Simulation, Analysis, and Validation of Computational Models

— Multi-Agent Systems —

Lecturer: Michael Herrmann
School of Informatics, University of Edinburgh
michael.herrmann@ed.ac.uk, +44 131 6 517177

Overview

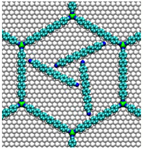
- Schelling model
- Self-organisation and emergence
- Multi-agent systems (a.k.a. self-organised systems)
- Mesa

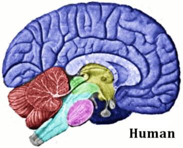
Multi-agent systems (MAS): Examples

- Stock exchange
- Team games
- Work groups
- Animal behaviour
- Swarm intelligence
- Microbial intelligence
- Artificial chemistry
- Self-driving cars
- Warehousing
- Logistics
- Social modelling
- Epidemology
- Sensor networks
- MARL


D. Dibenski: images.fws.gov


Serge Kernbach


C. A. Palma e.a. TUM



Looie496

MAS

Much of the MAS literature has more restrictive (or just different) assumptions, but in order for simulations to be interesting, agent should have the following properties:

- **Autonomy**: Agents have some level of independence.
- **Local sensing**: Agents sense only some aspect of the environment the complexity of which surpasses the complexity the agent.
- **Decentralisation**: The behaviour of an agent can influence "nearby" agents, but no agent is in control of the system.

Panait & Luke (2005) Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems* 11, 387–434.

Simulations can be useful for many other reasons: E.g. to find out about stability, robustness, efficiency, scaling of any complex systems, or to make sure that self-organisation is not an issue.

Unmixing and structuring systems

- Schelling model

T.C. Schelling (1971) Dynamic models of segregation. J. Math. Sociology 1, 143–186.

- Clustering by robots aimlessly pushing boxes
- Granular convection (“Brazil nuts effect”)

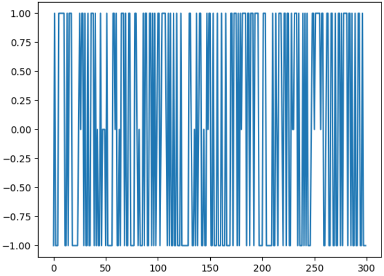
Rosato e.a. (1987) Why the Brazil Nuts are on Top. *Physical Review Letters* 58, 1038–41.

- Large scale structure of the universe

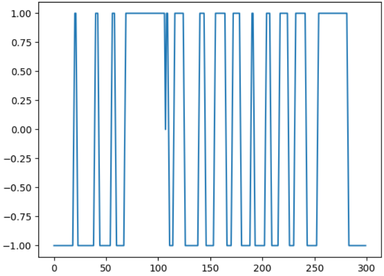
Schelling model

Original formulation: Consider city with people of group A and people of group B

If the fraction of neighbours of the same group is less than p, move to a more homogeneous place, until no more moves are possible.



1D case: Random initialisation



Segregated by Schelling model

Notebook from Li&Nakano: Consider different groups sizes, various p and fraction of free sites, and try to transfer to higher dimensions.

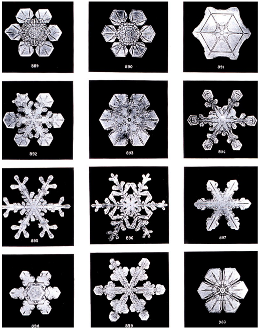
MAS

- Due to the activity and interaction of many agents, the order in the system can increase
- As the agents use energy from outside this is not in conflict with the *2nd law of thermodynamics*
- Energy import enables entropy export
- Often phenomena such as *self-organisation* or *emergence* occur, i.e. "qualitative changes by more of the same".

Emergence

A property of a systems is called emergent, if it is not present as a property of its elements.

- E.g. *kinetic energy* of a multi-particle systems "averages out", and *temperature* because a meaningful description
- A single neuron (both biological or artificial) can perform a (linear) binary classification, while a neural network can recognise complex patterns.
- An ant can find food, but many ants are needed to start a colony.



