University of BRISTOL

# Introduction to GAs

Seth Bullock

bristol.ac.uk

# Introduction

Evolutionary Computing (EC)/Evolutionary Algorithms (EAs)

- used for black-box function optimization/design
  - e.g., find a wing design with low drag & weight and good lift
- also used for discovering software agent strategies, robot control structures, neural network designs, art, etc. …
- and sometimes used for biological modelling
- or even exploring “life-as-it-could-be” (“Artificial Life”)

# Biological Inspiration

Evolutionary algorithms are a nature-inspired/bio-inspired computing paradigm, of which there are many examples:

- Neural Networks
- Ant Colony Optimization
- Swarm Intelligence
- Artificial Chemistry
- DNA Computing
- Artificial Immune Systems

# Biological Inspiration

Nature is a great source of inspiration, but:

- All of these paradigms are simplified compared to nature
  - is something important missing?
  - has something irrelevant been included?
  - what exactly makes these paradigms work well in nature?
  - in what situations will they work well for us?

# Biological Inspiration

Nature is a great source of inspiration, but:

- Nature's solutions may not be best for us:
  - brains: highly parallel, but individual elements are very slow
  - desktop computer: mostly serial, very fast components
- Nature has only explored a subset of possible solutions:
  - e.g., what about three or four parents, rather than just two
  - e.g., what if kids inherited parents' knowledge in their genes?
    - Not possible in nature, but it is possible in a computer…

University of BRISTOL

# Biological Inspiration

Nature is a great source of inspiration, but:

- Natural systems have limits
  - Evolved creatures are amazing, but they evolved to fit a *niche*
  - They did not evolve to be general purpose or universal
  - An immune system has limits; An ant colony has limits
  - Even a brain has limits – it is not good at all types of thinking
  - Just because biology is cool, doesn't mean a bio-inspired approach is the best… Copying nature is not a magic bullet!
    - <u>Naturalistic Fallacy:</u> “You can’t get an *ought* from an *is*.”

University of BRISTOL

## Engineering View

EC is a form of stochastic (randomized) search

- We seek the highest quality solution in the least time
- Evaluating all solutions is guaranteed to find the best one
    - ...but it is slow!
- EC's defining feature: a *population* of solutions search over the (possibly infinite) space of solutions
- EC is often used when traditional search methods are not appropriate or the problem is poorly understood...

# Biological View

- Nature rewards effective behaviour with more offspring:
  - Better at catching food => more offspring ☺
  - More effective immune system => more offspring ☺
  - Better eyesight => more offspring ☺
  - Tendency to fall off cliffs => fewer offspring 😐
- "Fit behaviour": Behaviour that "fits" its environment.
- "Fitness": The tendency to generate more offspring.
  - …*in a context*, environment, task, setting…

# Biological View

Nature is a 'blind watchmaker':

- Creates complex things… …that seem *consciously designed*
- But it is just a noisy copying process that rewards success
    - 'heredity+variation+selection' ..or.. 'copying+error+constraints'

To breed dogs, we hijack the mechanisms that nature uses:

- To get faster dogs / we select the fastest doggy parents
- We do 'artificial selection' .. Nature does 'natural selection'

# Evolution in an Algorithm

We need to have a few elements:

1. A *population* of individuals, each represented by ‘genes’
2. A way to evaluate their fitness: a *fitness function*
3. A way to *select* the fitter individuals as parents
4. A way to introduce some *variety* in their offspring:
   - copying errors (‘mutation’)
   - mixing genes from two parents (‘sex’)
5. A *loop* over steps 2, 3, & 4 + a *stopping condition*

# Evolution in an Algorithm

We specify what we want evolution to find by defining a *fitness function*

- Analogous to an *objective function*, or *reward function* in Machine Learning or Reinforcement Learning
- A fitness function assigns a fitness score to one entire individual, based on its overall performance:
  - A Wing: $fitness_i = f(drag_i, weight_i, lift_i)$, e.g., $lift_i/(drag_i+weight_i)$
    - A Robot Cat: $fitness_i = 1/(1+num\_rats\_in\_my\_house)$

University of BRISTOL

# Law of Unintended Consequences

- An unwritten assumption: Defining a good fitness function is easier than working out the solution yourself by hand
- But designing the fitness function can be difficult…
- It may be tricky to exploit our knowledge of the problem:
  - E.g., assigning a fitness boost to a chess player when it captures the queen may end up rewarding poor chess-playing strategies
- Fitness functions can encourage *degenerate* solutions:
  - E.g., car fitness = “don’t crash” => a car that never moves

