University of BRISTOL

More Advanced GA Concepts

Seth Bullock

bristol.ac.uk

University of BRISTOL

Contents

This lecture covers a number of Evolutionary Computation concepts:

- Premature Convergence
- Neutrality
- Quasi-Species
- Epistasis
- Problem Modularity
- No Free Lunch
- Evolvability

University of BRISTOL

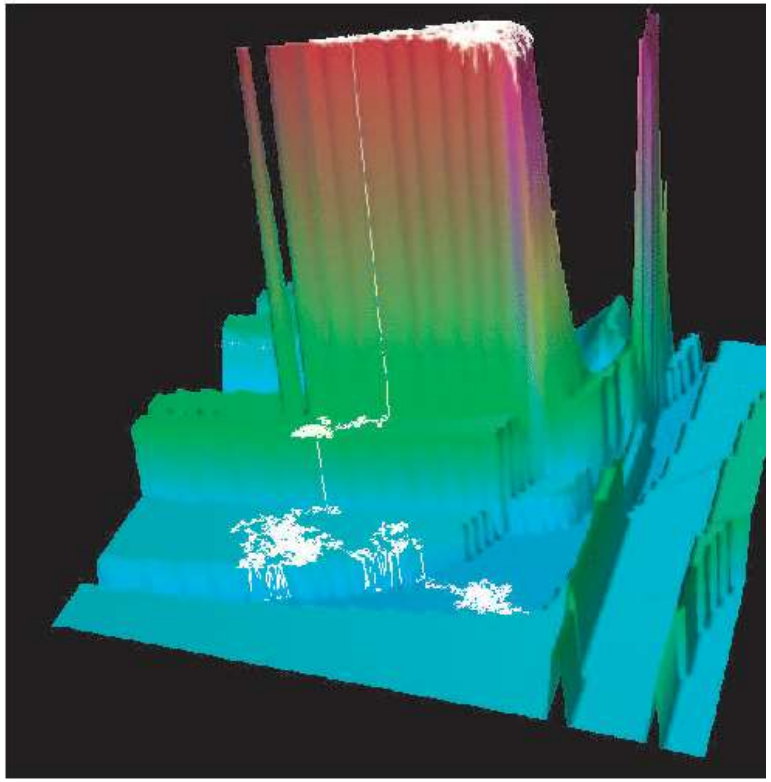
The Problem of Local Optima

- Hill-climbers get stuck on local optima; Exhaustive search does not
- GAs are less likely to get stuck on local optima than hill-climbers:
  - …because they maintain and evolve a *population of solutions*
  - …because genetic operators may generate a *wider range of neighbours*
  - These reasons exploit and rely on *diversity* in the evolving population
- Local optima are still a problem as *pop diversity is not guaranteed*
- Should a population of solutions end up *converged* within one basin of attraction they can find it difficult to escape
    - Getting stuck like this is called: *premature convergence*

University of BRISTOL

Neutrality

- Neighbouring genotypes with equal fitness are ‘selectively neutral’
  - …even if their phenotypes are different.
  - Evolution cannot choose between them…
  - …which results in ‘evolutionary drift’
- Neutrality is often easy to overlook and hard to visualize, but it can be key…
  - E.g., ‘neutral networks’ that percolate the search space may allow converged populations to escape “local optima”

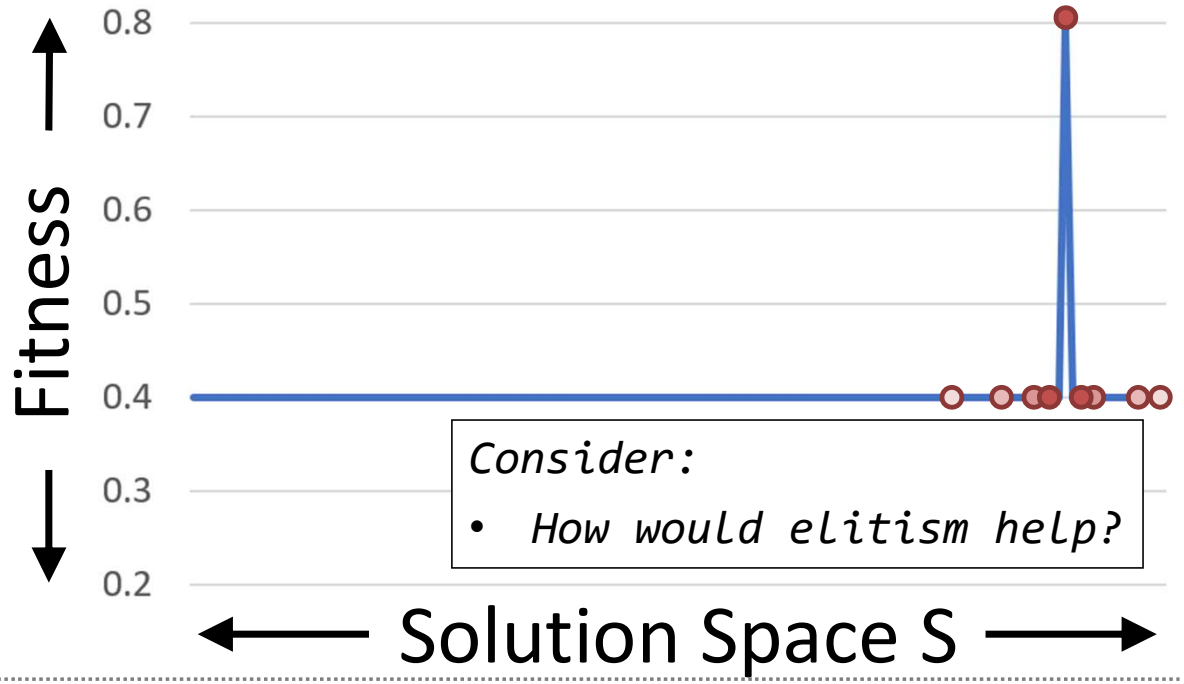


Barnett (2002).
Explorations in Evolutionary Visualisation

University of BRISTOL

The Invisible Needle

- Recall the “needle” fitness landscape we saw in a previous lecture
- The needle is hard to find because there’s no gradient to climb
- But even if we start the pop *on the needle*, we still might not find it…
  - Notice: it’s very rare for an offspring to be an exact copy of their parent
  - So a perfect solution will tend to have unfit offspring…



University of BRISTOL

The Quasi-Species Concept

- Manfred Eigen & Peter Schuster developed the *quasi-species* concept to explain what's happening here:
  - High GA mutation rates (much higher than for DNA) mean that the evolving population is a *cloud* of points on the landscape: a *quasi-species*
  - If the selection pressure on the population to reward good solutions…
  - …is overcome by the mutation pressure that constantly corrupts them…
  - …then the population effectively cannot even "see" the needle.
- Conversely: if an evolving population *is* able to find a good solution, *and stay there*, we can infer that the solution must be surrounded by other pretty good solutions – it is *robust*

Epistasis

- If we're very lucky, fitness is a *linear function* of the gene alleles:
  - The contribution of each gene to fitness is *independent* of other genes
  - `1-max` is a linear fitness function: fitness = the number of 1s in a bitstring
  - Trivial to optimize: optimize each gene independently
- But almost always, fitness is a *non-linear function* of gene alleles:
  - The contribution of genes to fitness is *interdependent* to some extent
  - The best allele choice at one gene *depends* on the alleles at other genes
  - E.g., baby skull size ↔ size of birth canal; weapon range ↔ visual range
    - This is *epistasis*: more epistasis makes a search problem harder

University of BRISTOL

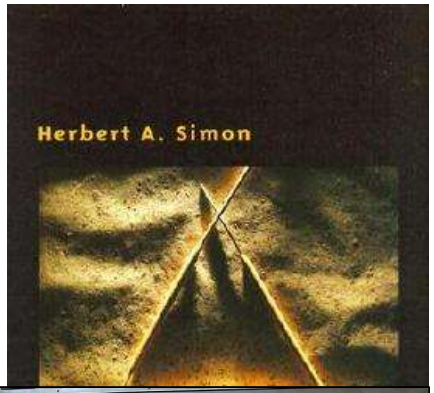
Epistasis

- The presence of epistasis ⇔ the existence of local optima.
  - Epistasis / local optima are different ways of describing the same thing:
  - Non-linearities in the mapping of fitness onto the search space
- Epistasis means we can't optimize each gene *independently*
  - Instead our algorithm must *co-optimize* the alleles of multiple genes
- How big is the set of interacting genes? (the 'order' of interaction)
- How many sets of interacting genes? How do they overlap?
- More generally, what is the *structural modularity* of the problem?

University of BRISTOL

Modular Decomposability

- Herb Simon introduces a nice way of thinking about problem modularity in his book *The Sciences of the Artificial* (1969): trying to crack a safe
  - A bank safe has *N* dials, each with 100 different settings
  - Only *one* setting is correct on each dial
  - *All* dials need to be correct to open the safe
  - (Recall that Dawkins uses the same example in his *Horizon* episode…)
    - How hard is it to open the safe?



University of BRISTOL

Modular Decomposability

The answer depends on the *modularity* of the safe

- For a safe with perfectly silent dials: we must search all 100^N possibilities
  - We can expect to have to check ½ of them => $100^N/2$ to crack the safe
  - No modularity; no useful problem structure; solution has strong epistasis
- But if each dial gives a little *click* only when it is at the correct setting:
  - We can expect to have to check ½ of the 100 possibilities *for each dial*
  - => $100N/2$ tries to crack the whole safe (much less time)
  - Full modularity; useful problem structure; solution has zero epistasis
    - The same as the "Methinks it is like a weasel" & 1-max problems

University of BRISTOL

Partial Decomposability

- But real problems tend to lie somewhere in between:
  - Partial modularity; useful problem structure; solution has some epistasis
  - Compare: “Methinks…” vs. evolving any correct English sentence
  - English sentences have modules (words) that depend on each other
- Watson (2006): consider a safe with dials that are sensitive to each other. For each dial, *n* settings (including the correct one) give a little click:
  - *n*=1 (full modularity); *n*=100 (no modularity); *n*=20 (some modularity)
  - We must try $n^N/2$ combinations (much less than $100^N/2$) to crack the safe
  - (After we find which are the *n* clicky settings on each dial: 100N tries)
    - What if *n* is halved for every dial that is in the correct setting?

