8. Evolution, Co-evolution (and Artificial Life)

Modelling Social Interaction in Information Systems

http://davidhales.com/msiis

David Hales, University of Szeged

dave@davidhales.com

What is evolution?

- It is a very old idea that predates modern science
- It is a theory of change
- Originally applied to human societies and ideas – because people could see these changed over time
- It isn't until recently (fossils etc) that it was realised biological life forms changed over time

Biological evolution

- Evolution in everyday language has come to mean biological evolution
- Darwin did his famous empirical work observing biological organisms
- Biological evolution draws on empirical facts and theoretical models (often mathematical)
- Here we will focus on “abstract evolution” simulated in computer programs

Abstract evolution

- Evolution can be viewed as an algorithmic abstraction that can be used to understand / implement a process of change given:
  - Things that replicate / get copied (**units** of selection)
  - **Variation** in replicators (mutation)
  - Differential **selection** of replicators
- "fitness" means how good a replicator is at replicating (how many copies are made)
- In this context "survival of the fittest" is a tautology

Book: Daniel Dennett (1995) Darwins Dangerous Idea. Simon & Schuster

Abstract evolution (GA's)

- Genetic algorithms (which I think you know)
- Are an optimisation technique
- Define a space of solutions to a problem
- Code different candidate solutions in an "artificial chromosome" (often a bitstring but not always)
- Use an evolutionary algorithm to adapt solutions towards better (hopefully optimal or good enough) solutions
- Do this through some form of selection, recombination (crossover), mutation and reproduction
- John Holland early 70's, Alan Turing 50's, other earlier thinkers...

Book: Holland, John (1975). Adaptation in Natural and Artificial Systems. Cambridge, MA: MIT Press

Genetic Algorithms

- Initialise a population of (N) random chrom.
- Loop for some (G) number of generations
  - Loop for each chrom.
    - Test chrom. against an objective function f() – award a fitness score
  - End loop solutions
  - Reproduce chrom probabilistically proportionally into the next generation based on fitness score
  - Apply some genetic operator (such as crossover)
  - Mutate reproduced chrom. with small prob. (m)
- End loop generations

Reproduction / Selection

- Many ways to simulate reproduction:
  - Roulette Wheel Selection
  - Tournament Selection
  - Other kinds…
- In general you want an easy to implement and fast method
- That will allow for fitter solutions to tend to increase in the population over time

Roulette Wheel Selection

- Suppose you have a population of chromosomes and each has been allocated a fitness based on f()
- Add up the fitness's of all chromosomes = tf
- Repeat until next generation is full:
  - Generate a random number R in that range 0..tf
  - Select the first chromosome in the population that when all previous fitness's are added - gives you at least the value R
  - Reproduce the selected chrom. Into the next generation
- Hence it is like a roulette wheel where each spot on the wheel (representing a chromosome) is the size of the fitness of the associated chromosome