University of BRISTOL

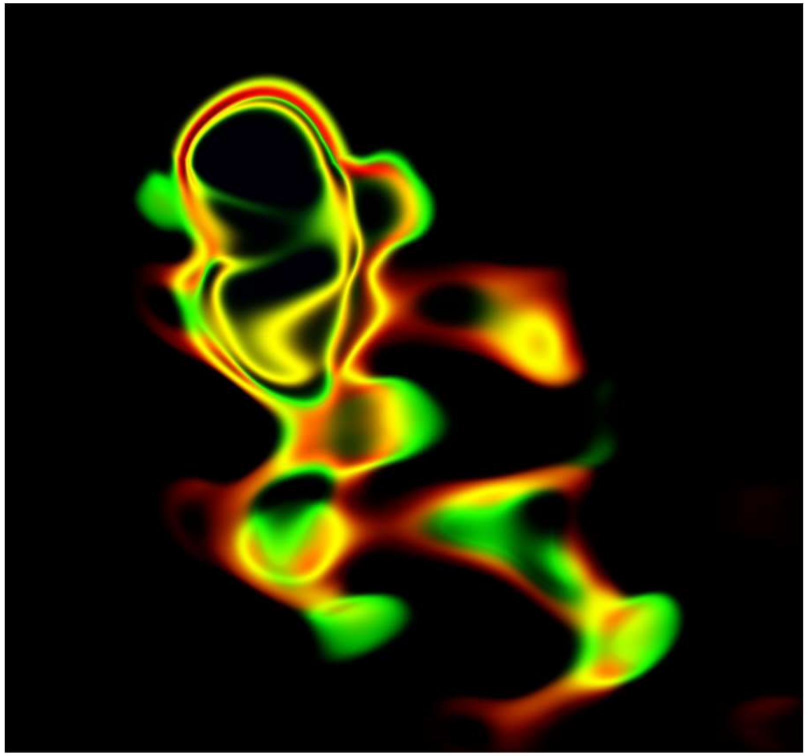
# Further Reading/Watching

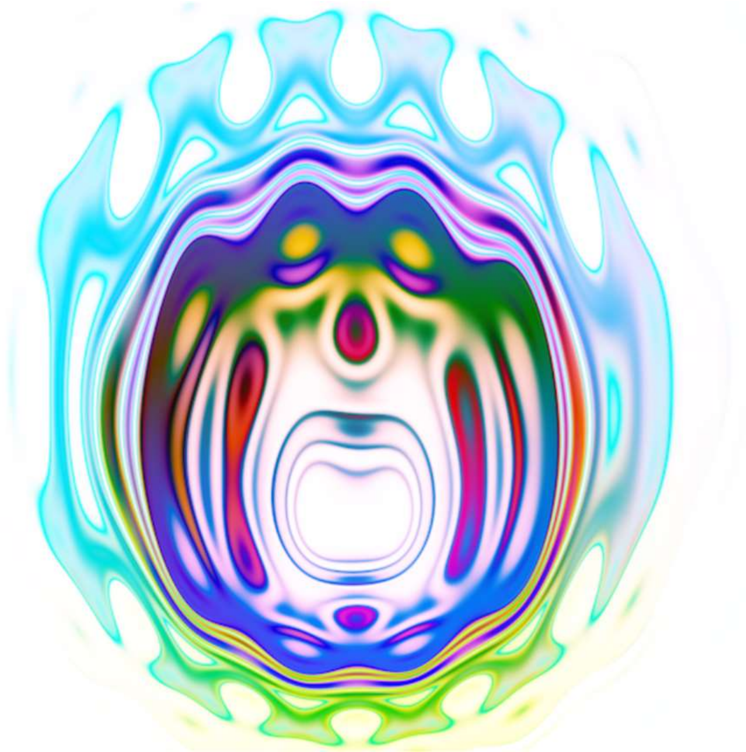
- *The Blind Watchmaker*, Richard Dawkins, Norton & Co., 1986.
- *Biomorphs*: Interactive Web App + Explanation + Simple Code https://www.defy.org/hacks/biomorphs/
- Karl Sims' *Genetic Images*: Explanation + Gallery + Papers, https://www.karlsims.com/genetic-images.html
- David Ha's *Neurogram*: https://blog.otoro.net/2015/07/31/neurogram/
- Video of Richard Dawkins' *Horizon* Episode (45 min.): http://www.dailymotion.com/video/x223a3n
  - "Methinks it is like a weasel", Biomorphs, Rechenberg, Holland..

# Karl Sims (1991)

https://www.karlsims.com/papers/siggraph91.html

# David Ha, Google Brain (2015)





https://blog.otoro.net/2015/07/31/neurogram/

# Applications: Optimization

- Some papers from last year (top page of Google Scholar):
  - *Automatically designing CNN architectures using the genetic algorithm for image classification*, Y Sun, B Xue, M Zhang, G Yen ..
  - *Solving the open vehicle routing problem with capacity and distance constraints with a biased random key genetic algorithm*, E Ruiz, V Soto-Mendoza, AER Barbosa ..
  - *Extended genetic algorithm for solving open-shop scheduling problem*, AAR Hosseinabadi, J Vahidi, B Saemi, AK Sangaiah ..
  - *Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis*, GT Reddy, MPK Reddy, K Lakshmanna ..

University of BRISTOL

# Applications: AI, ALife & Robotics

- Karl Sims (Blockies): https://youtu.be/mA8z0GndiYI
- Hod Lipson (GOLEM): https://youtu.be/qSI0HSkzG1E
- Hod Lipson (Walking Robot): https://youtu.be/iNL5-0_T1D0?t=273
- OpenAI (Hide & Seek): https://openai.com/blog/emergent-tool-use/
- Josh Bongard (Soft Robots): https://youtu.be/gXf2Chu4L9A
- F. Corucci (Swimming Robots): https://youtu.be/4ZqdvYrZ3ro
- Joachimczak et al. (Development) https://youtu.be/CXTZHHQ7ZiQ

University of BRISTOL

# Example Questions

- Which of the following are/are not required for evolution: *selection*, *sex*, *metabolism*, *reproduction* *[1 mark]*
- Name three kinds of bio-inspired algorithm . *[1 mark]*
- Was Richard Dawkins' Biomorphs system an example of natural selection or artificial selection? Explain. *[3 marks]*
- Give one reason why Dawkins' "methinks it is like a weasel" example is a *good* way to introduce evolution as an algorithm and one reason why it is *not*. *[4 marks]*

University of BRISTOL

# Thank you!