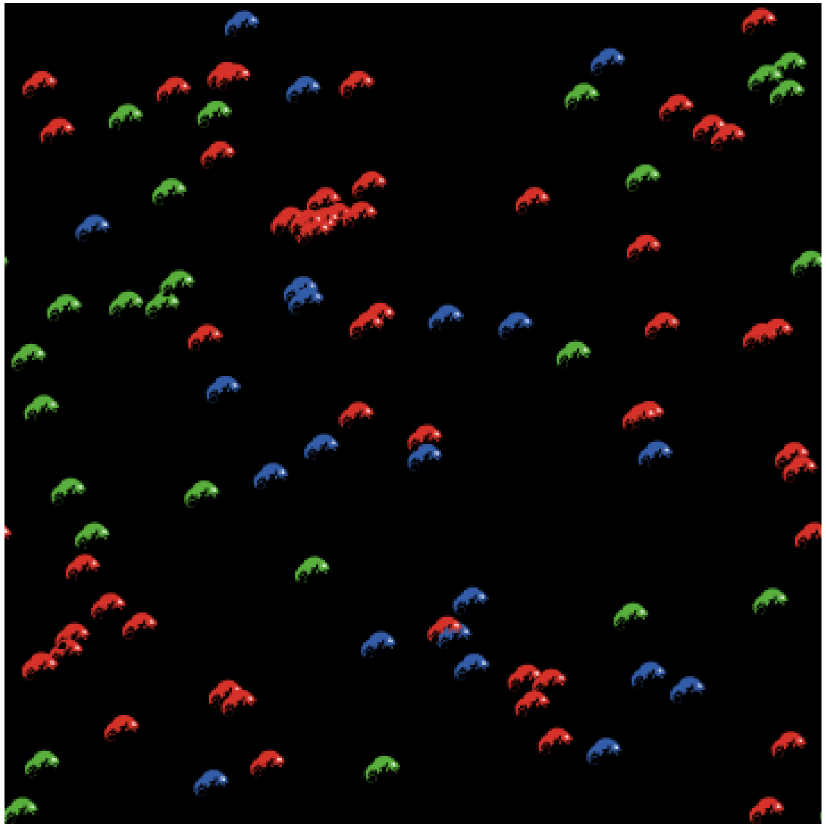
mixed preferences

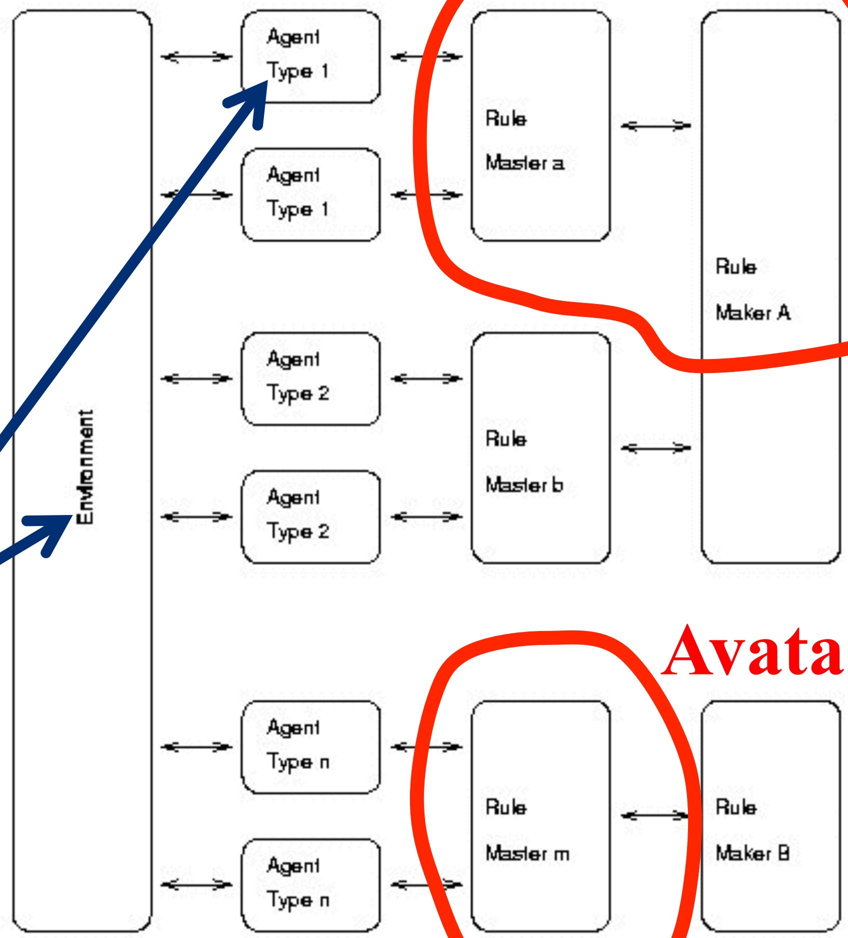


Learning chameleons (http://goo.gl/W9nd8)

In a work of mine you can find, finally, agents requiring a lot of computational capability to learn and behave. They are chameleons changing color when getting in touch with other ones; they can learn strategies, via trials and errors procedures, to avoid that event.







Fixed rules

ANN

(CS)