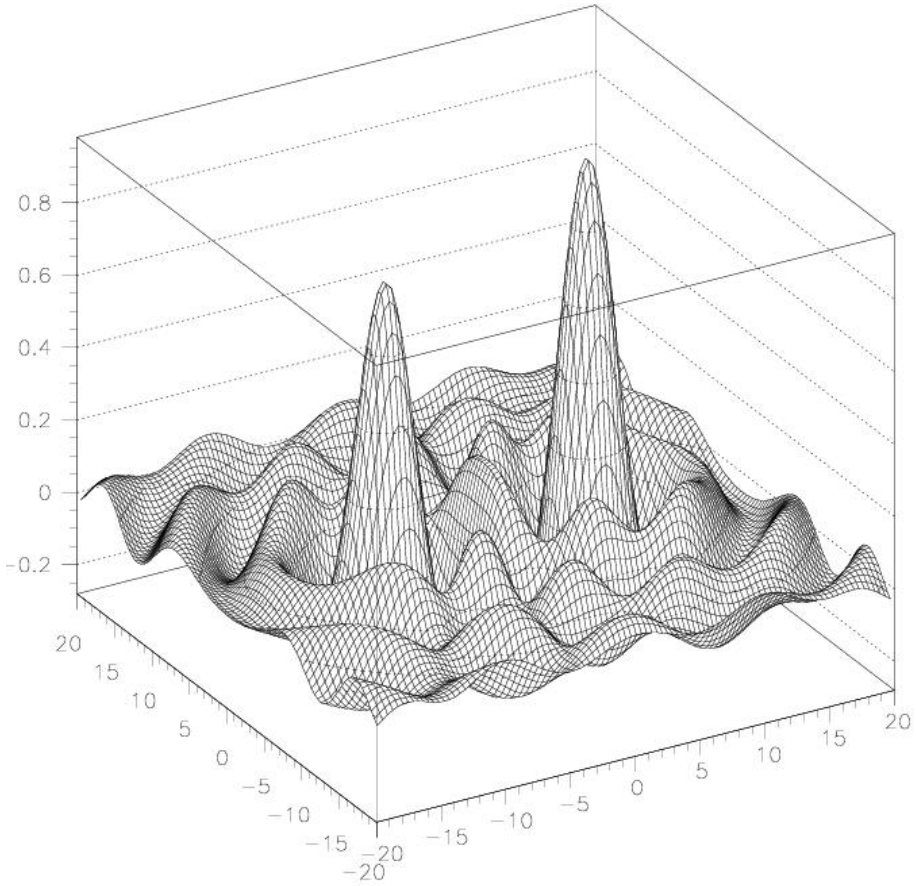
Tournament selection

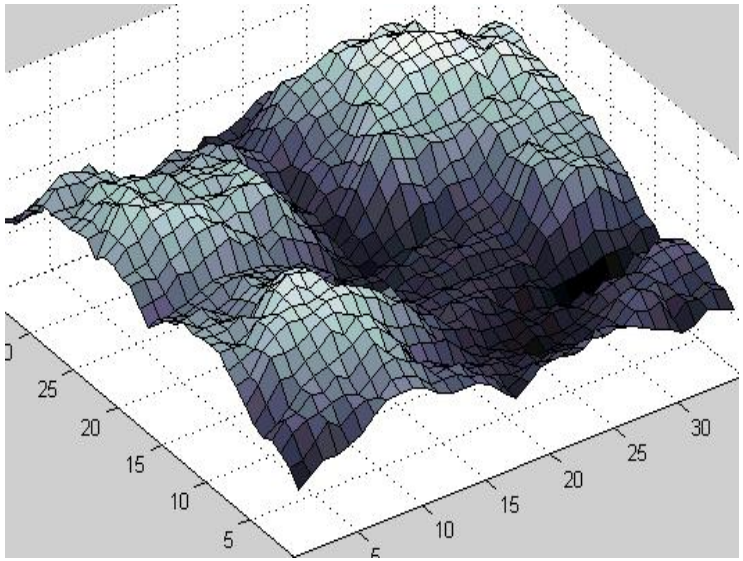
- Many variants but a very simple form is:
- Repeat until next generation is full
  - Select pairs of chromosomes randomly
  - Reproduce the one with the highest fitness
  - Or a random one if they have same fitness

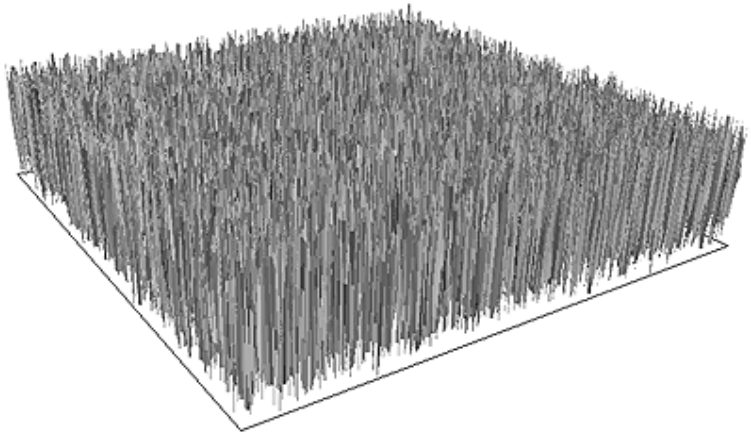
Genetic Algorithms (crossover)