University of BRISTOL

No Free Lunches

- Wolpert and Macready (1997): No Free Lunch for Optimization:
  - When algorithm performance is averaged across *all possible problems*:
  - *Any two optimization algorithms will exhibit equivalent performance*
  - *i.e., no optimization algorithm can do better than blind random search*
- The insight here is that every search algorithm other than random search has a *search bias*: it works on a hunch, gamble, or heuristic
- For every time the hunch pays off, there's a time where it doesn't
- Perhaps need to shift focus to *satisficing* rather than optimising…
- Algorithm quality depends on the *problem structure* that it faces

University of BRISTOL

Learning How to Search

- Recall that the structure of *S* (including it's modularity) is influenced by our choice of *representation* and *genetic operators*
- So, we can restructure the space by making different choices
- Good choices could be the difference between success and failure
  - Might it be possible to learn which operators to use when?
  - Or to learn a good genetic representation?
  - ..either for a whole class of different, related problems (e.g., timetables)?
  - ..or *online* for one problem *as we are trying to solve it*
    - Could a GA implement evolution that gets more + more powerful?

University of BRISTOL

The Evolution of Evolvability

- Natural evolution has got more powerful over time:
  - cf. the 'major transitions' in evolution (Maynard Smith and Szathmáry)
  - E.g., Single=>Multi-cellularity; A-sex=>Sex; Non-Social=>Social/Cultural
- In fact, Dawkins' paper at the first Artificial Life conference is on what his *Biomorphs* tell us about "the Evolution of Evolvability".
  - 'The ability of a population to generate *adaptive* genetic diversity'
  - (Here *adaptive* means 'well adapted' to the problem at hand)
  - i.e., evolvability is: how good is a population at evolving
    - A poor GA has low evolvability; Nature exhibits high evolvability

University of BRISTOL

The Evolution of Evolvability

- More subtle mechanisms in nature:
  - DNA repair – maintains some parts of the genome more carefully
  - Chromosome Organisation – nearby genes less likely to be disrupted
- Genetic Algorithms can look to exploit similar tricks:
  - Analyse how fitness varies in the current population – build a model
  - Use this model to predict what type of genotypic variation will be best
  - Effectively learning the structure of the problem during evolution
- E.g., “How can evolution learn?” (Watson & Szathmáry, 2015)
  - http://www.bbc.com/earth/story/20170301-life-may-actually-be-getting-better-at-evolving

University of BRISTOL

Example Questions

- Complete the missing fitness entries in the two tables of genotypes and their associated fitness values, below, such that there is (i) no epistasis, (ii) some epistasis. *[2 marks]*
- Give an example of two aspects of a real-world problem that are linked epistatically. Explain your answer. *[4 marks]*
- A problem comprises two complex but almost independent sub-problems. How should a GA's genotype be structured, and what type of crossover should be used? *[6 marks]*

University of BRISTOL

Thank you!

University of BRISTOL

Coevolution

Seth Bullock

bristol.ac.uk

University of BRISTOL

Fitness Function Design Is Hard

- As we have seen, a GAs success or failure is often down to our design decisions:
  - Our choice of genetic representation of the phenotype space
  - Our choice of genetic operators and selection scheme
  - Our design of a fitness function
- The third of these is particularly trying:
  - Many fitness functions are hard for GAs to make progress on due to local optima, ruggedness, neutrality, deception, etc.

University of BRISTOL

Coevolution

- But what if we could skip that tricky bit?
- Real evolving populations *do not consult a fitness function* to find out how many offspring each individual is allowed.
  - They just get on with it
  - Organisms co-operate & compete with the other organisms in their environment, and have offspring as part of this activity
- Coevolution: *when multiple populations of evolving organisms influence each other's evolution*

University of BRISTOL

Coevolution

- Coevolution = An automatic fitness function:
  - There is no fitness function designer in nature
- Coevolution = An implicit fitness function:
  - Instead of being set an *explicit* objective target by the fitness function, real organisms are assessed *relative* to their current environment
- Coevolution = A dynamic fitness function:
  - Instead of being assessed against a *fixed* target, real organisms cope with environments that change/evolve over time

University of BRISTOL

Example

- We would like to evolve a robot goalkeeper for RoboSoccer
  - Designing a fitness function has proven to be quite hard
- Instead, we decide to *coevolve* a population of goalkeeper robots against a population of robot penalty takers
  - We start off with random penalty takers and random goalkeepers
  - As soon as penalty takers start to be able to beat random goalkeepers..
  - ..there is pressure on the goalkeepers to get better..
  - ..putting pressure on the penalty takers to get better.. ..and so on.
    - If things work out, coevolution generates a good solution without us having to define a 'good' strategy via an explicit fitness function

University of BRISTOL

Coevolutionary GA

- Co-operative Coevolution:
  - Each population deals with a separate part of the overall problem
  - A solution combines individuals from each population
  - Appropriate when the overall problem is highly structured
- Competitive Coevolution:
  - One population of “solutions” to the problem that we are interested in coevolves against a population of “challengers” that must be defeated
  - Solutions are fit if they defeat many challengers.
  - Challengers are fit if they defeat many solutions.
    - Reminiscent of natural host-parasite or predator-prey coevolution.

University of BRISTOL

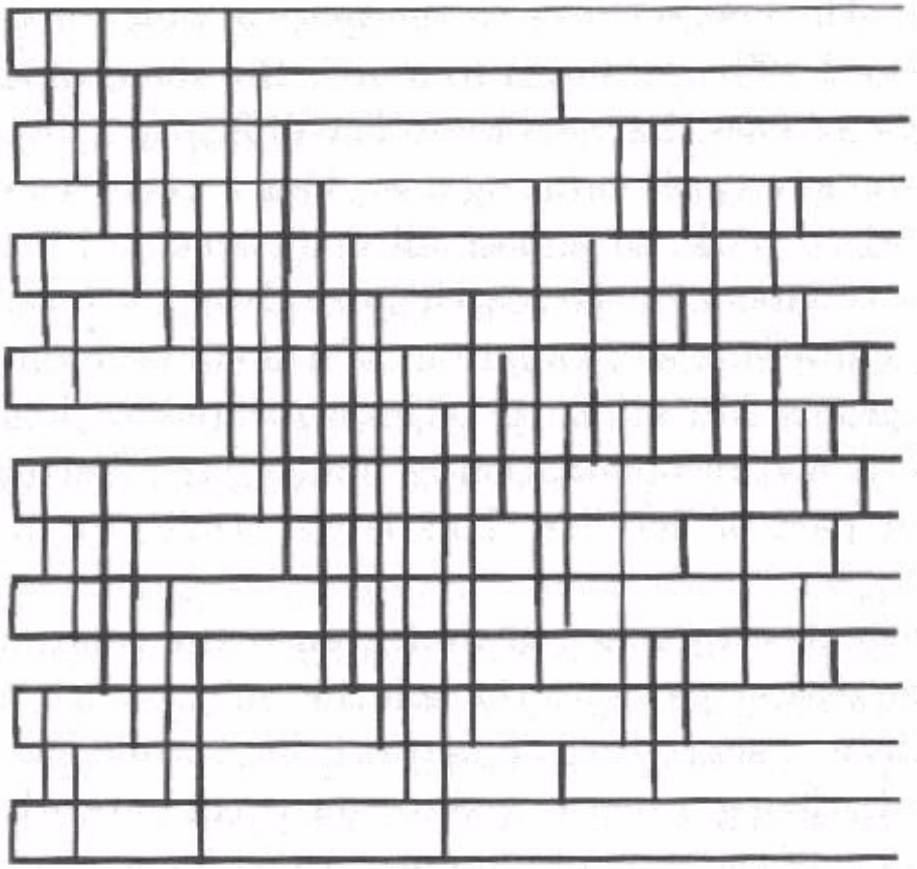
Examples

- The first coevolutionary GA was due to Hillis (1990) and was applied to solving a list-sorting problem.
  - What's the most efficient way to sort a list of length *n*?
  - Hillis pitched a population of 'hosts' (sorting networks) against a population of 'parasites' (each a set of lists to be sorted)
  - He initialised his host population with a reasonable solution and initialised his parasite population with random lists
  - Hosts rewarded for sorting parasite lists; parasites for being unsorted
  - In a few hours, he was able to coevolve a sorter that used 61 'gates'
    - Engineers had taken years to achieve a sorter with 60 gates

University of BRISTOL

List Sorting Example

- History of *n*=16 sorting networks
  - Batcher & Knuth (1964) **63** gates
  - Shapiro (1969) **62** gates
  - Green (1969) **60** gates (best known)
  - Hillis (1990): Evolved **65** gates…
    Hillis (1990): Coevolved **61** gates…
- Used in telecoms switches
- Now also relevant to GPUs
  - Saving even one gate can make a huge performance difference



University of BRISTOL

Why Did Regular Evolution Fail?