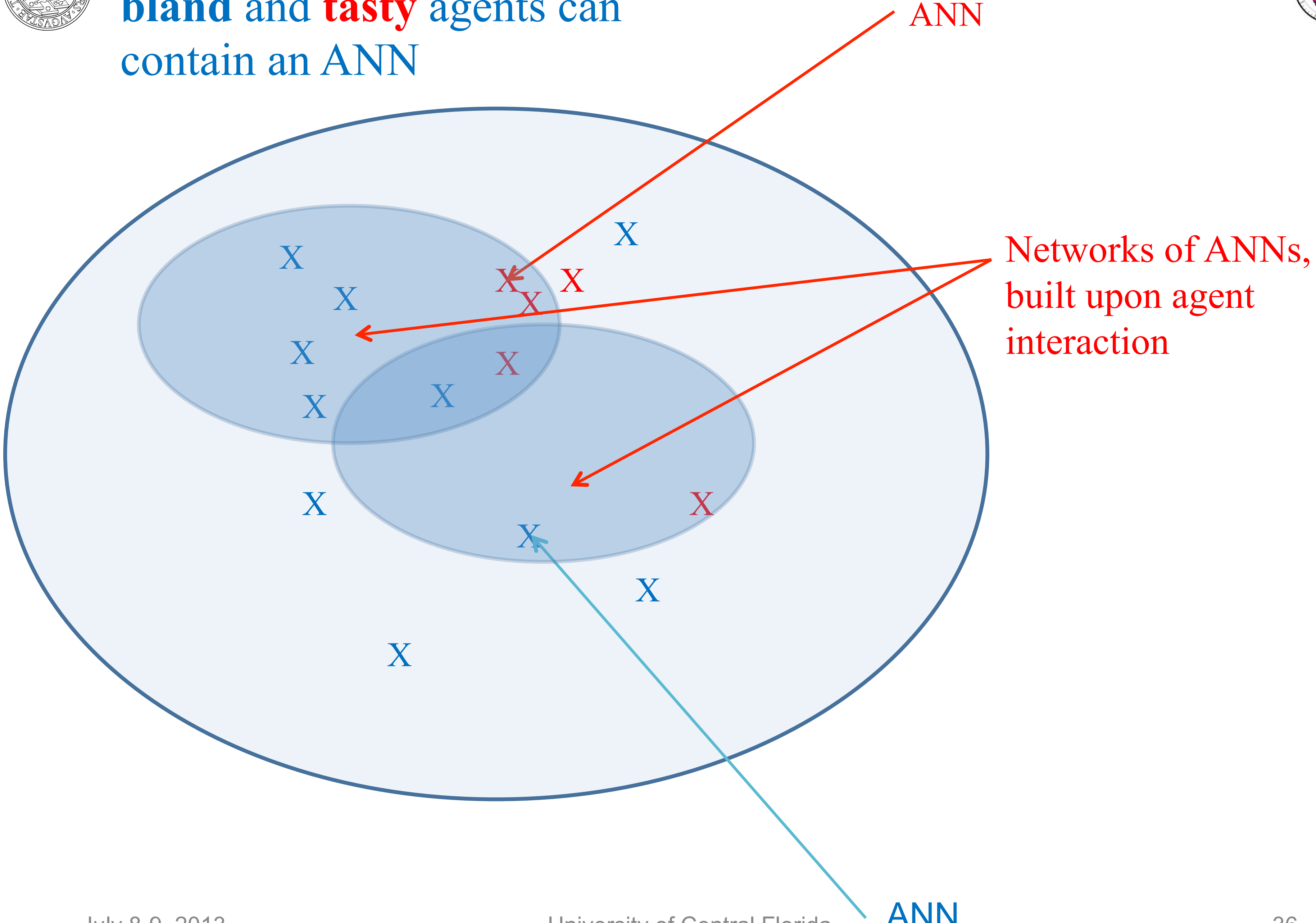
(GA)

Avatar

Reinforcement learning

Microstructures, mainly related to time and parallelism

bland and **tasty** agents can contain an ANN



$$y = g(x,z) = f(B\, f(A\, (x',z')'))$$

(1) (n+m)

effect information action/s

or, if $z = \{z_1, z_2, \ldots, z_m\}$

$$y = g_m(x,z) = f(B\, f(A\, (x))$$

(m) (n)

an effect for each possible action

I - Learning from actual data, coming from experiments or observations

Rule master

Xα,Zα	Yα
Xα,1;Zα,1	Yα,1
…	…
Xα,m;Zα,1	
Yα,m	

Xβ,Zβ	Yβ
Xβ,1;Zβ,1	Yβ,1
…	…
Xβ,m;Zβ,1	
Yβ,m	

Different agents (α and β), with NN built on different set of data (i.e. data from real world or from trial and errors experiments), so with different matrixes A and B of parameters

IIa- Artificial learning, via a trial and errors process, while acting

$$y = g([x,z]) = f(B\ f(A\ (x',z')'))$$

(p)

(n+m)

actions

Rule master

effects

information

Different agent, generating and using different sets, A and B, of parameters (or using the same set of parameters, as collective learning)

Coming from the simulation

the agents will choose y maximizing:

(i) individual U, with norms
(ii) societal wellbeing

IIb- Artificial learning, via a trial and errors process, while acting

$$y = g([x,z]) = f(B\ f(A\ (x',z')'))$$

(p)

(n+m)

actions

Rule master

effects

information

Different agent, generating and using different sets, A and B, of parameters (or using the same set of parameters, as collective learning)

Coming from the simulation

Emergence of new *norms* **[modifying U=f(z) , as new norms do]** and *laws* **[modifying the set y, as new laws do]**

the agents will choose y maximizing:

(i) individual U, with norms

(ii) societal wellbeing

accounting for social norms

Learning of **A** and **B** matrixes of parameters, to determine **y**, scalar of vector, in the two previous slides (learning while acting), can be:

- continuous, with the actual values, while each agent is acting;