- You have to choose, G, N, m and f()
- Often GA's use **crossover** (recombination) of reproduced chromosomes as well as mutation
- This involves splicing together parts of chromosomes
- Can be compared to sexual reproduction
- 1-point crossover: take two chrom., select a random cut-point and spice together the chrom. of the two parents
- Holland developed "Schema Theory" to understand how various genetic operators (such as crossover) work
- Using GA's for optimisation very much an "art"
- There is no "free lunch" for search problems!

Fitness landscapes







Complex fitness function?

- Can we still use evolutionary algorithms without simple explicit fitness functions?
- Yes, simple way, let a person look at solutions and select some they like better
- Dawkins “bimorphs” (NetLogo model library: biology/evolution/sunflower biomorphs)
- OR somehow let the “world” supply the fitness function – or a simulation of the world
- Evolving robots with “real physics”

Endogenous fitness functions

- Suppose our solutions are “agents” that must interact socially with each other in a simulated environment to gain fitness
- The fitness of an agent depends on how the other agents behave
- Remember Axelrod’s tournaments?
- To get a score (or fitness) for each algorithm he had to play them off in simulated tournaments
- Since the fitness of any agent is dependent on the other agents in the population
- This is called **co-evolution** because each agent evolves relative to the others rather than optimising an exogenous fixed fitness f()
- In this sense f() takes as inputs all the other agents
- When the agents are strategies in a simple game with known payoffs this relates to **evolutionary game theory**

An evolutionary PD game

- Suppose:
  - agents as 1 bit strategy in the the PD game where 1 = coop and 0 = defect
  - population of (N = 100) such strategies initialised at random (0 or 1)
  - Apply an evolutionary algorithm where each generation each agent is randomly paired with some other agent in the population and plays a game of PD
  - Reproduction (roulette wheel) using the accumulated payoff from the games as the fitness of each agent
  - Apply some small (m = 0.01) mutation to each reproduced agent that causes it to flip its strategy

Evolving PD strategies

- Initialise population N to random strategies
- Loop some number of generations
  - Loop for each agent (a) in the population
    - Select another agent (b) at random from the population
    - Play PD between (a) and (b) based on their strategies accumulate payoffs in agents
  - End loop for each agent
  - Reproduce a new population of size N probabilistically in proportion to fitness and apply mutation with probability m
- End loop for number of generations

Note: Random pairing of strategies is sometimes called “mean field” interaction or “homogenous mixing”. Reproduction without cross-over is called “asexual reproduction”.

Evolving PD strategies

- In this case with simple (pure) PD strategies and mean field mixing…
- Evolution will quickly lead to all defect dominating the population
- This is called an **Evolutionary Stable Strategy** (or ESS)
- ESS means that if all the population are using a given strategy then no other strategy can "invade" the population through mutation (when mutation is low)
- In game-like interactions, many ESS are Nash equlibria
- Hence a link is found between game theory and evolutionary theory which biologists discovered and applied

Book: John Maynard Smith (1982) Evolution and the theory of games. Oxford University Press

Sociobiology

- More generally the application of biological evolutionary approaches to understand social interactions is called Sociobiology
- When it is applied to human social systems it is can be highly controversial
- Critics worry it starts to look like “Social Darwinism” and overlooks the role of culture as the determinant of human social systems and behaviour
- We will not discuss this controversy here but it is worthwhile to be aware of it

Book: E. O. Wilson (1975) Sociobiology: The New Synthesis.