Evolution (~65K hosts for 5K generations) had problems:

1. Many local optima: reasonable sorters are surrounded by rubbish
2. Exhaustive assessment is prohibitively slow
   - There are 16! different lists to consider…
   - (Even reducing to 2^{16} binary lists is still a lot to check for each sorter)
3. But assessing against a random sample of lists is also inefficient:
   - Most randomly generated lists are sorted by reasonable sorters
   - Really challenging lists are rare, so there's a lack of pressure to improve

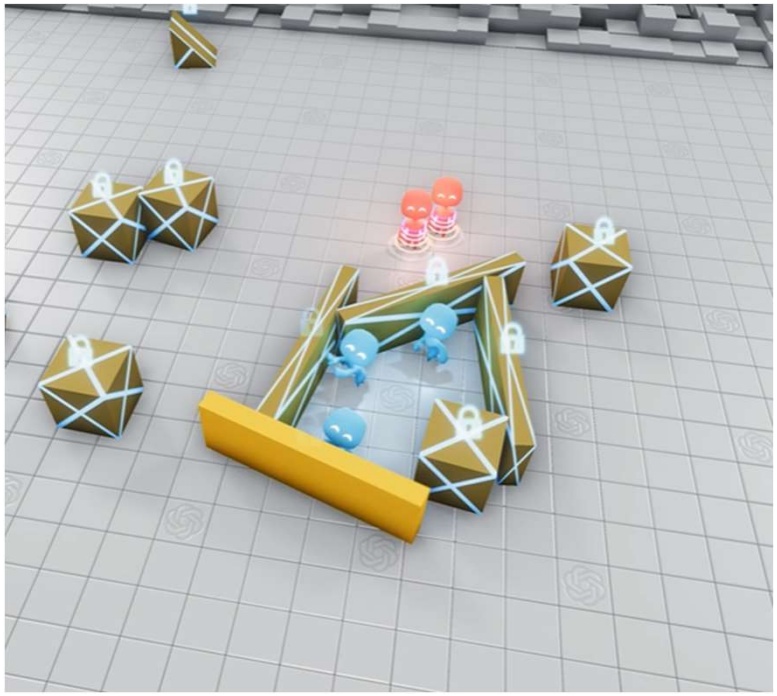
University of BRISTOL

Why Was Coevolution So Much Better?

- Coevolution solved these problems automatically:
  - Instead of random lists, sorters needed to sort sets of lists that were selected to become harder and harder.
  - If sorters got stuck on a local optimum, parasites were under pressure to punish that strategy by finding the lists that it couldn’t sort.
  - When improvements in the sorter population meant selection pressure got weaker, parasites were selected to get still harder.

Recall

From Karl Sims' (1994) *Blockies* to OpenAI's (2019) *Hide & Seek:*



University of BRISTOL

Thank you!

University of BRISTOL

Coevolution II

Seth Bullock

bristol.ac.uk

University of BRISTOL

Problems

- Coevolutionary arms races are supposed to *escalate* to increasing levels of sophistication… but sometimes that doesn't happen:
  - Stagnation: populations reach a poised state in which there is a turnover of individuals in the population but no phenotypic progress
  - Cycling: populations repeatedly cycle through the same weak strategies as if they were playing a game of scissors-paper-stone (the Red Queen)
  - Disengagement: every host is beaten by every parasite, all selection pressure is extinguished and coevolution effectively ceases (see Lab 2)
  - Measuring Progress: without a fitness function it can be difficult to assess whether genuine progress is being made by coevolving populations.

University of BRISTOL

Diversity

- To a large extent the problems with coevolution are familiar:
  - Our old foe: the loss of genetic and phenotypic diversity
- How can we stop the populations from converging and becoming vulnerable to stagnation, cycling, disengagement?
- Three different ideas:
  - Population structure
  - Fitness sharing
  - Reduced virulence

University of BRISTOL

Population Structure

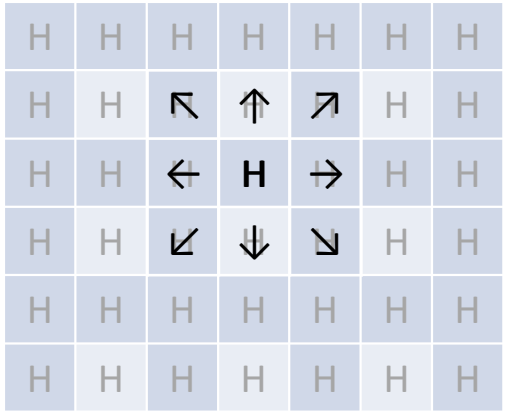
- The natural world is not randomly mixed up like an evolving soup.
- Populations are structured such that not every organism can interact with or compete directly with every other organism
- This is why island ecologies can be so distinctive and interesting
- GAs can do something similar
  - split populations into quasi-isolated sub-populations called ‘demes’
  - spread a population over a space and only allow local interactions

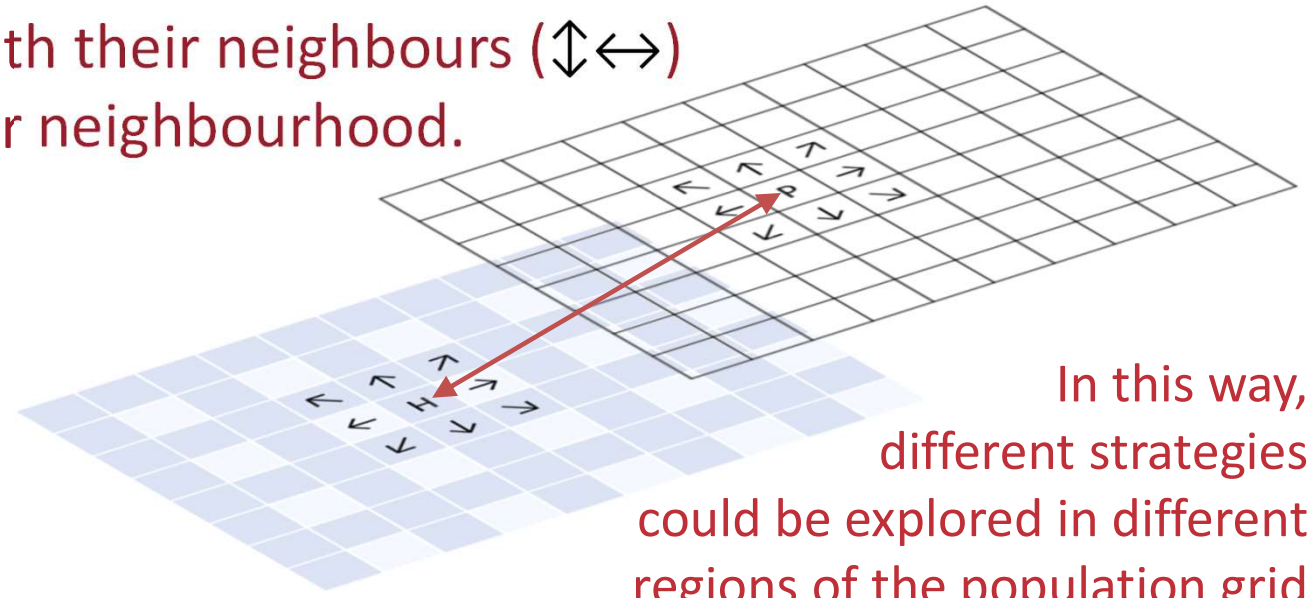
http://earthporm.com/alien-island/ Ben Shread © MoD/Crown Copyright

University of BRISTOL

Population Structure

- Hillis employed this idea in his list sorting work:
  - Each host was placed on a grid. Parasites were placed in a different grid.
  - Each host played the parasite at the same location in the other grid (↔)
  - Individuals competed with their neighbours (↕↔) to leave offspring in their neighbourhood.



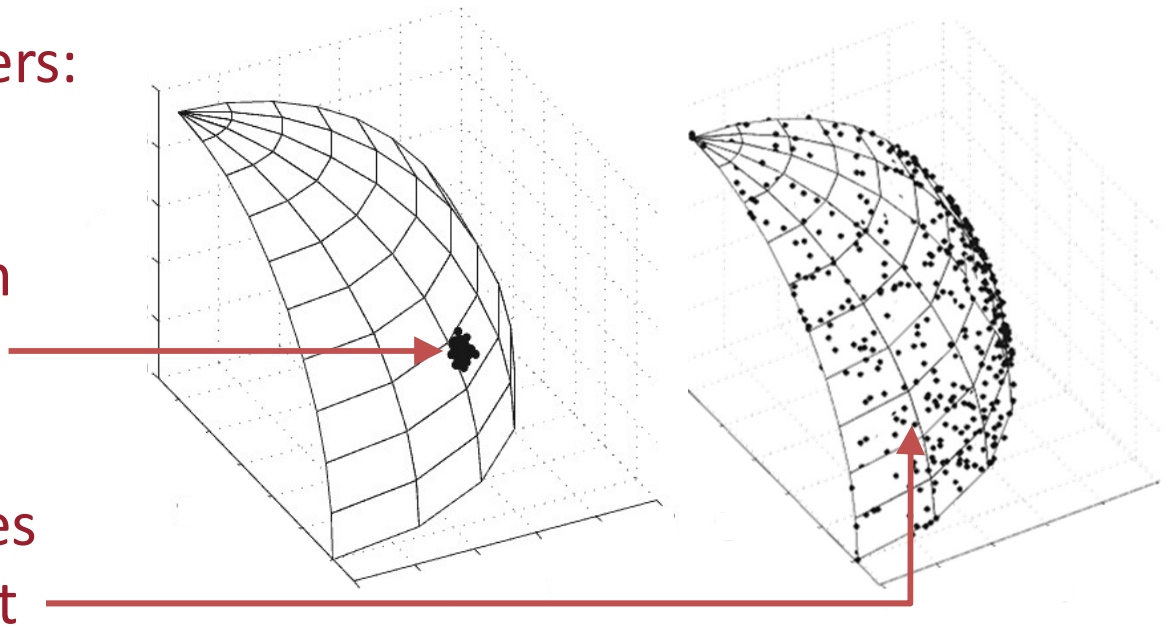


In this way, different strategies could be explored in different regions of the population grid

University of BRISTOL

Fitness Sharing

- A more deliberate way of maintaining diversity is to punish solutions that are similar to other solutions in the population:
  - Define a similarity measure (e.g., genetic hamming distance < threshold)
  - If solution *i* is similar to *n* others:
    - $f'_i = f_i / \sum_{j=1}^{j=n} \mathrm{sim}(i,j)$
  - Now, instead of converging on one region of this manifold of equally fit solutions…
    - …fitness sharing pushes solutions to spread out