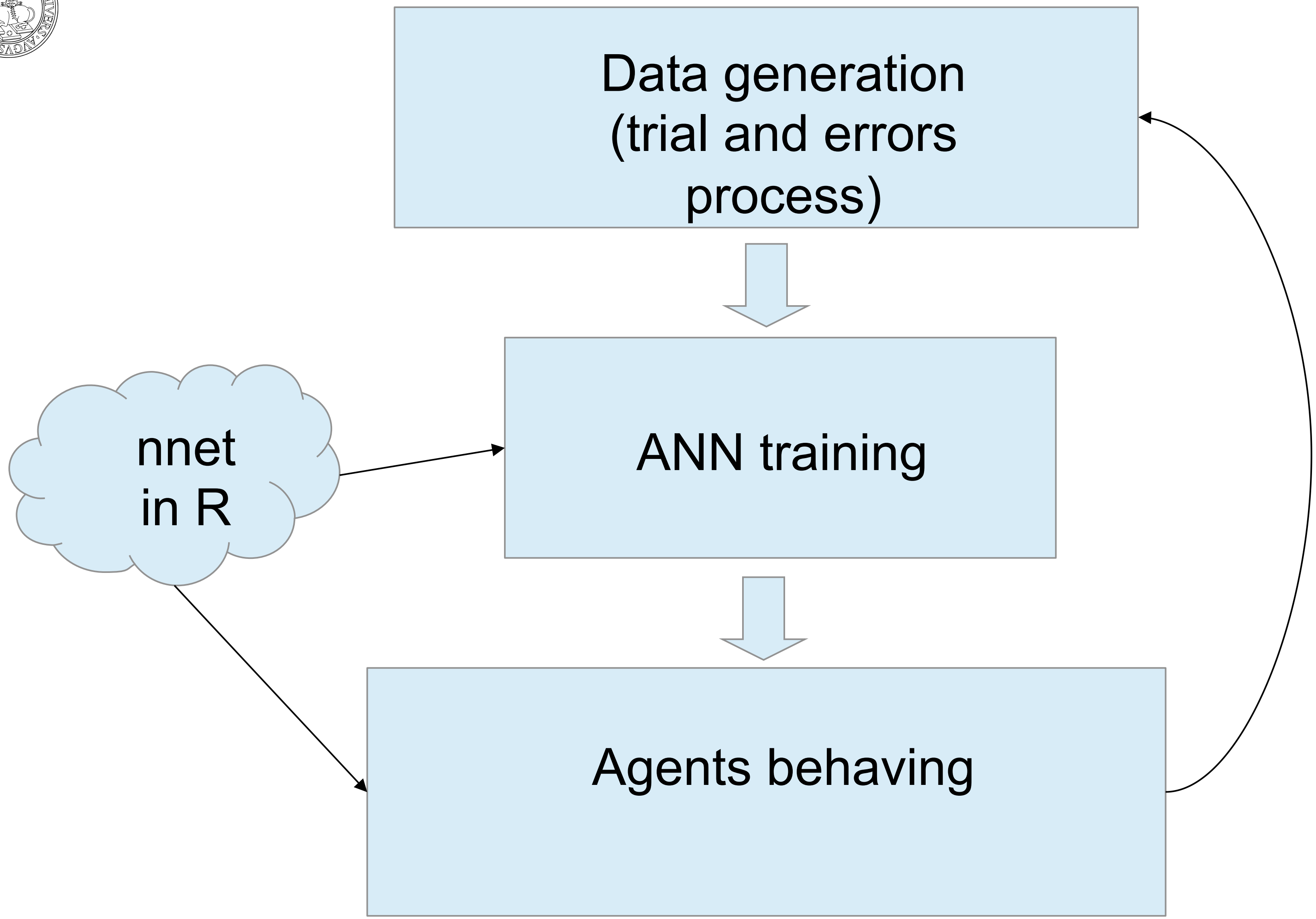
- in batch, with the actual values after the action of all the agents (as presently done).

A learning agent environment related to SLAPP:

nnet&reinforcementLearning

look at the file **z_learningAgents_v.?.?.zip** at **goo.gl/SBmyv**

You need Rserve running; instruction at **http://goo.gl/BawNN**


Data generation (trial and errors process)
nnet in R
ANN training
Agents behaving

nnet {nnet}

R Documentation

Fit Neural Networks

Description

Fit single-hidden-layer neural network, possibly with skip-layer connections.

Usage

`nnet(x, …)`

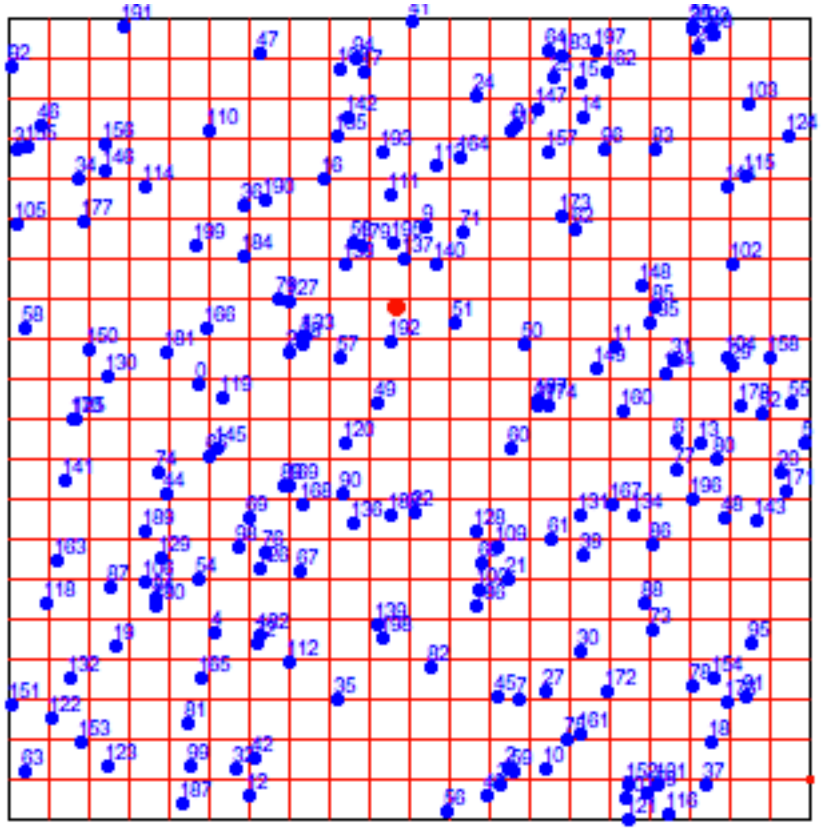
Optimization is done via the BFGS method of optim.