Evolving PD strategies

- More complex strategies can be evolved in this way and analysed to see if they are ESS
- Axelrod noted in his book that tit-for-tat was "collectively stable" (almost an ESS)
- The relationship between ESS, Nash and, say, Pareto efficiency is subtle and complex even in mean field models
- However, people can produce analysis (not just simulation) to determine these properties of strategies in known games

Paper: Nowak, Sigmund, Esam (1995) Automata, repeated games and noise. J. Math. Biol. 33: 703-722

Evolution of strategies

- Even if we can calculate ESS for given strategies this is not necessarily tell us the dynamics (trajectories) that evolution will take from any given starting point
- In simple systems “replicator dynamics” equations can be used to prove things (assuming no mutation!)
- In general, simulation experiments are used to see what happens when it gets complicated

Non-random interactions

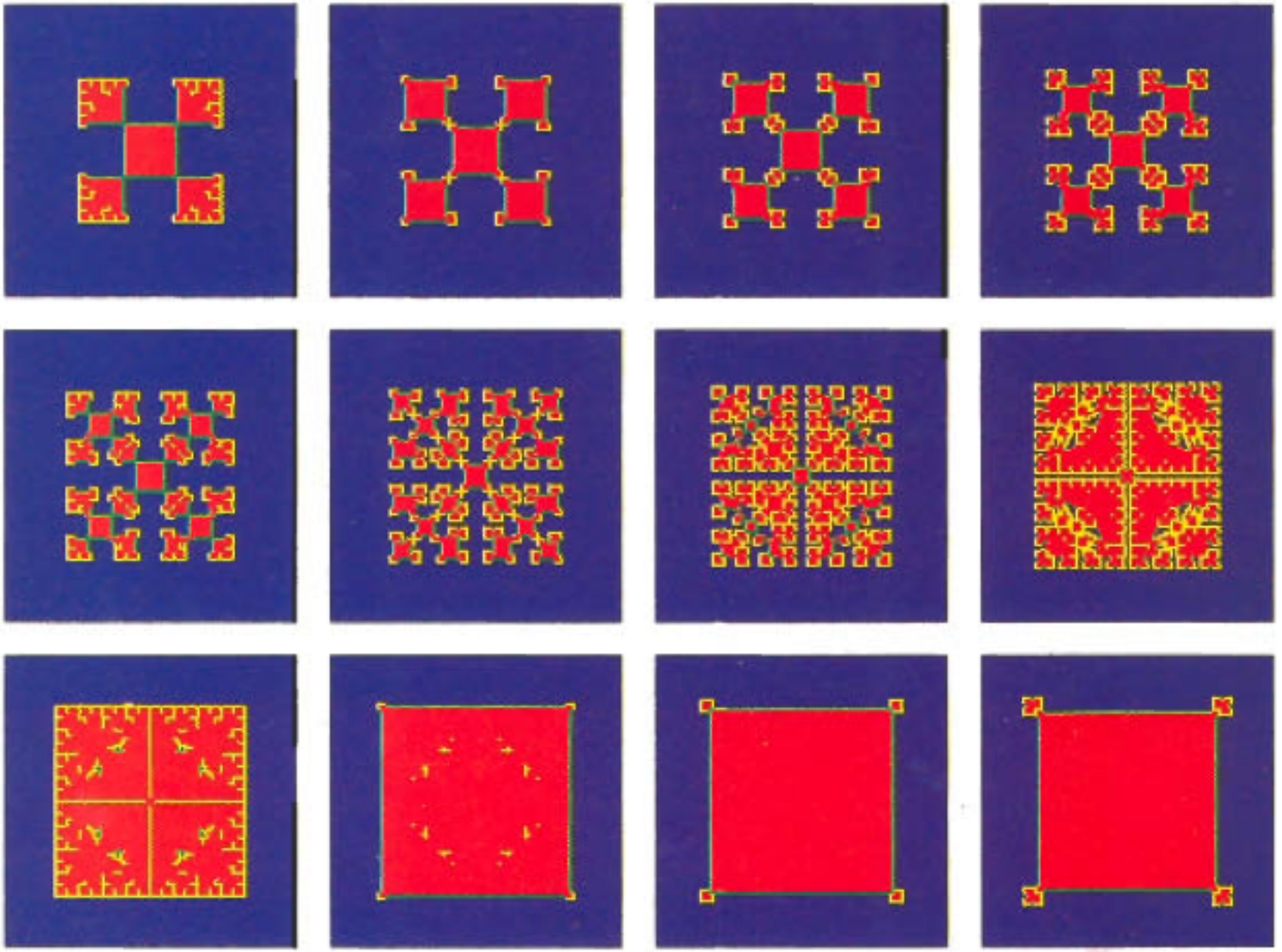
- Many forms of interaction in the "real world" are non-random
- Some work has explored this using a cellular automata (CA) where:
  - Each cell is either coop or defect state
  - Plays PD with each of it's neighbours (and possibly itself)
  - Copies the the strategy of fittest neighbour (or stays same if it is fittest)
  - Sometimes mutation is used sometimes not