University of BRISTOL

Reduced Virulence

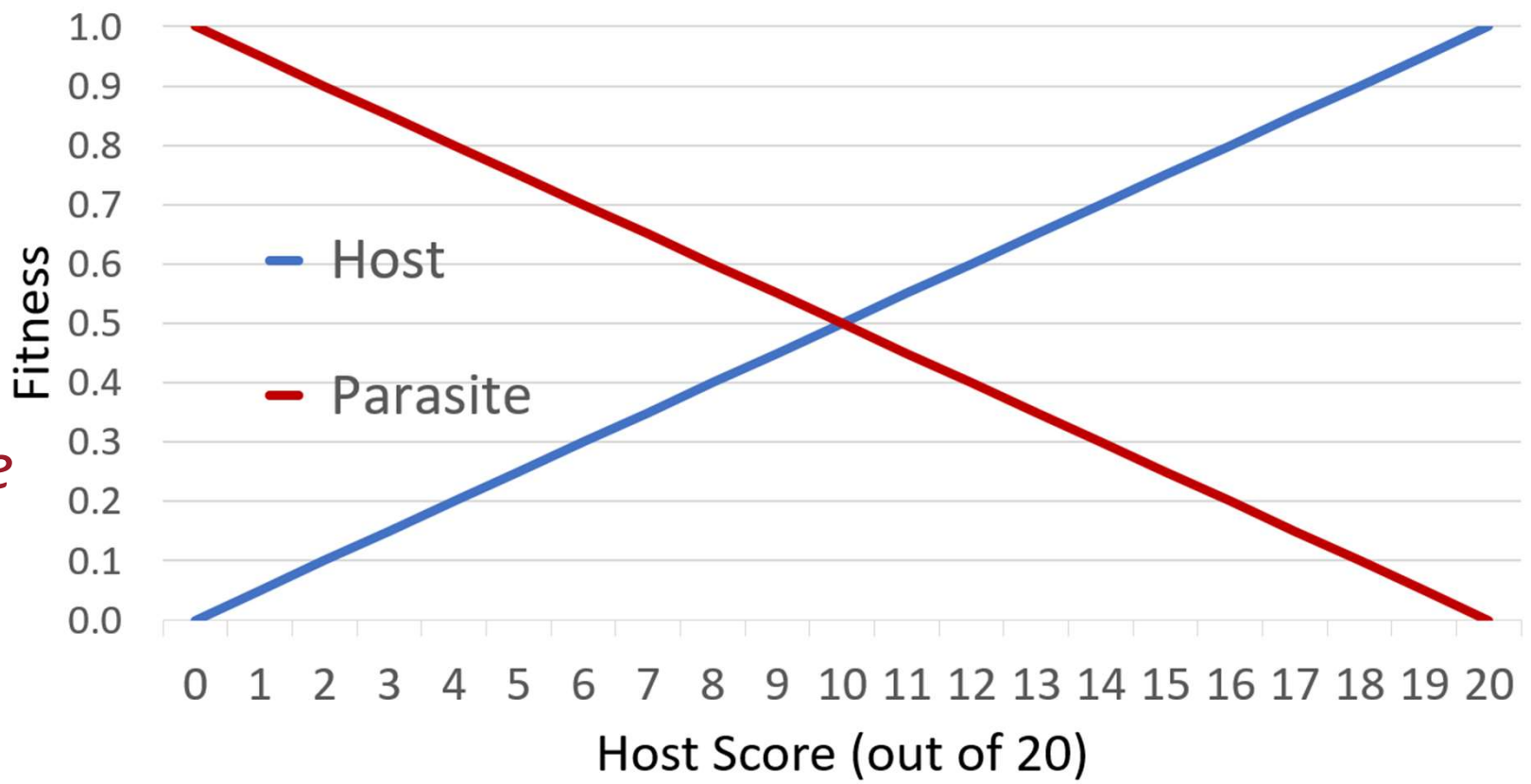
- Natural parasites / viruses often reduce their *virulence* over time
  - …it's not in their interest to kill their host quickly.
  - Consider the common cold or a tapeworm vs. extremely virulent Ebola
- But parasites in a coevolutionary GA are often rewarded for being *maximally damaging* to the hosts that they are assessed against…
- Our initial random robot goalkeepers won't benefit from facing penalty takers that smack the ball into the top corner every time.
- It would be better to start off with shots that are easy to save, and then get gradually harder…

University of BRISTOL

(Normal) Maximal Virulence

Normally, when a parasite plays against a host:

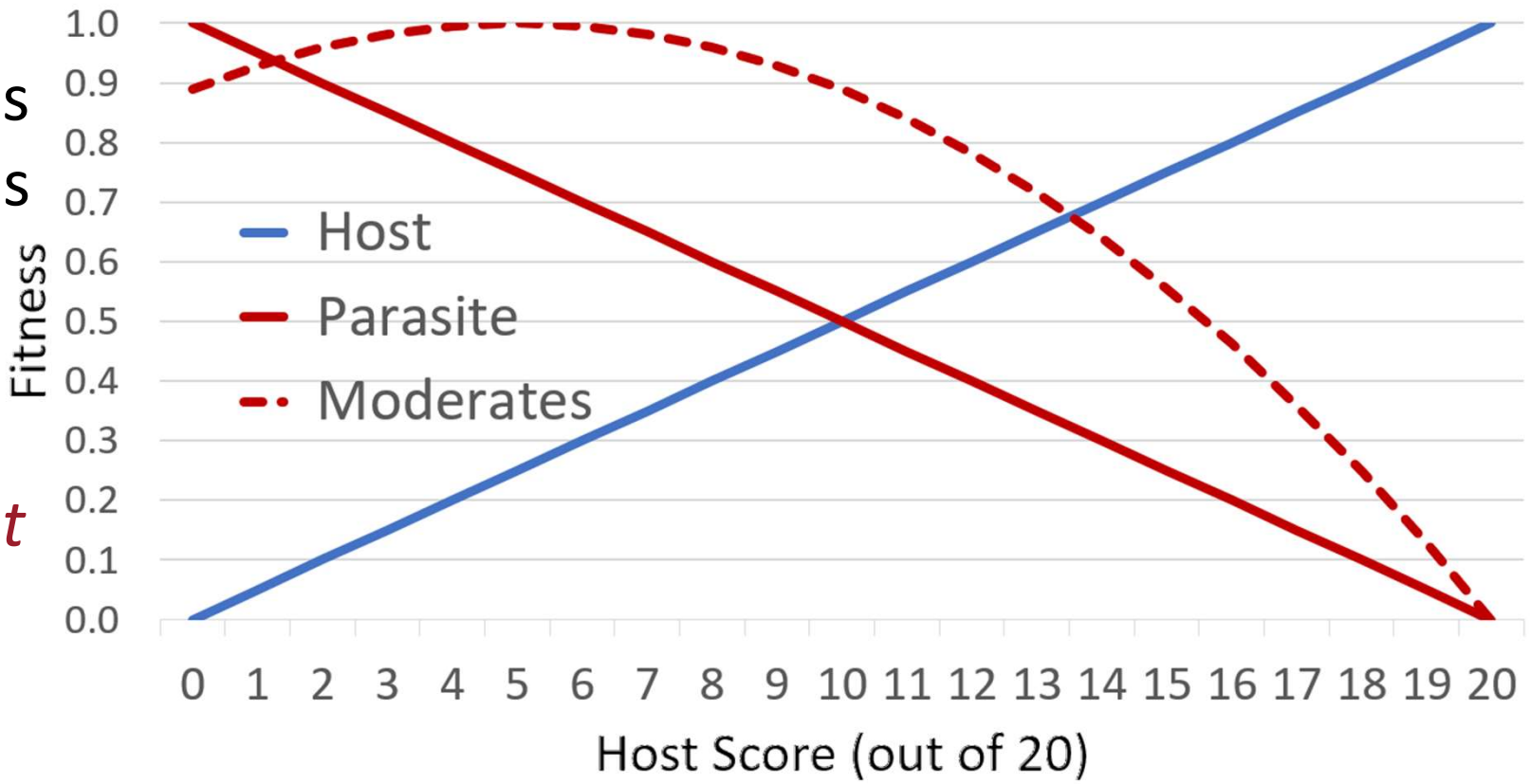
- If the *Host* scores *x* …
- … the *Parasite* scores 1-*x*



University of BRISTOL

Reduced Virulence

A ‘moderate virulence’ fitness function rewards parasites most for defeating hosts *most of the time, but not all of the time*…



University of BRISTOL

Coevolution

- More on these ideas in this weeks' GA Lab
- Ahead of the lab, please take a look at
  - Cartlidge, J. & Bullock, S. (2004). Combating coevolutionary disengagement by reducing parasite virulence. *Evolutionary Computation*, **12**(2), 193-222.
  - This lays out the details of the virulence idea, and introduces a simple game that we will be using to explore it.
  - It's a long paper, so don't feel you need to deep read all of it. ☺
    - Just focus on sections 1-4 (15 pages)

University of BRISTOL

Free Lunches?

- Recall Wolpert & Macready's (1995) No Free Lunch theorem
  - No search alg. can do better than random across *all* problems.
- More recently, in 2005, the same authors claim that *coevolutionary* free lunches are possible…
  - By using a separate (non-random) population to steer its search, a search algorithm may be able to always rule out some bad solutions, and thus out-perform random search…
- Check this stack exchange post for some intuitions…

University of BRISTOL

Example Questions

- Give an example of coevolution from nature. *[1 mark]*
- Name 3 problems that beset coevolutionary GAs. *[2 marks]*
- Why did Hillis embed his populations on a grid? *[4 marks]*
- What is premature convergence in a GA population. How do coevolutionary GAs hope to avoid it? *[4 marks]*
- Explain why a coevolutionary GA might cycle through the same poor solutions over and over again, rather than finding a better general purpose solution. *[6 marks]*

University of BRISTOL

Thank you!