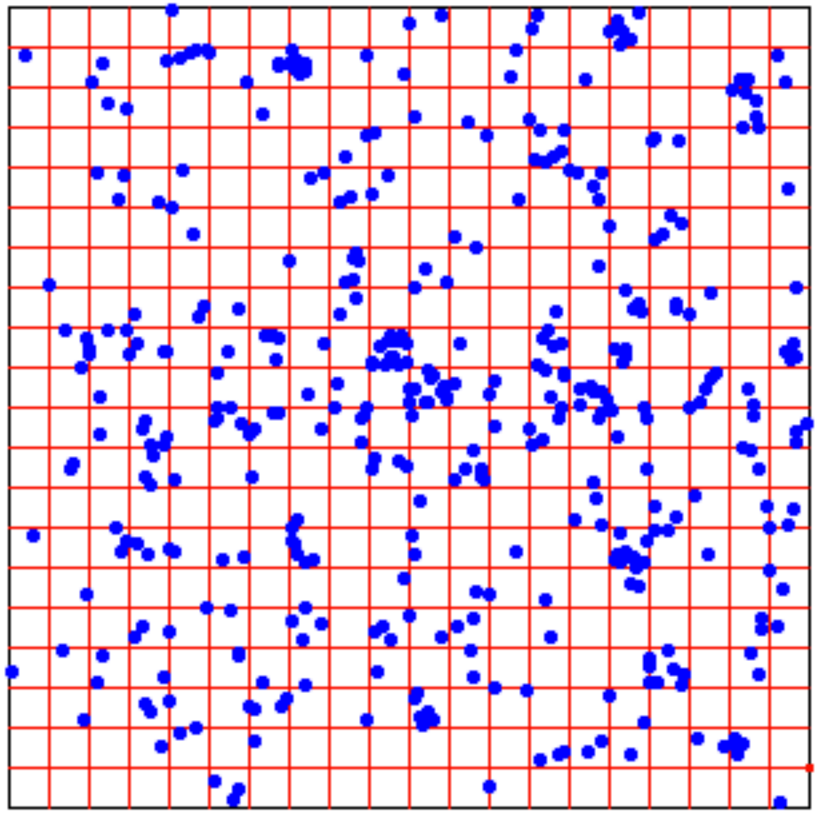
Method `"BFGS"` is a quasi-Newton method (also known as a variable metric algorithm), specifically that published simultaneously in 1970 by Broyden, Fletcher, Goldfarb and Shanno. This uses function values and gradients to build up a picture of the surface to be optimized.

References

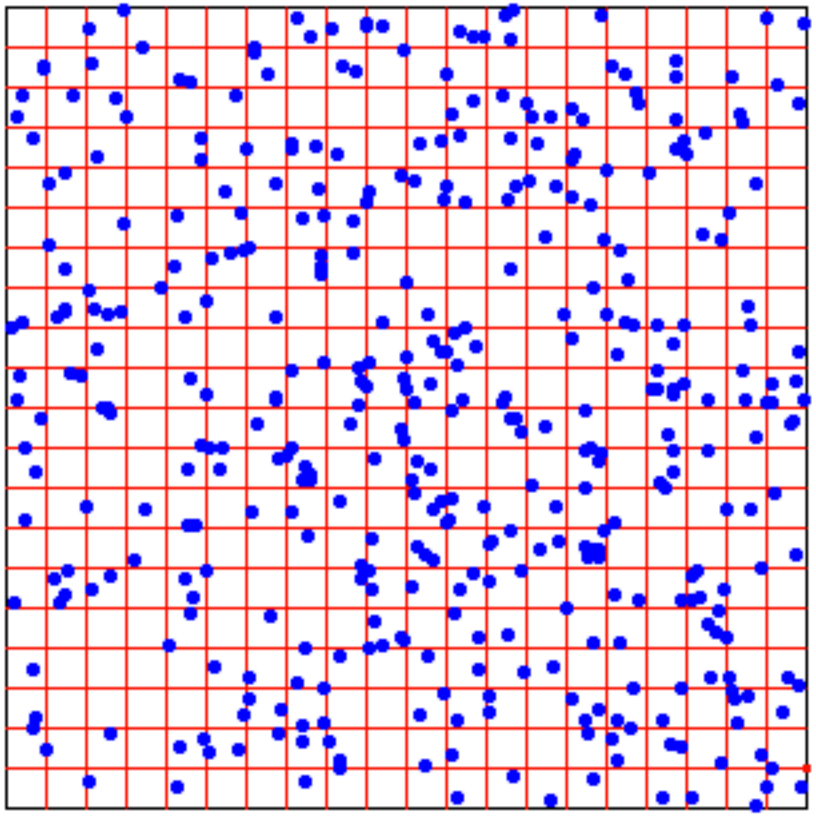
Ripley, B. D. (1996) *Pattern Recognition and Neural Networks.* Cambridge.

Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S.* Fourth edition. Springer.

