CSCU9YE - Artificial Intelligence

Lecture 6: Evolutionary Algorithms

Prof. Gabriela Ochoa, University of Stirling

Outline

Optimisation problems

- The travelling salesman problem
- Vehicle routing
- Other ‘fun’ and practical examples

Optimisation methods

- Constructive Heuristics,
- Single point algorithms
- **Population-based algorithms**

Combinatorial explosion...

Consider that Tom would be traveling through Europe and wishes to visit all 28 EU capitals.

In this case, there would be 5 billion, billion, billion possible tours...

$\underbrace{5,444,434,725,}_{\text{billion}}\underbrace{209,176,080,}_{\text{billion}}\underbrace{384,000,000}_{\text{billion}}$ possible tours...

???

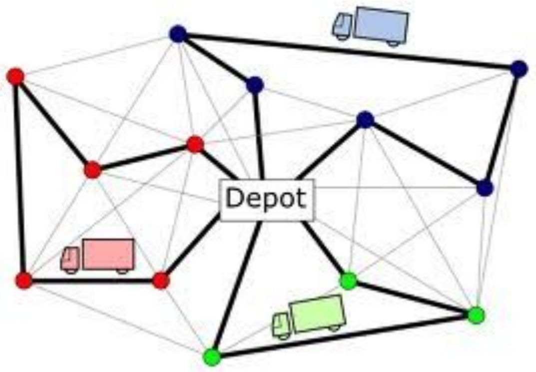
```
                                                     Factorial
                                               1 =  1!
                                               2 =  2!
                                               6 =  3!
                                              24 =  4!
                                             120 =  5!   8-bit 255
                                             720 =  6!
                                            5040 =  7!
                                           40320 =  8!  16-bit 65535
                                         3 62880 =  9!
                                        36 28800 = 10!
                                       399 16800 = 11!
                                      4790 01600 = 12!  32-bit 42949 67295
                                     62270 20800 = 13!
                                   8 71782 91200 = 14!
                                 130 76743 68000 = 15!
                                2092 27898 88000 = 16!
                               35568 74280 96000 = 17!
                             6 40237 37057 28000 = 18!
                           121 64510 04088 32000 = 19!
                          2432 90200 81766 40000 = 20!  64-bit 18446 74407 37095 51615
                         51090 94217 17094 40000 = 21!
                      11 24000 72777 76076 80000 = 22!
                     258 52016 73888 49766 40000 = 23!
                    6204 48401 73323 94393 60000 = 24!
                  1 55112 10043 33098 59840 00000 = 25!
                 40 32914 61126 60563 55840 00000 = 26!
               1088 88694 50418 35216 07680 00000 = 27!
              30488 83446 11713 86050 15040 00000 = 28!
            8 84176 19937 39701 95454 36160 00000 = 29!
          265 25285 98121 91058 63630 84800 00000 = 30!
         8222 83865 41779 22817 72556 28800 00000 = 31!
       2 63130 83693 36935 30167 21801 21600 00000 = 32!
      86 83317 61881 18864 95518 19440 12800 00000 = 33!
    2952 32799 03960 41408 47618 60964 35200 00000 = 34! 128-bit 3402 82366 92093 84634 63374 60743 17682 11455
  1 03331 47966 38614 49296 66651 33752 32000 00000 = 35!
```