University of BRISTOL

Artificial Life

Seth Bullock

bristol.ac.uk

University of BRISTOL

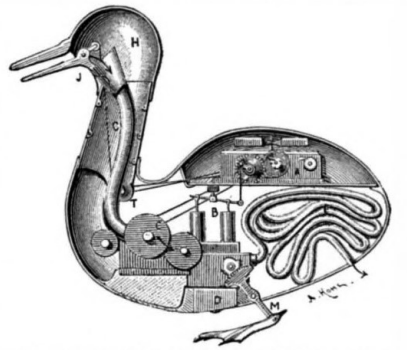
Alife and AI

- Artificial Life is a sister discipline for Artificial Intelligence
- Both take a functionalist approach to a mysterious subject:
  - AI: an artificial system can be *intelligent* if it is organised right
  - Alife: an artificial system can be *alive* if it is organised right
- Like AI, Alife combines a Strong project and a Weak project
  - Strong Alife: creating life inside a computer or in a robot
  - Weak Alife: understanding life by building life-like systems
    - Alife's pithy strapline: "The study of life-as-it-could-be"

University of BRISTOL

History

- Chris Langton named + founded Artificial Life in 1987
- But Alife has been going on for much longer than AI
  - Frankenstein, Golem, various ‘Automata’ and ‘Living Machines’:
  - Vaucanson’s Duck (1739): quacking, drinking eating, “excreting”
- The biggest names in CS were in fact Alifers…
  - Babbage: Evolutionary Models of Miracles
  - Turing: Models of Biological Morphogenesis
    - Von Neumann: Self-Reproducing Machines



INTERIOR OF VAUCANSON'S AUTOMATIC DUCK.
A, clockwork; *B*, pump; *C*, mill for grinding grain; *F*, intestinal tube; *J*, bill; *H*, head; *M*, feet.

University of BRISTOL

Self-Reproducing Machines

- How could a machine make a more complex copy of itself?
- This conundrum sounds impossible – paradoxical
- Von Neumann arrived at a two-part solution in 1948:
  1. Information encoding the structure of a machine …DNA
  2. Machinery that uses the information to make a new machine + a new copy of the information …cellular machinery
- In doing so, JVM predicted the contents of our cells…

…before the 1952 discovery of the DNA double helix.

University of BRISTOL

Universal Constructors

- But John Von Neumann went further than theorising and actually designed the first self-replicating machine!
- The first “Cellular Automaton” (CA):
  - A grid of cells, each containing a value
  - All values are updated according to a set of rules applied to each cell grid
    - Careful design of the rules plus the initial state of the grid, and…

University of BRISTOL

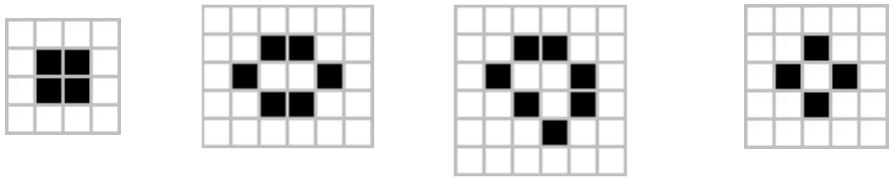
Conway's Game of Life

- You may be familiar with the most famous CA: Game of Life
- John Horton Conway (1937-2020) created the *Game of Life* in an effort to make a CA that had interesting behaviour:
  - Proliferating, repeating patterns; universal construction; etc.
- He settled on the following simple rules:
  - Each cell has a binary state – it can be alive or it can be dead
  - A live cell with >3 living neighbours or <2 living neighbours dies
    - A dead cell with three living neighbours becomes alive

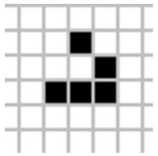
University of BRISTOL

Conway's Game of Life

- Some configurations are static:



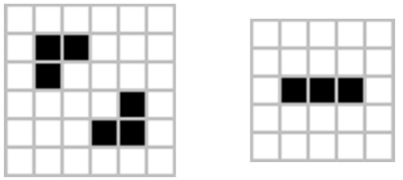
- Some oscillate on the move:



  - A "glider"

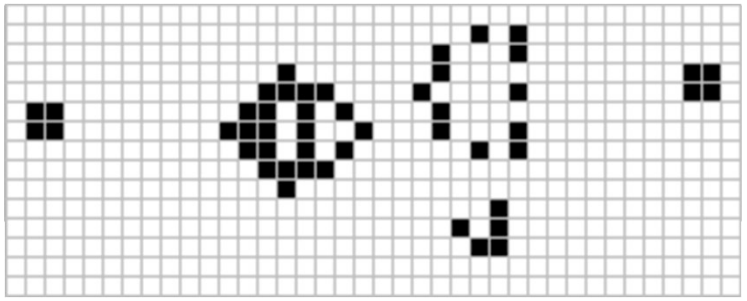
- Some destroy each other when they collide in the right way…

- Some oscillate in place:



  - A "beacon" and a "blinker"

- Some *generate* other structures: e.g., a *glider gun*

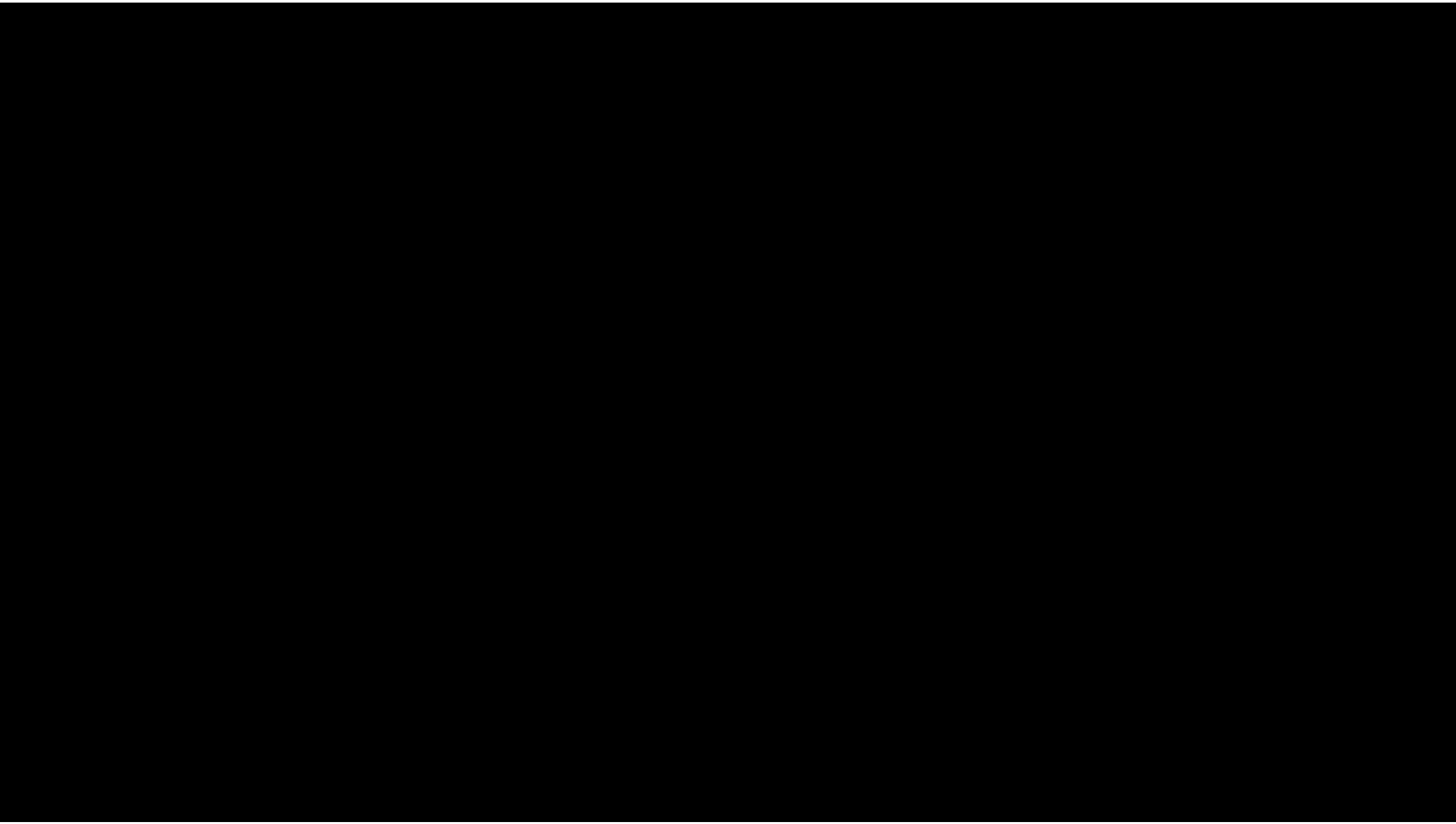


  - Amazingly, the Game of Life is provably Turing Complete…

Life in Life

Phillip Bradbury

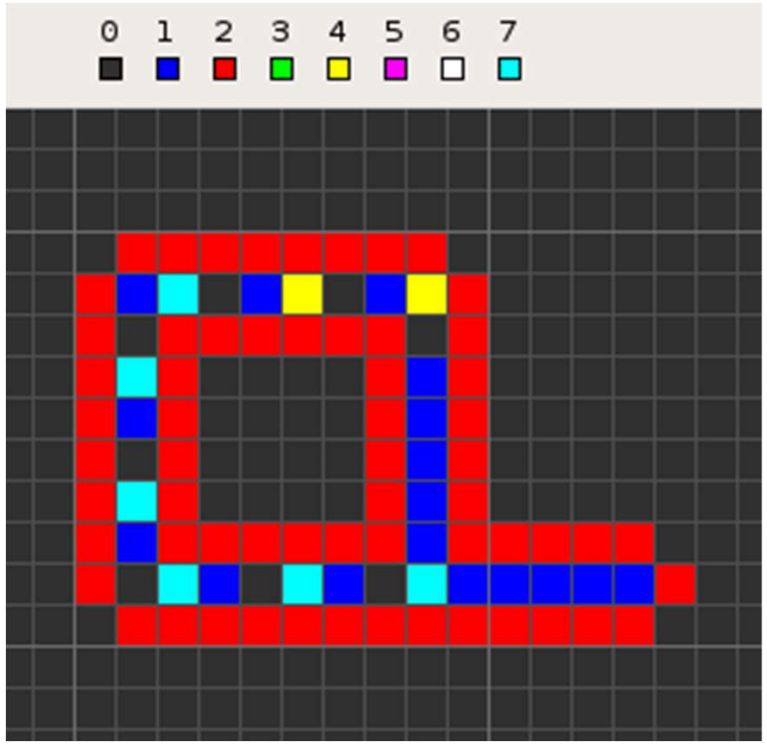
Roger Pincombe



University of BRISTOL

Langton's Loop

- Langton studied CAs + made a more simple self-replicator:

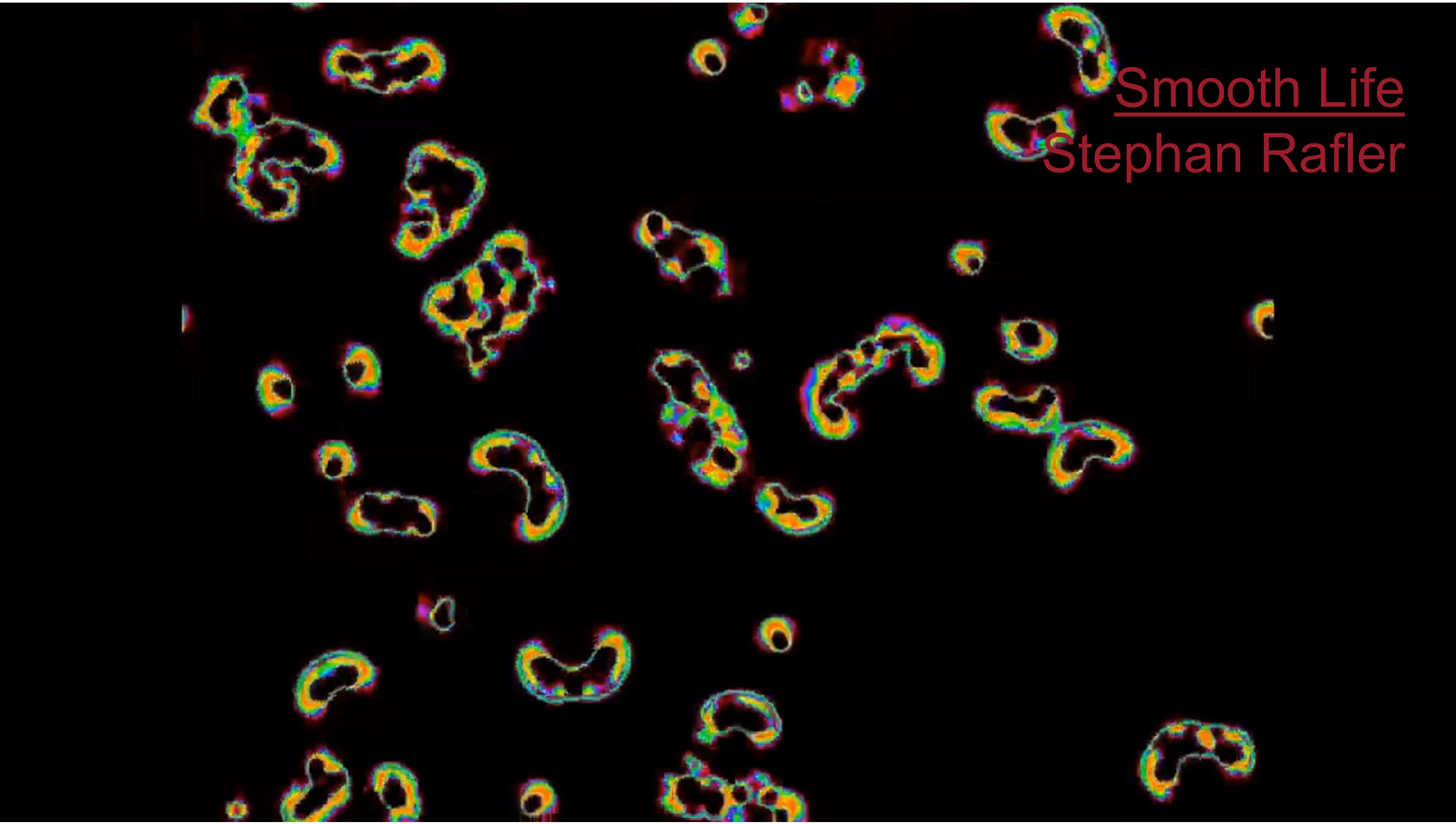


- Some CAs are boring, dead, inert, fixed;

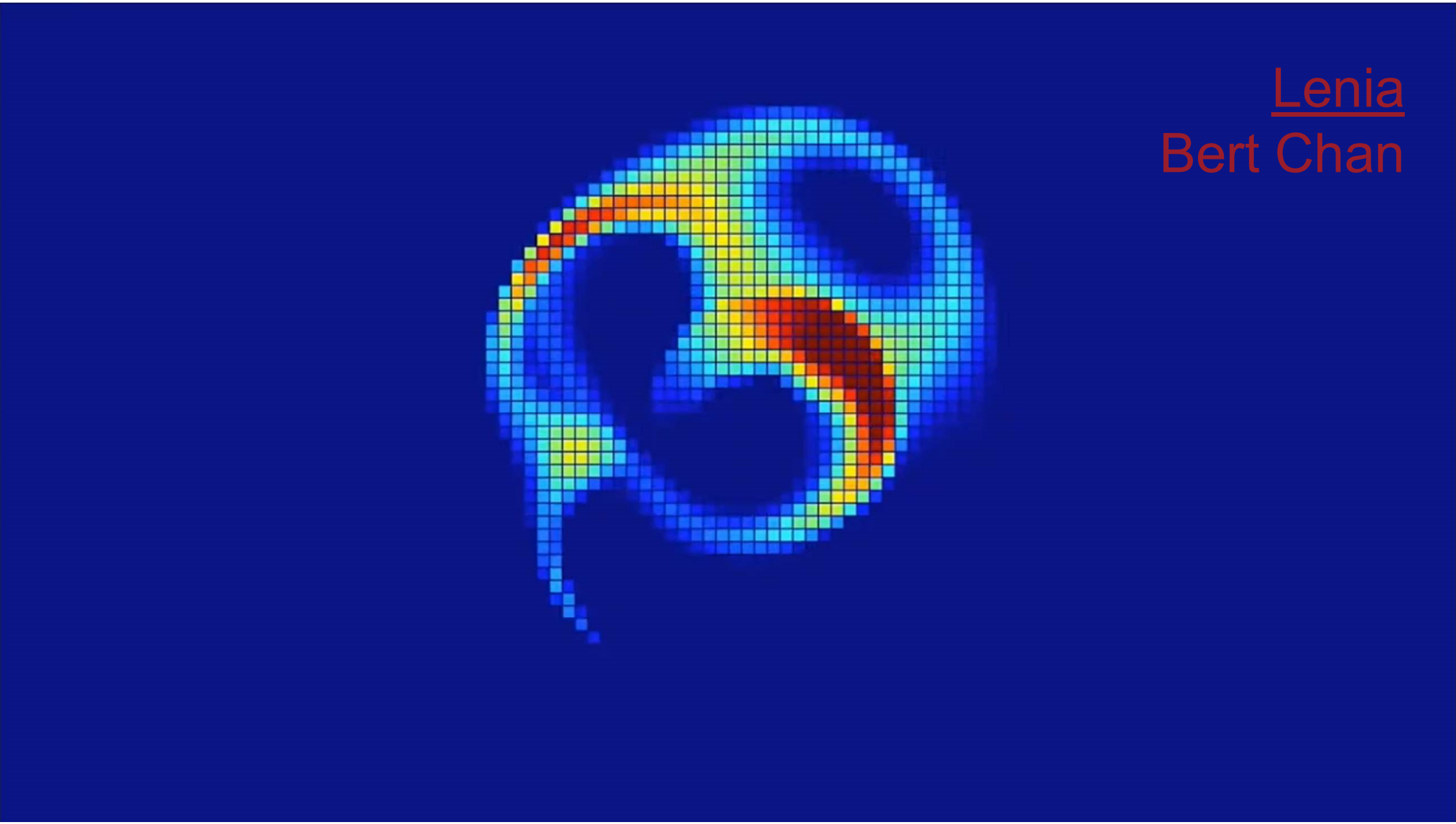
…some just repeat; some are messy chaos;