See NetLogo model library/biology/evolution/altruism

Evolving PD on a CA

- In general it has been found that over a broad range of parameters:
  - Cooperation can be sustained
  - Dynamic patterns emerge over time
  - Groups of cooperators and defectors because they are spatially clustered create these interesting dynamics
  - Pretty patterns can be produced
- The argument is that many biological and social phenomena interact in space and this can be a major factor in sustaining the evolution of cooperation

Paper: Nowak, May (1993) The Spatial Dilemmas of Evolution. Int. J. of Bifurcation and Chaos, Vol. 3, No. 1. 35-78



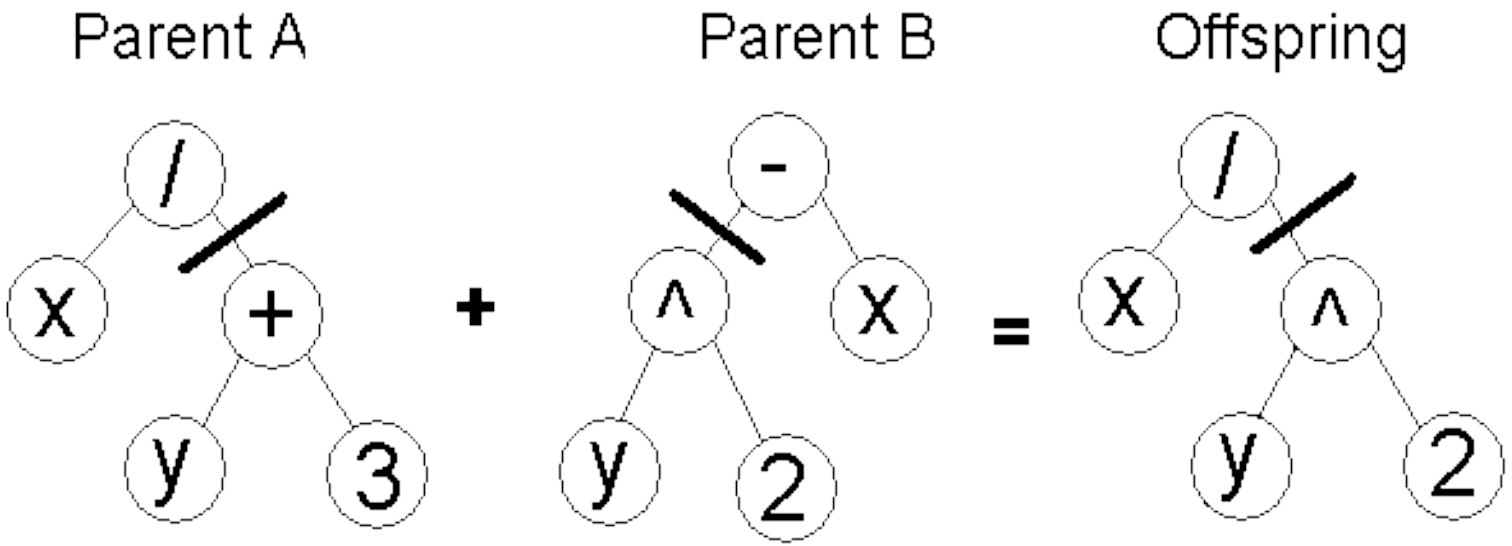
Taken from Nowak and May (1992)

Unknown coding of solution?