The combinatorial explosion

The growth of the Factorial Function

Travelling Salesman size of the search space:
$(n-1)!/2$

The vehicle routing problem



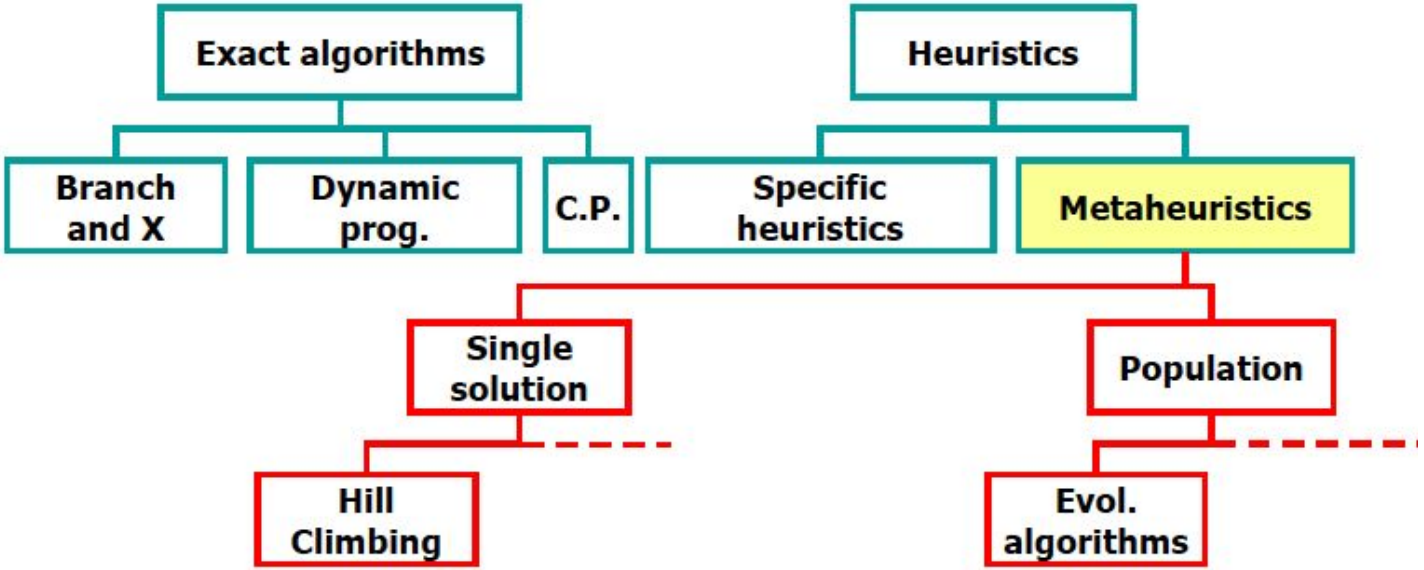
Given: A set of customers, a set of vehicles and a central depot

Goal: Design minimum cost routes visiting all customers

Representation: A set of routes one for each truck

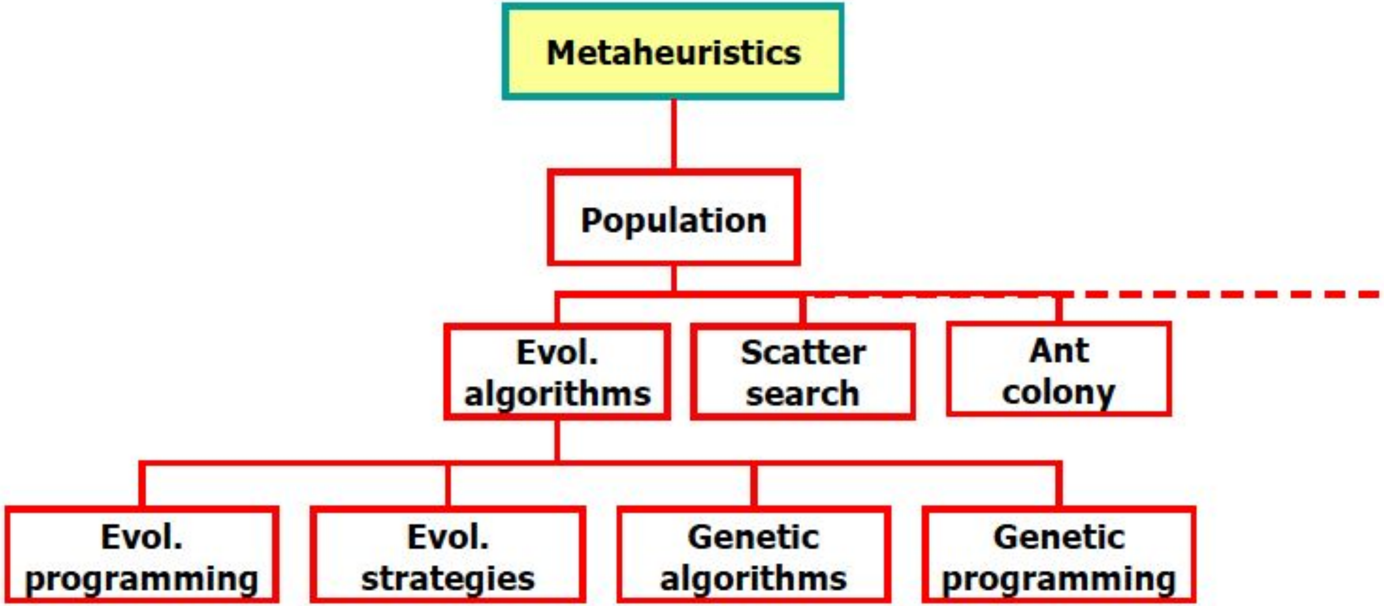
Objective: Reduce the number of trucks used and the total distance travelled