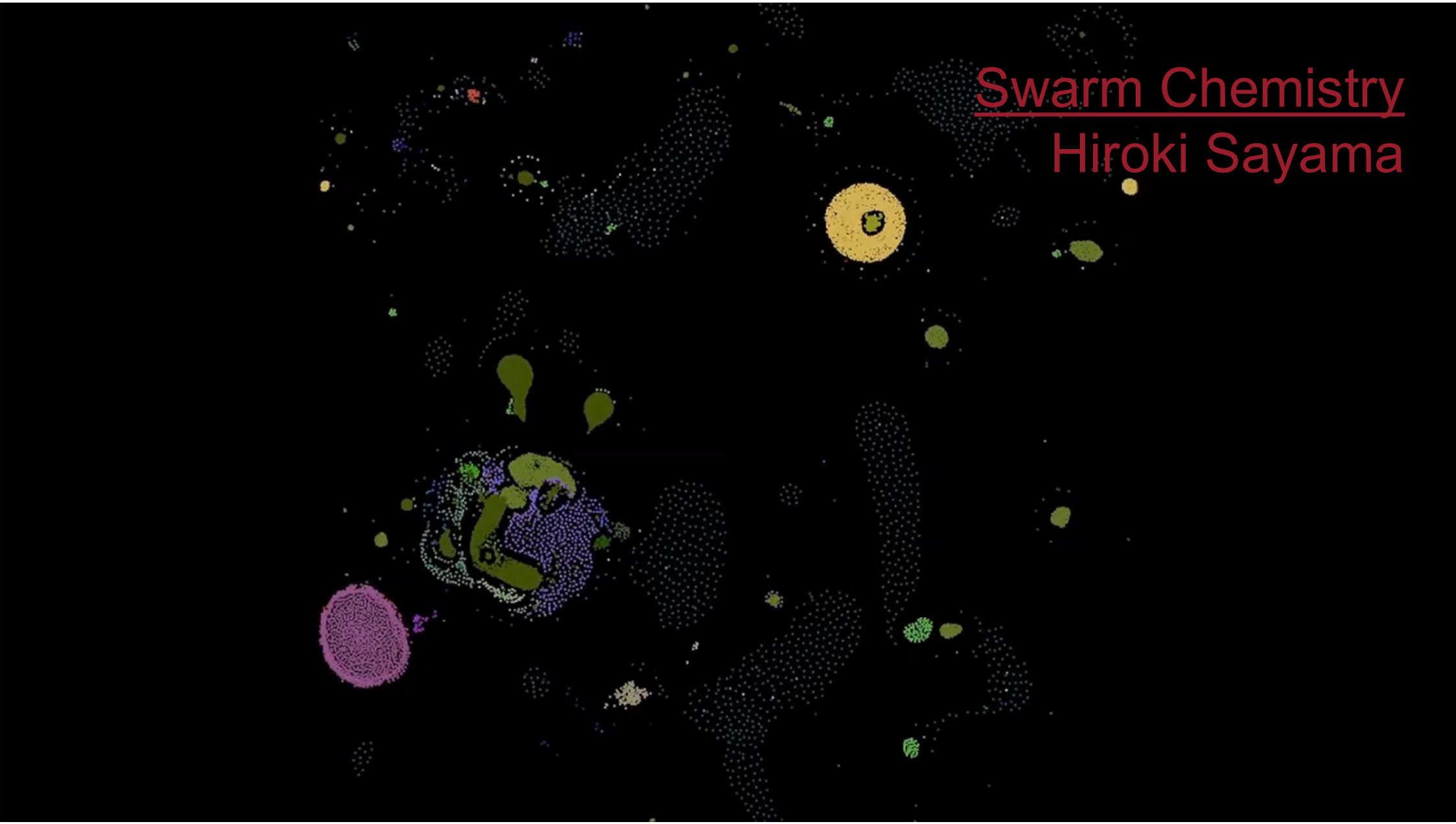
…and some are 'interesting' for a long time

- What is the physics of this complexity?
- Self-replicators + some kind of mutation = the possibility of *evolving replicators...*
- But how can truly 'open-ended evolution' be achieved in artificial systems?


Smooth Life
Stephan Rafler


Smooth LifeL
(Tim Hutton)

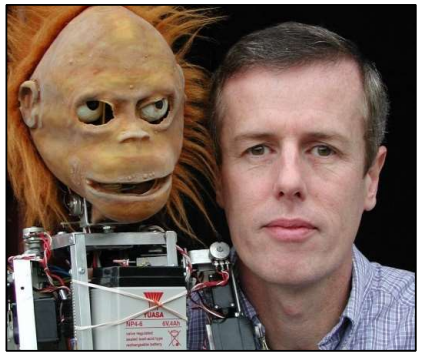

Lenia
Bert Chan

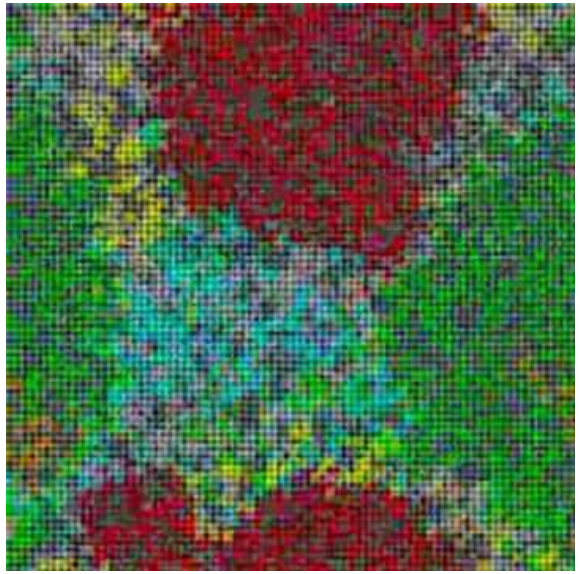

Swarm Chemistry
Hiroki Sayama

University of BRISTOL

Tierra to Avida

- Other Alifers explored the evolution of “digital organisms”:
  - Tom Ray’s *Tierra*: from replicators to parasites to hyperparasites
  - Chris Adami’s *Avida*: a kind of digital evolutionary laboratory
  - Alastair Channon’s *Geb*; Yaeger’s *Polyworld*.
  - Steve Grand’s video game: *Creatures*

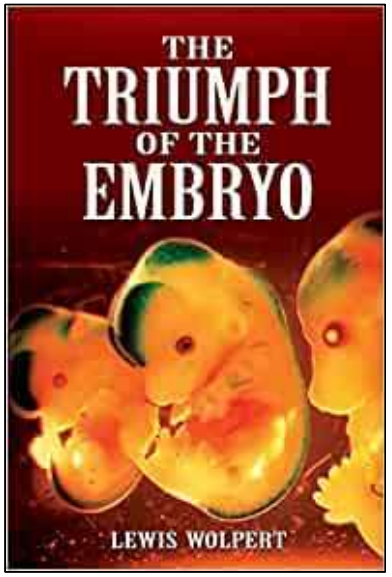




University of BRISTOL

From Evolution to Development…

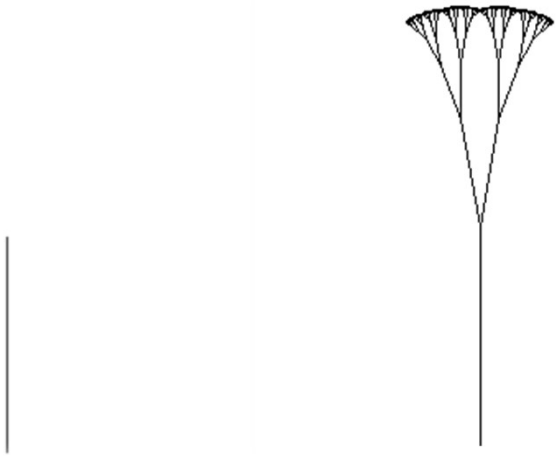
- Real biological lifeforms *develop* over time: morphogenesis
- Understanding how exactly an egg grows into a person is still one of the most profound scientific challenges
- Since before Turing: *growing* an AI has looked far more feasible than hand-crafting one
  - How can artificial systems grow and develop?
  - How can this development be steered or guided?



University of BRISTOL

L-Systems

- Lindenmayer Systems, Aristid Lindenmayer (1925-’89)
  - A model of plant development based on grammar rewriting
  - In language: S -> NP+VP; VP -> V+NP; etc., etc.
  - In L-Systems: I -> Y
  - More complex rules:

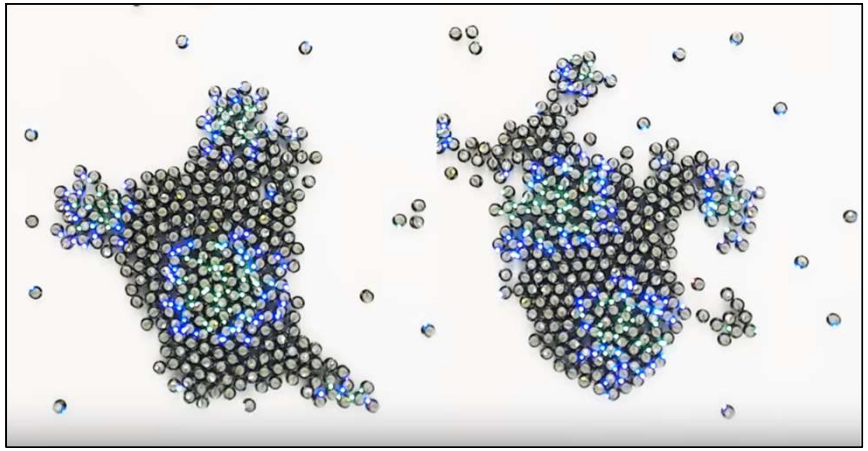


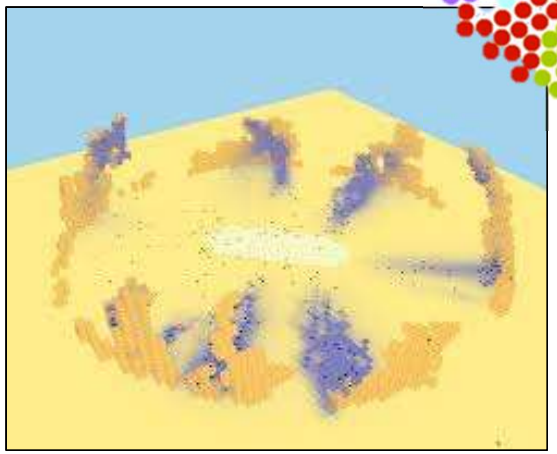
Przemysław Prusinkiewicz, A. Lindenmayer (1990). *The Algorithmic Beauty of Plants*

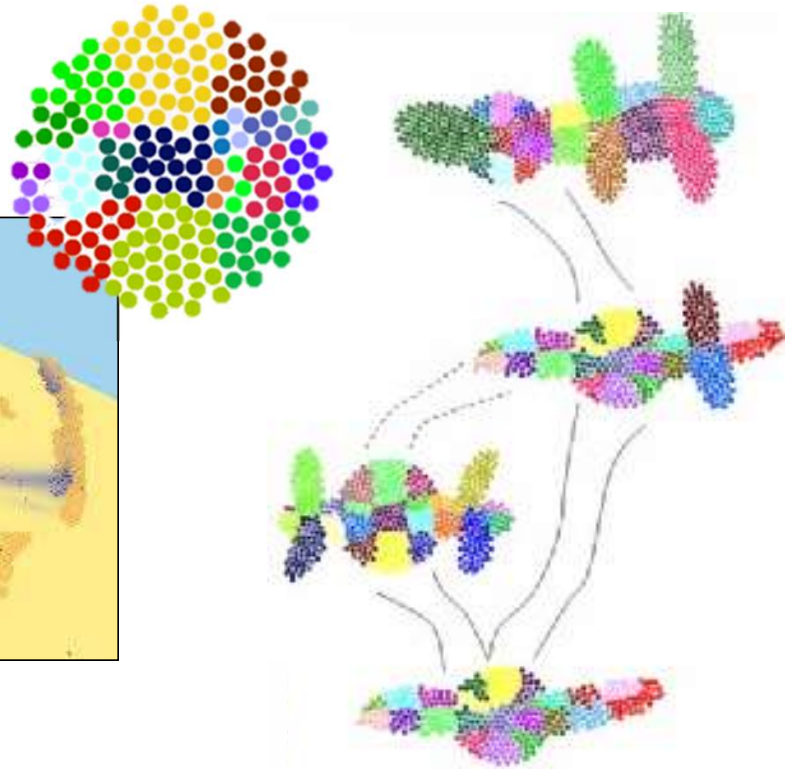
University of BRISTOL

Artificial Morphogenesis

- Morphogenesis is still a growing area of research in Alife: 😂
  - "Morphogenetic Engineering"
  - Swarm Construction