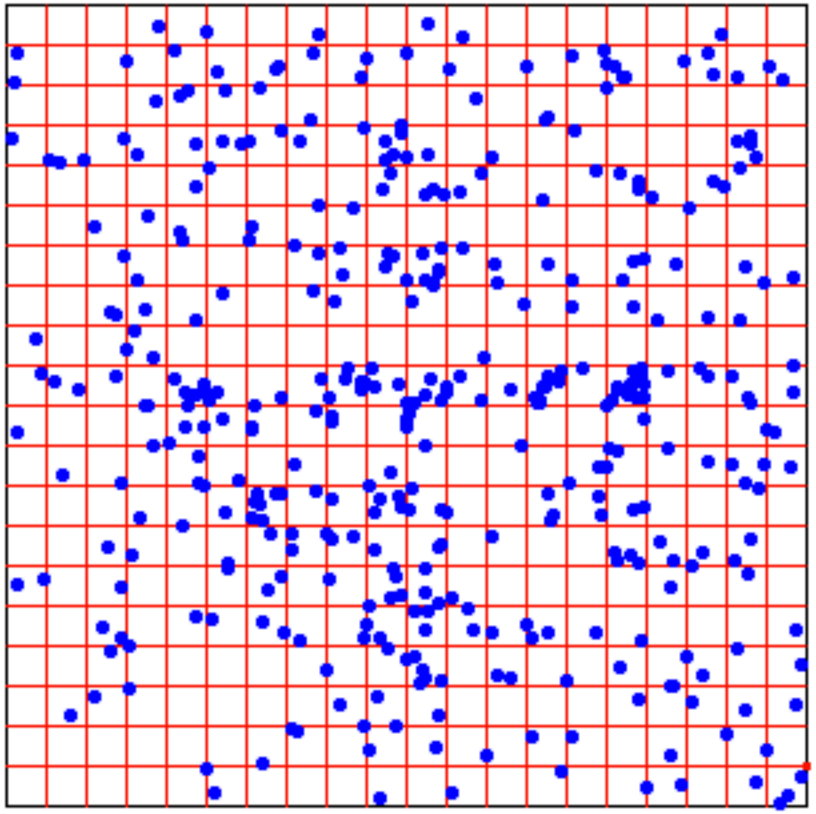




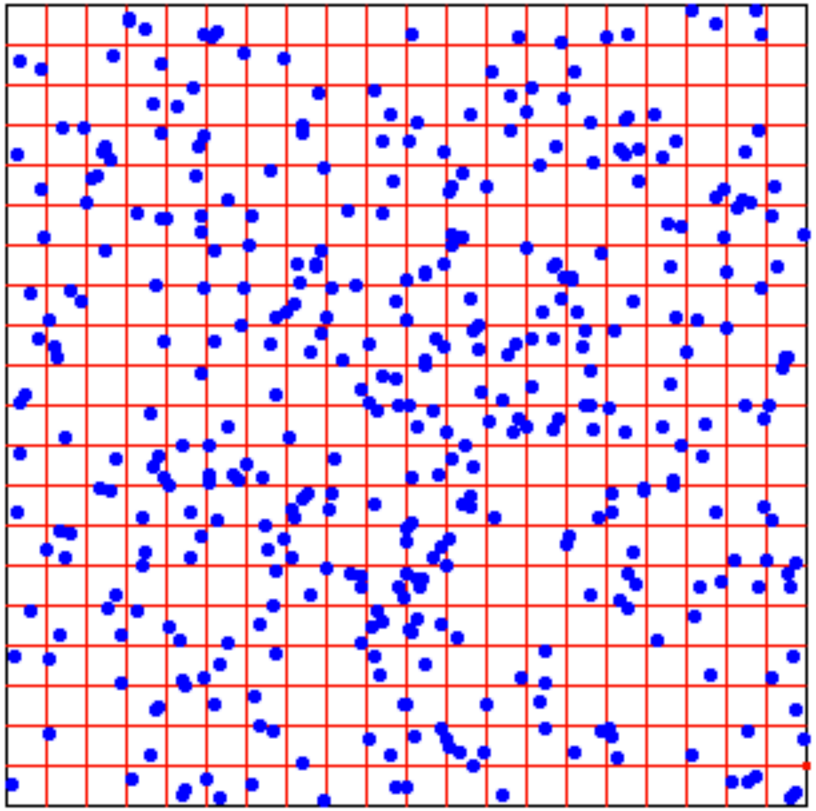
400 agents, going closer to other people, hard parallelism



400 agents, searching for empty spaces, hard parallelism



400 agents, going closer to other people, soft parallelism



400 agents, searching for empty spaces, soft parallelism

Why the agents act in that way?

The crucial question now is: *why* they do that?

Apparently, it is an irrelevant question: they do that because we asked them to learn how to behave to accomplish that kind of action, but we are considering a tiny problem.

In a highly complex one, with different types of agents, acting in very distinctive ways, to have the capability of tracing, in our simulator, in a detailed way, the kind of behavior that the agents are following and the explanation of their choice is extremely important.

We have to add, in our model, a layer dealing with the so-called Beliefs Desires Intentions (BDI) agent definition.

In SLAPP that layer presently does not exist.

We can quite easily refer to an extension of NetLogo, adding BDI capabilities, with a few simplifications, as a project of the University of Macedonia, in Greece, at http://users.uom.gr/~iliass/projects/NetLogo/.

Have a look to the their Taxi_scenario_Cooperative_Streets.nlogo example

… and networks?

We have to use networks - as systems that have individuals as nodes (agents), and social and economic relationships as edges - within the agent-based framework.

In networks, we have to take into considerations:

- the time dimension;
- the probabilistic behavior of the different links and actions.

A never ending research project

- agents
- agents + learning
- agents + BDI
- agents + learning + BDI
- agents + networks
- agents + networks + learning + BDI

Using ABMs in policy making …

Decisions and actions in policy and law: could agent-based simulation help?

The key idea is now that of defining policy in a participative way, with citizens, and from there to evolve laws and improve social norms in a better understood and shared approach.

Is that a dream or a research field?

At http://www.ecb.int/ecb/educational/economia/html/index.en.html, with €CONOMIA - The Monetary Policy Game - we can play to be the president of the European Central Bank.

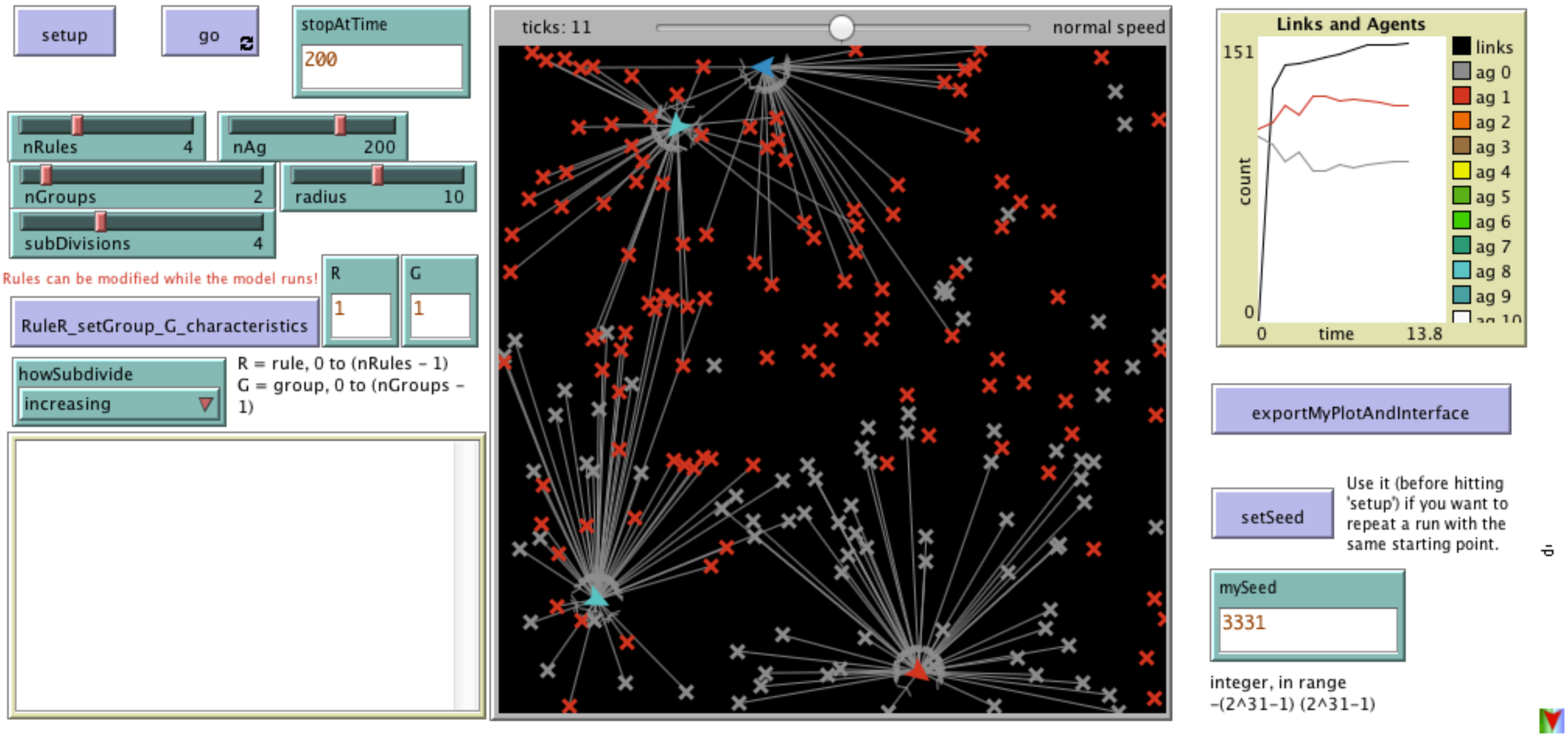
.. or *eGovernment meets the eSociety*
A further step, close to be tangible, at http://wegov-project.eu