Classification of optimisation algorithms



- Single solution algorithms are exploitation oriented
- Population-based algorithms are exploration oriented

Classification of metaheuristics

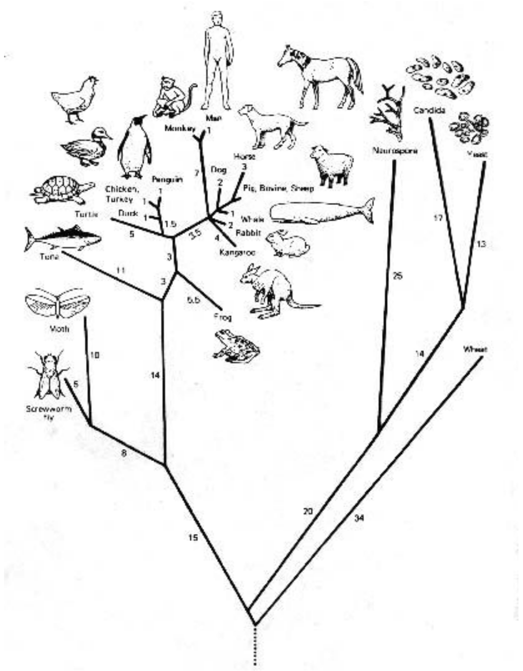


What is an evolutionary algorithm?

- EAs fall into the category metaheuristic algorithms
- They are stochastic, population-based algorithms
- Inspired by the process of Evolution by Natural Selection
- Variation operators (recombination and mutation) create the necessary diversity and thereby facilitate novelty
- Selection reduces diversity and acts as a force pushing quality

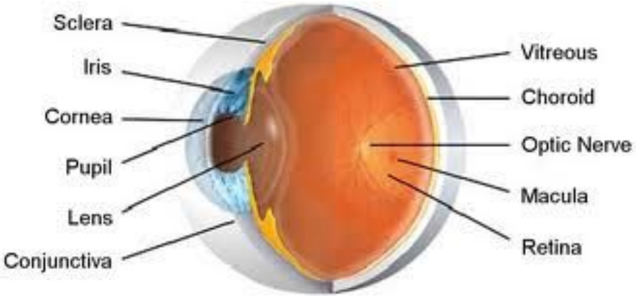
Natural evolution: fact and theory

- Changes across successive generations in the heritable characteristics of biological populations
- Theory of Evolution: One of the great intellectual revolutions of human history
- Life on Earth evolved from a universal common ancestor approximately 3.8 billion years ago

