Nature Inspired Search and Optimisation
Advanced Aspects of Nature Inspired Search and Optimisation
Introduction

Shan He

School for Computational Science
University of Birmingham

January 14, 2020

Outline of Topics

Questions and answers

- Q: What is Nature Inspired Optimisation and Search?
- A: The study of computational systems inspired by nature such as natural evolution (the principle of survival of the fittest)
- Q: Why study it?
- A: Useful! It provides a **general** method for solving 'search for solutions' type of problems, such as optimisation, learning, and design.

The modules

20 credits (Advanced)

Three lab sessions per week

10 credits

Two lectures per week

Lectures times/venues

- Lecturers
  - Shan He: Weeks 1-5 (Email: s.he@cs.bham.ac.uk, Office hour: Thursday 17:00pm)
  - Per Kristian: Weeks 6-10
  - Both: Week 11 (revision)
- Lectures time and location
  - Tuesday 11:00am in Aston Web Main LT
  - Thursday 17:00pm in Aston Web Main LT
- Advanced version: Lab sessions time and location (Weeks 1-5)
  - Monday 11:00pm SoCS UG04
  - Help sessions: Monday 17:00 and Tuesday 9:00 SoCS UG04

Aims of the module

We will learn:

- The main concepts, techniques and applications in the field of randomised search heuristics and nature-inspired computing with a focus on optimisation.
- When such techniques are useful and how to use them in practice
- Advanced version (20 credits): How to implement them with different programming languages.

Learning outcomes

- All versions:
  - Describe different nature-inspired search and optimisation methods and explain how they are applied to solve real world problems
  - Discuss relations, similarities and differences between the most important heuristics and nature-inspired algorithms presented in the module and other search and optimisation techniques
  - Design and adapt nature-inspired algorithms including operators, representations, fitness functions and potential hybridisations for non-trivial problems
- Advanced version (20 credits):
  - Implement nature-inspired algorithms using different programming languages and compare them experimentally

Exam and Continuous Assessment

20 credits (Advanced)

- 60%: Two CAs and class tests (quizzes)
- 40%:

10 credits

- 1.5 hours exam (90%)
- Class tests (10%)

What will be covered by the two lectures?

- Problems:
  - Optimisation problems: continuous and combinatorial
  - Machine learning problems: classification and regression
- Algorithms
  - Randomised algorithm and Randomized Search Heuristics
  - Evolutionary Algorithms: Genetic Algorithms, Genetic Programming, Evolutionary Programming and Evolution Strategies
  - Swarm Algorithms: Particle Swarm Optimiser
  - Multi-model and Multi-objective optimisation
- Theory
  - Schema Theorem
  - Convergence and Convergence Rate
  - Computational Complexity
  - No Free Lunch Theorem
  - Fitness Landscape

How about the two lab sessions?

- We will focus on implementing the algorithms we learn to solve real-world problems, e.g., travelling salesman, vehicle routing, and crew scheduling problems.
- I will present and explain some **code examples**
- You can try those code examples and complete **some exercises** to solve the problems.

problem formulation

Cool example 1: NASA's Evolved antenna

- Satellite antenna evolved fully designed by an evolutionary algorithm
- Used for a 2006 NASA mission called Space Technology 5 (ST5)
  - The mission consists of three satellites that will take measurements in Earth's magnetosphere.
  - Each satellite has two communication antennas to talk to ground stations.
  - Unusual design requirements, e.g., unusual radiation patterns,
  - The mission successfully launched on March 22, 2006
- It was the first artificially-evolved object to fly in space
- Youtube video

Cool example 2: Evolutionary robotics

- Evolutionary robotics: a methodology that uses evolutionary computation to design, develop and control autonomous robots.
- One of the famous example: The Golem Project
  - Genetically Organized Lifelike Electro Mechanics
  - H. Lipson and J. B. Pollack (2000), "Automatic design and Manufacture of Robotic Lifeforms", Nature 406, pp. 974-978
  - Evolutionary Computation + 3D printing = Evolutionary Robots
- Youtube video
- Youtube video of new evolving soft robots

Cool example 3: Blondie24 - Evolved master checkers player

- Blondie24: an artificial intelligence checkers-playing computer program – an early predecessor of DeepMind's AlphaGo
- The most special feature: learn playing checkers by itself
- How: given the checkers rules, co-evolve artificial neural networks as opponent players, the wining players will be selected and breed the next generation
- Did it work? Rated as Expert on Microsoft's Gaming Zone as claimed by the authors.
- Video

Cool example 4: Automated Invention Machine

- An ambitious goal: to automate creativity
- How: using evolutionary computation, especially Genetic Programming
- A case study: invent/re-invent electronic circuits by Genetic Programming
- Reinvented 21 Previously Patented Inventions
- Invented 2 new patentable inventions
- John Koza's website

Cool example 5: Evolutionary art

- William Latham - Mutator 1 + 2 : Evolutionary Art
- Music evolved by Evolutionary Computation

A few new developments: Sentient

- In 2014, a company called Sentient was set up to commercialise evolutionary computation, e.g., their evolutionary AI SaaS platform called LEAF (Learning and Evolutionary Algorithm Framework)
- Sentient was acquired by Evolv Technology in March 2019:
  - Evolv Technologies Raises $10M Series A to Scale Ascend Evolutionary AI-powered Optimization Platform
- One video produced by Sentient is not longer available in Evolv.ai, but here at prnewswire
- Some videos and articles from Sentient are now available at Cognizant, a NASDAQ 100 company: https://www.cognizant.com/ai/evolutionary-ai

A few new developments: Sentient

- In 2016, UBER launched his AI lab by acquiring Geometric Intelligence, a start-up specialising in evolutionary neural networks (Link)
- In 2018, Facebook AI lab released an algorithm library Nevergrad, which implemented many nature inspired optimisation algorithms