- Suppose we don't have simple space of solutions (or strategies) that each "chromosome" can code
- Can we use evolutionary algorithms without a simple coding of the solution space?
- Yes, evolve a computer program directly (or an artificial neural network)
- **Genetic Programming** (GP) uses simple (functional) languages and tree-like crossovers
- Such languages have to be "robust" to mutation and crossover – i.e. not "brittle" (unlike most computer languages where if you change one thing it breaks)
- In general GP are used to evolve small programs (or functions) for optimisation purposes

Book: Koza, J.R. (1992). Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press.

Genetic Programming



It is possible to evolve whole programs like this but only small ones.
Large programs with complex functions present big problems

Endogenous self-replication

- If we fully endogenise evolution then we should evolve the very process of reproduction itself
- This means endogenous selection – i.e. evolve things that can reproduce copies of themselves
- Remember von Neumanns self-replicating CA?
- How could we do this in a computer simulation?
- Tom Ray created a system (early 1990's) called "Tierra" that did just that to explore what has been termed **open ended co-evolution**
- This is considered formative work in an area called "**artificial life**" – what is Life?
- Artificial life tries to understand life by building it

Tom Ray's Tierra

- A simulated virtual machine with fixed size memory
- Agents are programs written in the assembly language of that machine (designed to be not too "brittle"):
  - Compete for processor and memory
  - Evolve through mutation, death and replication
  - Evolve novel ways to overwrite other programs in memory to reproduce
- Ray seeds system with an initial hand-coded self-replicating program
- Complex dynamic parasitic ecologies emerge
- Ray was a biologist / ecologist trying to understand complex interactions between living things

Paper: Tom Ray (1992) Evolution, Ecology, and Optimization of Digital Organisms. Santa Fe Working Paper: 1992-08-042
Short documentary: http://youtu.be/Wl5rRGVD0QI

Aside: Core War game

- A simulation game “core war” was developed long ago
- People design algorithms to fight other algorithms to take over the memory space in a vm running “redcode”
- Some recent contestants have been *evolved*
- On some level, could one view the entire internet as something similar to this?
- Yet winning algorithms will need to make people run them too – social engineering
- Viral programming was researched early in networking / computing but has got a bad name due to malware / botnets using the technique for nefarious purposes

See: http://en.wikipedia.org/wiki/Core_War

Cultural Evolution

- When behaviours, strategies, beliefs etc. are copied between agents within their lifetime and / or outside of genetic inheritance
- This is sometimes called cultural transmission or cultural evolution
- Hence, if there is some mechanism for replication (imitation), innovation (mutation) and selection (selective imitation) then:
- Evolutionary algorithms can be viewed as cultural evolution rather than genetic evolution
- Detail analysis has looked at gene-culture co-evolution based on this view (or dual inheritance theory DIT)

Book: Boyd, R. and P. J. Richerson. 1985. Culture and the Evolutionary Process. Chicago: University of Chicago Press.

Cultural Evolution (memes)

- Richard Dawkins introduced the term “meme” in his famous book “the selfish gene”
- His idea was to show that evolution can occur outside of biology in a similar Darwinian way
- The “meme” was the cultural equivalent of the “gene” in biology
- Can be thought of as any unit of culture that can be copied accurately – such as a song, a joke, a way of throwing stones far etc.
- A group of people started to create an area called “memetics” that would study this form of cultural evolution
- But it died out as a serious area of study (that’s evolution for you!)
- I have my own ideas why this happened!
- The idea of cultural evolution in a memetic sense is still sometimes used in scientific work but other terms are often used to describe it
- The term “meme” seems now to be reserved for “internet memes”

Book: Dawkins, Richard (1976). The Selfish Gene. Oxford University Press

Evolving the interaction structure *and* strategy in a PD simulation

- Suppose we have a set of nodes in network (graph)
- Each stores some maximum number of undirected links to other nodes
- Each stores a PD strategy (C or D)
- Periodically nodes play PD games with their neighbours (those linked to) accumulating an average fitness

Network rewire model

Each node *p* periodically performs a PD game interaction with a randomly chosen neighbor

Each node *p* periodically executes the following:

q = SelectRandomPeer() // from entire population

If $\text{utility}_q > \text{utility}_p$

drop all current links with high probability

link to node *q* and copy its strategy and links

mutate (with low probability) strategy and links

Network rewire movie

What is this showing?