Or *Laws in a bottom-up process*
Finally, at http://gigaom.com/europe/online-crowdsourcing-can-now-help-build-new-laws-in-finland

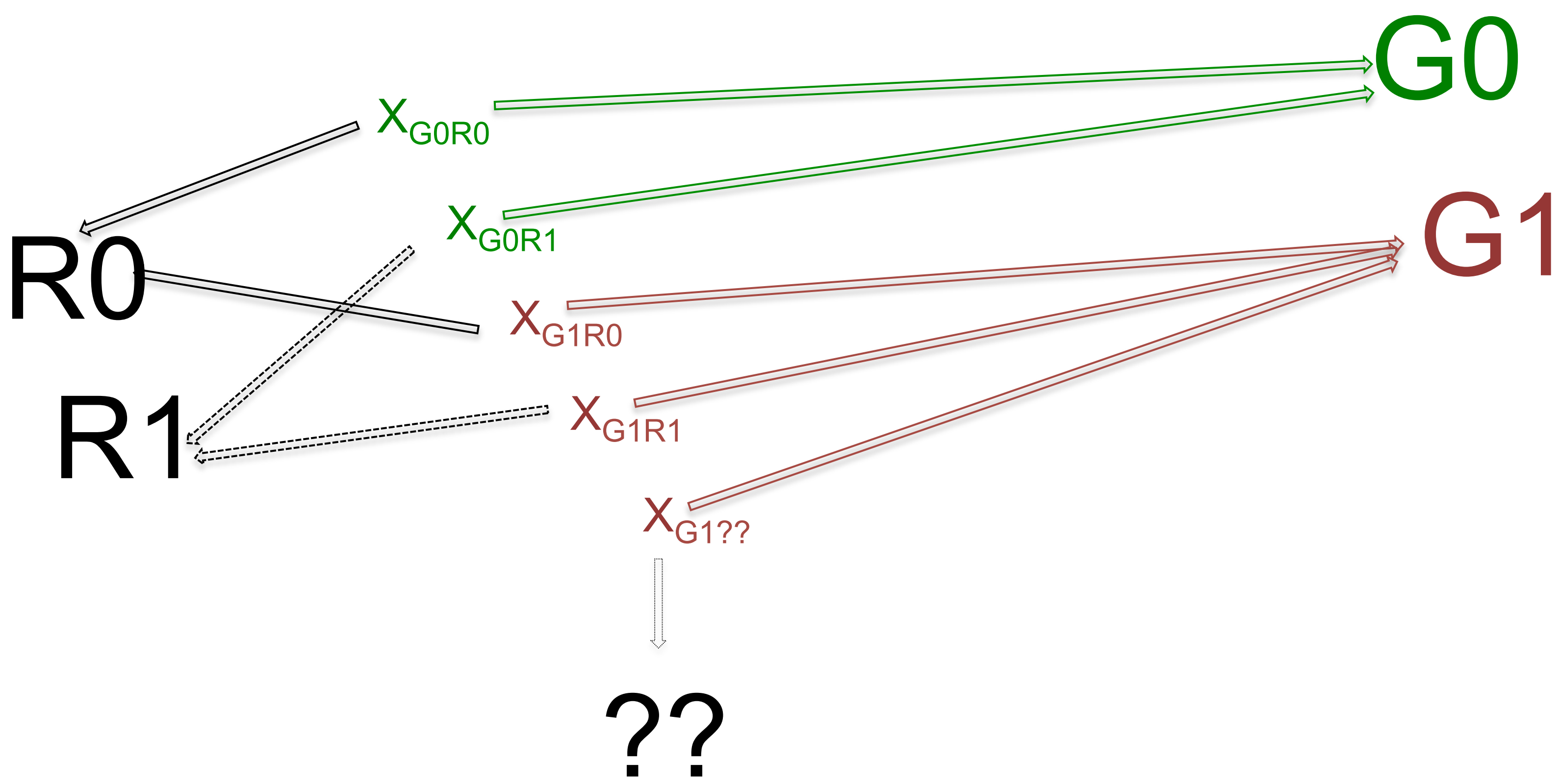
… and to understand how people interact in forming opinions …