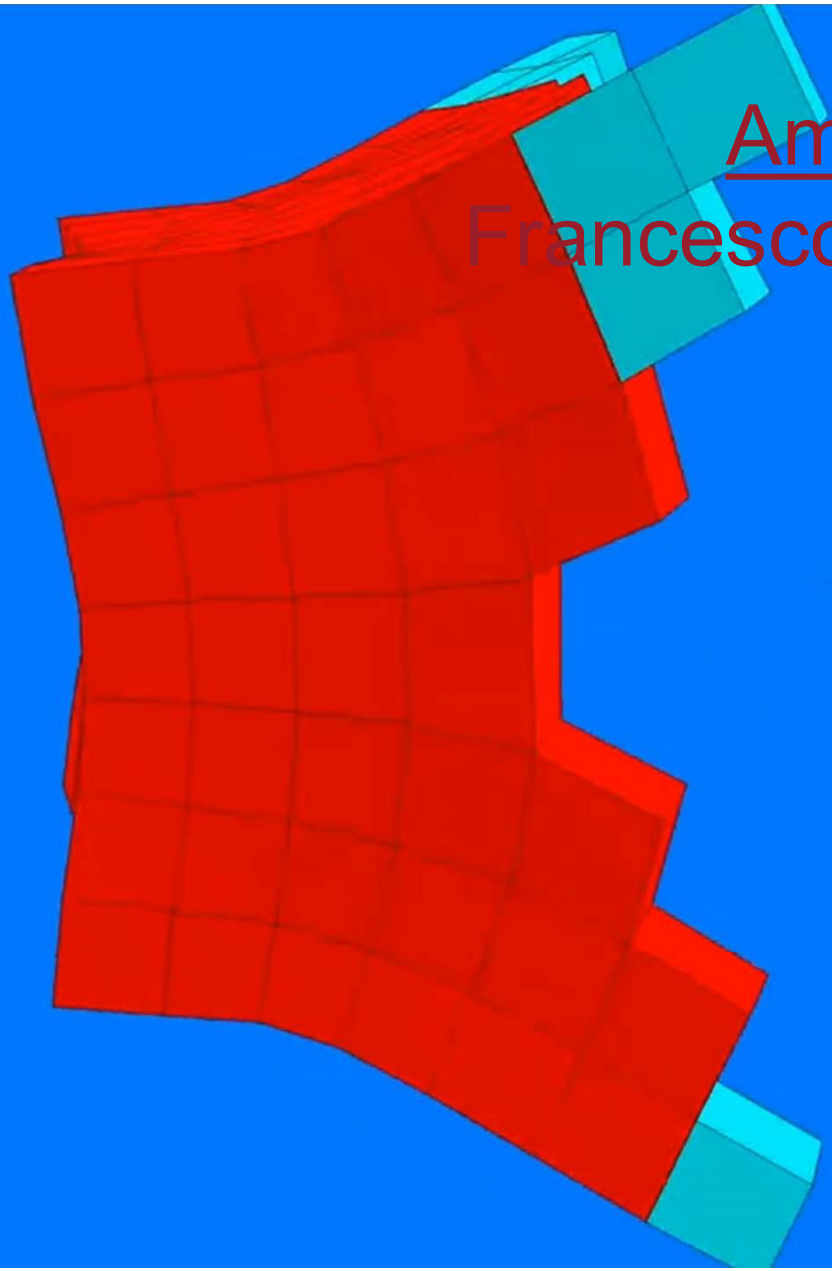






- Evo-Devo Soft Robotics

1



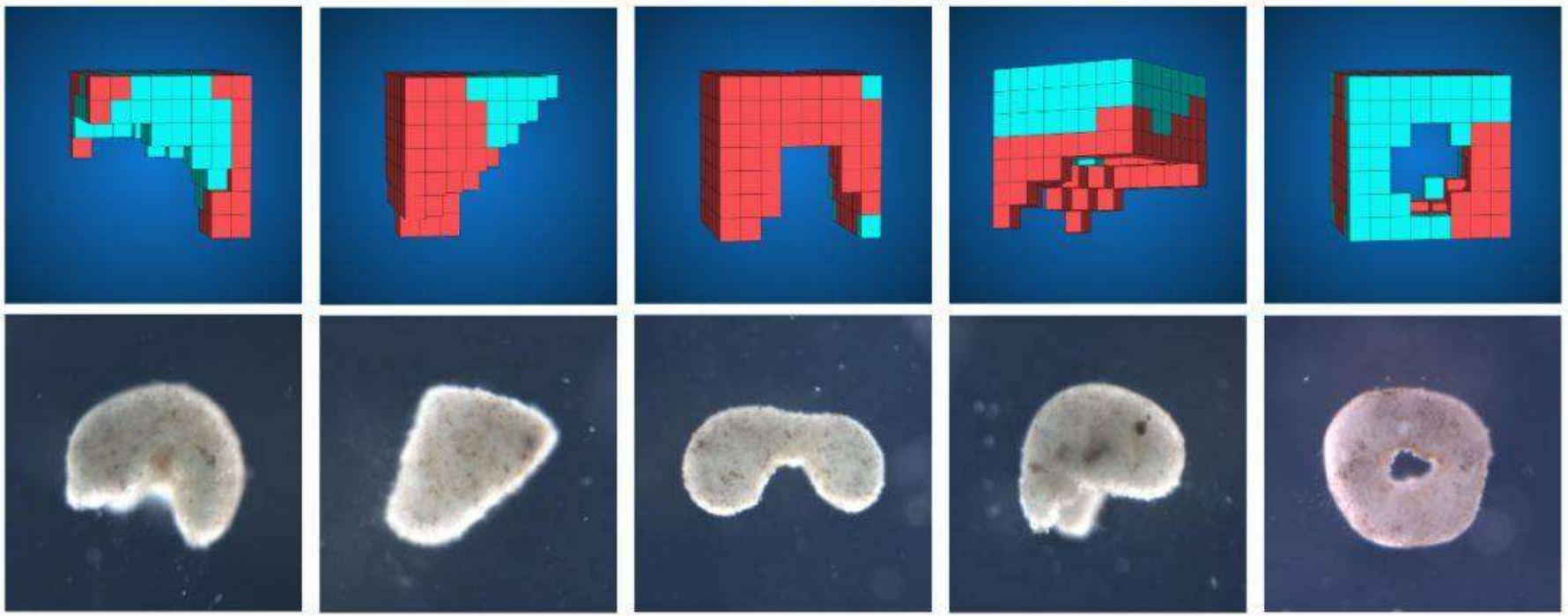
Amphibians

Francesco Corucci

University of BRISTOL

Xenobots

Robotics + soft-body simulation + evo-devo + bio-chemistry…



Josh Bongard (Vermont), Michael Levin (Tufts) and collaborators

University of BRISTOL

Types of Alife

Alife combines many disciplines

- Computer Science
- Robotics
- Biology
- Physics
- Chemistry
- Philosophy
- Linguistics

+ more

… and considers different life forms:

- Soft Alife: digital life
- Hard Alife: robotic life
- Wet Alife: biochemical life

- Whereas AI has focussed on:
  - Reasoning, Planning, Logic, etc.
- Alife has focussed on:
  - (Co-)Evolution, Development, Learning, Self-Organisation, Emergence, Origins of Life, etc.

University of BRISTOL

Open Problems

In 2000, a group of nine prominent Alife researchers authored a journal paper: "Open problems in Artificial Life": 14 challenges that are still live + relevant…

- How does life arise from the non-living? (5 problems)
  - Achieve the transition to life in an artificial chemistry.
  - Simulate a unicellular organism over its entire life cycle.
- What are the potentials and limits of living systems? (5 problems)
  - Determine what is inevitable in the open-ended evolution of life.
  - Develop a theory of information processing, flow and generation for evolving systems.
- How is life related to mind, machines, and culture? (4 problems)
  - Demonstrate emergence of intelligence in an artificial living system.
    - Establish ethical principles for artificial life.

University of BRISTOL

Women in Alife!

bristol.ac.uk

University of BRISTOL

Further Reading

- Langton, C. (1989). Artificial life, in *Artificial Life*, Addison-Wesley.
- Lehman, Clune, Misevic, Adami, et al., [52 authors!] (2020). The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *Artificial Life*, 26:2, 274-306.
- International / European Alife Conference proceedings at alife.org
- The *Artificial Life* journal at MIT Press | Lana Sinapayen's Prezi 🔥
- Hiroki Sayama's PyCX Python package and associated text book
    - NetLogo Life, Alife Virtual Seminars, Alife Papers

University of BRISTOL

Example Questions

- Which cells in the Game of Life grid shown below will survive to the next time step, and which will not? *[2 marks]*
- How did Von Neumann's insight into self-replicating machines anticipate the DNA molecule? *[4 marks]*
- Why might it be more feasible to grow a general purpose AI rather than code one by hand? *[4 marks]*
- Apply two iterations of the following L-system rules to the starting shape shown below. *[3 marks]*

University of BRISTOL

Thank you!