- This is a form of "cultural group selection"
- Between peers using tournament selection
- Except it's not *really* group selection
- It's individual selection that dynamically creates social interaction structures that support cooperative groups
- Could such ideas be applied in real P2P?

Paper: Hales, D. & Arteconi, S. (2006) Article: SLACER: A Self-Organizing Protocol for Coordination in P2P Networks. IEEE Intelligent Systems, 21(2):29-35

Paper: Santos F. C., Pacheco J. M., Lenaerts T. (2006) Cooperation prevails when individuals adjust their social ties. PLoS Comput Biol 2(10)

More recent overview paper: Hales, D., Shutters, S. (2012). Cooperation through the endogenous evolution of social structure. Proceedings of the Complex 2012 conference in Santa Fe, NM. Dec. 5-7th 2012, Springer

More about this model?

- There are two videos of me giving lectures on this model:
  - (2007) @ TUD, http://youtu.be/z9H5FqDsJ24
  - (2011) @ Henley, http://youtu.be/c-iLEg9yuBo

Artificial Life

- If you are interested in Alife then you could do worse that start with this recent and comprehensive review of the area:
  - Aguilar W, Santamaría-Bonfil G, Froese T and Gershenson C (2014) The past, present, and future of artificial life. Front. Robot. AI 1:8. doi: 10.3389/frobt.2014.00008
- Chris Langdon (remember his loop CA?) started the area and there is a journal and conferences.

John Holland

- Interdisciplinary thinker, background in physics, maths, comp. sci, psychology.
- Invented modern genetic algorithms
- Interested in “complex adaptive systems”
- Worked with models similar to Ray’s Tierra. Developed “Echo” model
- Produced the “classifier” agent which evolves it’s internal (production) rule set
- Has written widely on self-organisation and emergence in easy to understand language

Books: John Holland (1998) Emergence: from chaos to order. Perseus Books.

Holland 2008 lecture video: http://youtu.be/6aN6PlsvkpY

Richard Dawkins

- Background zoology / biology
- Great populariser of evolution as a way to explain life in his famous book “the selfish gene” and even culture (meme)
- He takes a strong “gene’s eye view” which is not accepted by many evolutionary biologists (e.g. Stephen J. Gould)
- Often debates those who believe in God saying that evolution is a better explanation for life
- Argues along with Daniel Dennett that the “abstract evolutionary algorithm” can explain a lot of complex biological, social and cultural phenomena
- Some have gone even further than this and argue evolution could explain the laws of physics themselves (but that’s another story!)

Aside: Pierre Teilhard de Chardin

- Jesuit priest, palaeontologist, philosopher (1881-1955)
- Tried to fuse Christian doctrine with evolutionary theory
- Believed evolution was following a *teleological* path towards the "omega point" (modern form: Singularity)
- Where the "noosphere" (mind) would supplant the biosphere (matter) towards our absorption into God
- Was suppressed in his time by Catholic Church (because violated orthodoxy) but resurrected by Pope Benedict XVI more recently
- His ideas not widely accepted by Biologists or Theologians
- Teleology is not part of the modern synthesis (orthodoxy) of evolutionary theory!

Readings and Questions

- Readings
  - Flake (1998) Chapter 5 – Adaptation
  - Gilbert et al (2005) Chapter 10 – Learning and Evo. models
- Questions
  - Can you think of a simple game in which the ESS is obviously not a Nash equilibrium?
  - Some claim TFT in the PD is an ESS others say it's not strictly an ESS. Why is this?
  - How could we use evolution to help design practical P2P protocols? Is it possible?
  - Is abstract evolution tautological? If so does it tell us anything about the "real world" ?