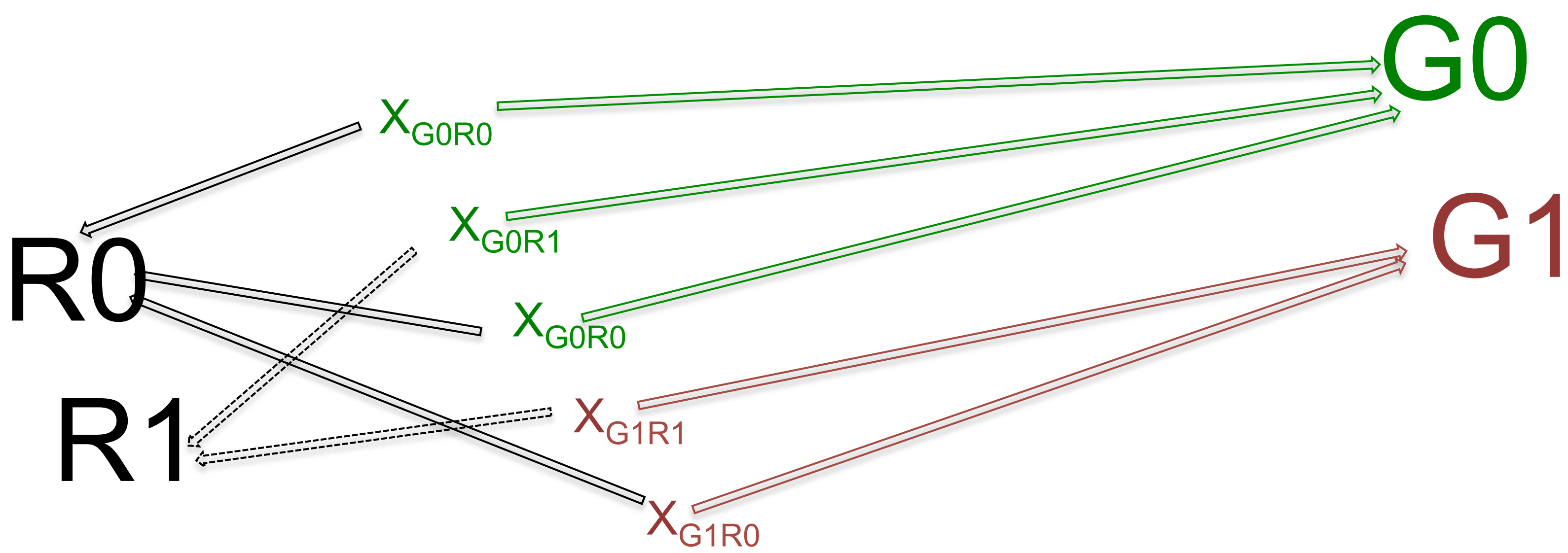
Rules and scenarios, at

http://eco83.econ.unito.it/terna/ruleScenarios/ruleScenariosEn.html



The idea …




G0
G1
X G0R0
X G0R1
R0
X G0R0
R1
X G1R1
X G1R0

Moving from the fascinating complex system of actions and counteractions designed by the collective behavior of a modern society, we propose a quite simple simulation model, to reason about people changing their mind about opinions, decisions, reference groups. The model has two underpinning structures:

(i) groups of people and

(i) rule systems, to which people belong.

We propose a *machine* useful to experiment with the effects of different situations, described by the system of rules.

The *machine* is running on-line at http://goo.gl/lFjJm, without installing anything locally, or you can download it from the same address.

The ruleScenarios.nlogo model is built with NetLogo (http://ccl.northwestern.edu/netlogo/) and can generate very different situations.

The current application is about two groups: one based upon a bad way of behaving (an example: using public resources for personal and private advantages) and the other one following the correct way of behaving.

We have from two to four systems of rules, with people adopting each specific system with different dynamics of the opinions about the two groups, reinforcing the first one or the second one or being neutral.

The scenarios emerging from simple systems of rules are surprisingly realistic and can explain several current political situations around the world.

How it works (an example)

We have nAg agents, between 0 and 300, completely idealized (individuals, companies, associations, institutions, ...) that interact by changing their membership to a group (nGroups, between 1 and 10) depending on the situation of the other agents in a neighborhood (radius, between 0 and 20, for comparison, the world is 33x33).

The agents bind themselves randomly to a system of rules, which manages the changes of opinions (the fact of belonging to a group).

The rule systems are approximations of the cultural tools available to each agent. Future versions of the model will be able to handle modifications into the systems of rules and the movements of agents among them.

We have nRules (between 1 and 10) rules; the links occur from an agent to one (and only one) rule, randomly, choosing among those that are inside a given radius from each agent.

Agents and rules move randomly in the space; once connected, you will see a link (link in the jargon of NetLogo and of network analysis). In the graph at the top right, the upper line indicates the number of links, which will reach the max when the # of links is equal to the # of agents nAg.

The rules contain transition matrices, such as the following (the content is given by random flat numbers between 0 and 1). The number of subdivisions specifies the granularity of the analysis.

Rule 0

	subDivisions		
	0	1	2
groups			
0	0.2	0.3	0.1
1	0.1	0.9	0.2
2	0.6	0.5	0.1

Rule 0

	subDivisions		
	0	1	2
groups			
0	0.2	0.3	0.1
1	0.1	0.9	0.2
2	0.6	0.5	0.1

The random data (always between 0 and 1, with more decimal, not just one as in the example) can be elaborated and normalized by adding them in a progressive way by row, from left to right (increasing order) or from right to left (decreasing order), dividing by the total.

Rule 0

	subDivisions 0	1	2
groups			
0	0.2	0.3	0.1
1	0.1	0.9	0.2
2	0.6	0.5	0.1

For example, the first line above becomes:

```
(0.2)/(0.2+0.3+0.1)   (0.2+0.3)/(0.2+0.3+0.1)   1 [increasing]
```

or

```
1   (0.1+0.3)/(0.2+0.3+0.1) (0.1)/(0.2+0.3+0.1)   [decreasing]
```

The first row (used to evaluate the group 0) would then be:

```
0.2    0.3    0.1      if random
0.333  0.833  1        if increasing
1      0.666  0.166    if decreasing
```

in the last case, transformed as follows:

```
-1   -0.666 -0.166
```

```
Rule 0 / Group 0
                          subDivisions
                          0    1    2
   groups
   0                      0.2  0.3   0.1       if random
   0                      0.333 0.833  1       if increasing
   0                     -1    -0.666 -0.166   if decreasing
```

Suppose that an agent, in its surroundings, has a composition for shares of the groups (referred to as 0, 1, 2), including itself, equal to 0.31, 0.25, 0.44 (total 1 or 100%)

Moving from left to right, we compare 0.31 with the values of the row 0 of the matrix and take note of the cases in which is greater (using -0.31, if decreasing) of the value of the column, keeping the last.

Rule 0 / Group 0

groups	subDivisions 0	1	2	
0	0.2	0.3	0.1	if random
0	0.333	0.833	1	if increasing
0	-1	-0.666	-0.166	if decreasing

In our case (0.31):

- considering the random construction, the last case and would be that of the third column (or column 2, calculating from 0), so with a score of 2; if not greater than any of the column values, the score would be -1;
- considering the increasing construction, we stop before the first column, with a score of -1; in the increasing case, we compare a value that will never exceed that of the last column, which contains 1, then, by symmetry, the values are incremented by 1; here we would get score: -1 +1 = 0;
- considering the decreasing construction, -0.31 is greater than -0.666 (right value to which it is greater), with a score of 1.

```
Rule 0 / Group 0
                              subDivisions
                              0    1    2
   groups
   0                          0.2  0.3   0.1      if random
   0                          0.333 0.833  1      if increasing
   0                         -1    -0.666 -0.166  if decreasing
```

The same operation is repeated for the other groups, with shares 0.23 and 0.44. The random construction gives random results; the increasing one rewards large groups and punishes the minorities, the decreasing ones rewards the minorities and punishes the large groups.