Self-organisation

- The mechanism of order arising from local interactions between parts of an initially unstructured system

  (One way to tell apart: Emergence is a phenomenon, self-organisation is a mechanism)

- Examples:
  - Waves forming driving by wind on a surface of water
  - Specific light frequencies are selected from wide spectrum to produce laser light
  - Convection patterns: Bénard effect, weather, cloud formation
  - Pattern formation, reaction-diffusion system
  - Traffic jams, stock-market crashes, rain, natural evolution, biological neural systems, cosmology etc.
- Simulations can serve various purposes, but most interesting are cases of self-organising processes, i.e. when system properties cannot be easily predicted from the elements.

'Since no system can correctly be said to be self-organizing, and since use of the phrase "self-organizing" tends to perpetuate a fundamentally confused and inconsistent way of looking at the subject, the phrase is probably better allowed to die out.' Ashby, 1962

Agents

- Generally (real or simulated agents)
  - Data: Sensing environment (e.g. neighbouring agents)
  - Model (Policy): mapping data (incl. internal states) $\rightarrow$ actions
  - Some form of actuator (critical for the agent being an agent)
  - Initialisation (e.g. for heterogeneous MAS)
  - Termination (or observation) of experiment
- In simulation:
  - Environment
  - Position in environment
  - Agent-ID (also for centrally controlled real agents)

Agents on a graph

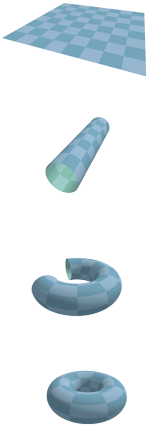
Agents can be

- moving in space e.g. articulated robot swarms
- stationary, e.g. agents on a grid interacting with their neighbors (see rock-paper-scissors example below)
- moving towards a position that is (at least temporarily) stationary, e.g. kilobots, modular robotics, coral polyps

Periodic boundary conditions

How to minimize effects of boundary corners, finite size of the simulation?

- Plain playing field becomes topologically a torus, while geometrically still being flat.
- No boundaries: Every point (or agent) is equally in the interior of the system.
- If spatial patterns play any other role in the dynamics, then Moiré-pattern-like interaction by that spreading either way can affect the dynamics.
- If spatial scales are small compared to the emerging patterns, usually no problem occurs, but for a large playing area any boundary conditions are likely to be fine.
- Magic numbers: For emerging patterns of size 20 for total size 50: destructive interference. For total size 60: constructive interference



Lucas Vieira (2012)

Agent-based in Mesa

- Mesa is an open-source Python library for agent-based modeling for simulating complex systems and exploring emergent behaviors.
- Time-order: Both in discrete and (numerical) continuous systems all agents (rather: dimensions) are updated either
  - synchronously: Calculate all, then update all
  - sequentially: Calculate and update one by one
- How to implement parallel activity to run[1] asynchronously ensuring that each agent has a fair share of *compute*?

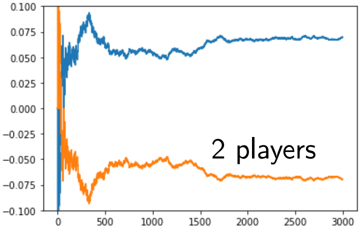
[1] This is about simulation of causality rather than hardware or real-time.

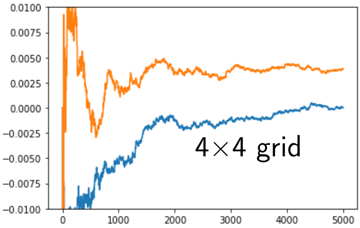
Agents playing Rock, Paper, Scissors (see Li & Nakano, Ch. 7)

Agent A ("stubborn"): Stays with probability 80% with previous choice, and switches with with probability 10% to one of the other two.

Agent B ("sneaky"): Use move that would have worked that previous time, or random after a draw

Points: "1" Not beaten by any neighbours , "-1" Not won against any of them, 0 otherwise (Teams A and B are arranged in checkerboard fashion with for neighbours from the opposing team).





In a 1-to-1 setting, sneaky can beat stubborn easily. On a square grid sneaky has to beat 4 neighbours, which is difficult when it is likely that at least one of them is changing its move randomly.

Conclusion

- Group behaviour is deviate strikingly from individual or pairwise behaviour
- Group behaviour be against the intention of each of the member even in a homogeneous group
- Simulations can show interesting system-level phenomena

Now: Q&A

- Coursework assignment
- Labs

Next week

- Case studies