If the panel of the program does not contain increasing or decreasing condition, the rule contains all random rows.

Suppose that the assessment of the groups, operated by an agent on the basis of the rule to which is linked, gives scores (with three groups): 0 -1 3

The agent would pass to the third group (referred to as group # 2). If two or more scores are the same (eg 1 1 -1), we choose randomly between the groups with the same score.

If an agent is not (yet) linked to a rule, it does not select any group and remains in the initial group.

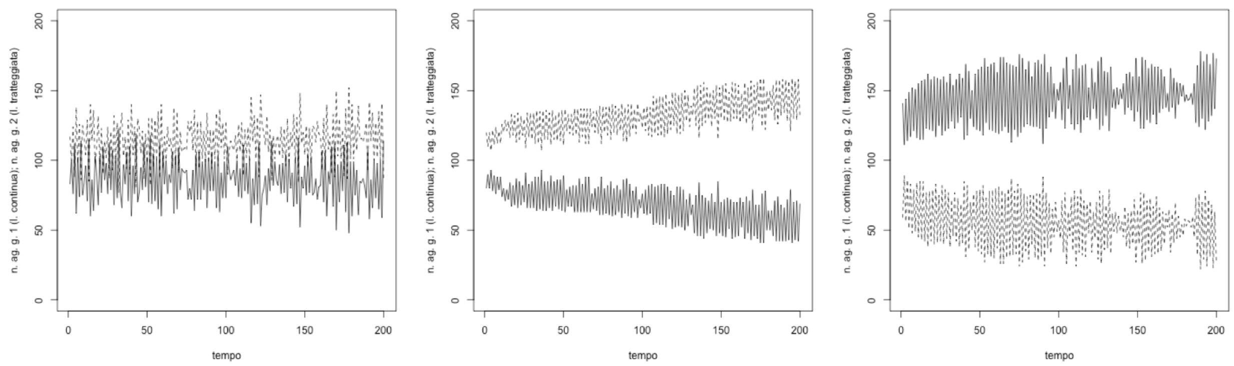
Examples

R G

G 1 as socially negative (dark in graphics)

0 0 *decreasing*
0 1 *increasing* or rule *di*

1 0 *increasing*
1 1 *decreasing* or rule *id*



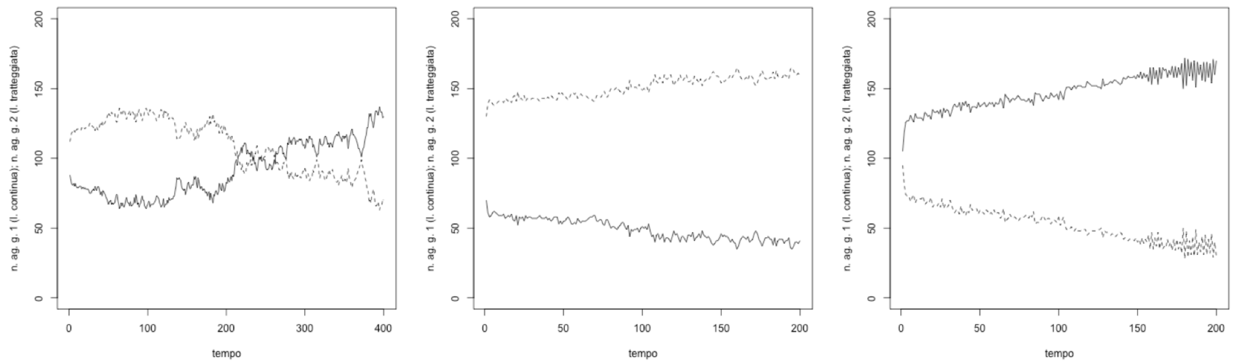
R G

G 1 as socially negative (dark in graphics)

0 0 *decreasing*
0 1 *increasing* or rule *di*

1 0 *increasing*
1 1 *decreasing* or rule *id*

2 0 *increasing*
2 1 *increasing* or rule *ii*



R G

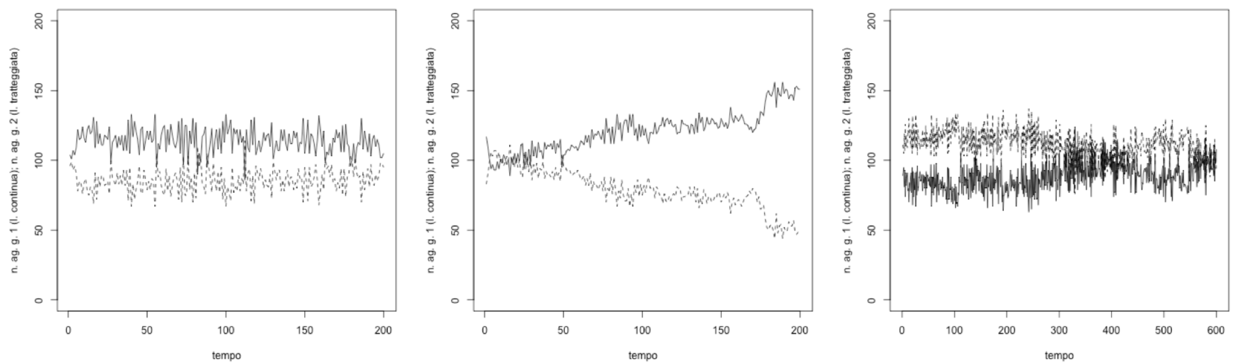
0 0 *decreasing*
0 1 *increasing* or rule *di*

1 0 *increasing*
1 1 *decreasing* or rule *id*

2 0 *increasing*
2 1 *increasing* or rule *ii*

3 0 *increasing*
3 1 *decreasing* or rule *id*

G 1 as socially negative (dark in graphics)



Thanks

Pietro Terna, pietro.terna@unito.it

the presentation is at goo.gl/UMs4C