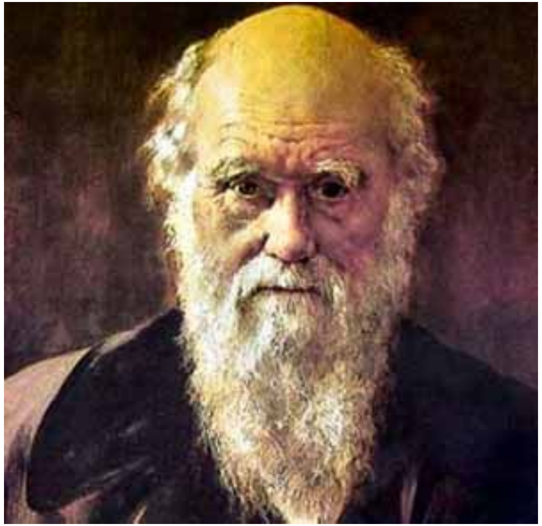


The Great Tree of Life

Examples of *Apparent Design* in Nature



Evolution by natural selection



Natural Selection

1. Variation
2. Hereditary transmission
3. High rate of population growth
4. Differential survival and reproduction

Charles Darwin & Alfred Wallace: Theory of evolution by means of Natural Selection.

1859: *On the Origin of Species by Means of Natural Selection: Or, The Preservation of Favoured Races in the Struggle for Life*

Evolutionary Computation

Branch of *Computational Intelligence* that study methods that mimic evolution by natural selection, with the aim of solving complex design and optimisation problems

EVOLUTION		PROBLEM SOLVING
Environment	⟷	Problem
Individual	⟷	Candidate Solution
Fitness	⟷	Quality

Fitness → chances for survival and reproduction
Quality → chance for seeding new solutions

Outline of an Evolutionary Algorithm

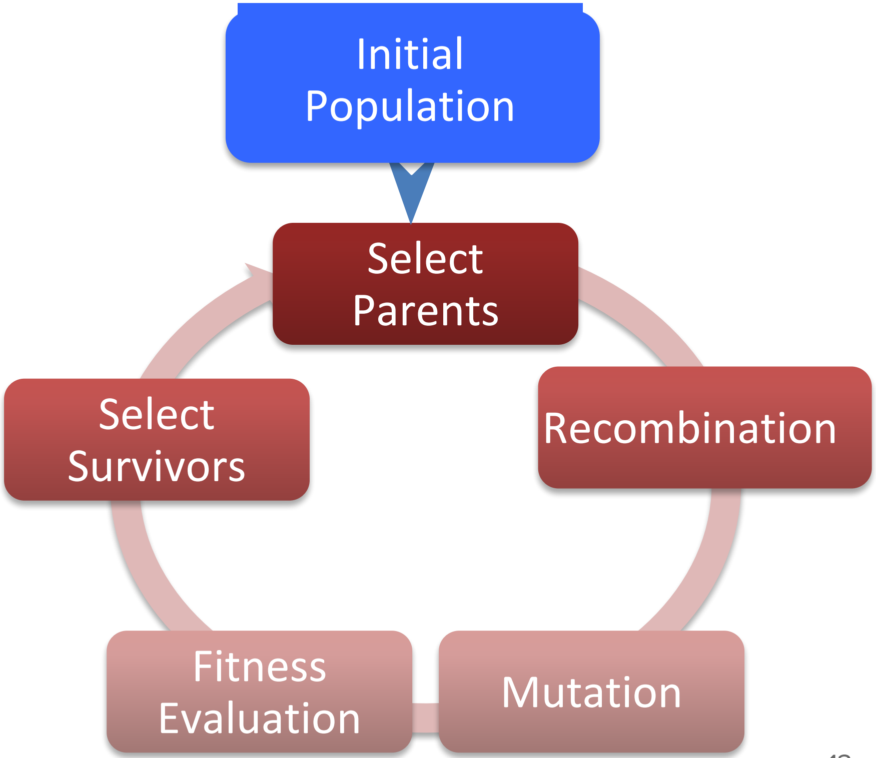
Generate [P(0)]
t = 0
WHILE NOT *Termination_Criterion* [P(*t*)] **DO**

    Evaluate [P(*t*)]
    P' (*t*) = *Select* [P(*t*)]
    P''(*t*) =*Apply_Variation_Operators* [P'(*t*)]
    P(*t*+1) = *Replace* [P(*t*), P''(*t*)]
    t = *t* + 1

END
RETURN *Best_Solution*

Representation: Genetic material

Fitness Function: Task to perform



Origins of Evolutionary Algorithms

Evolution Strategy (Germany)

Ingo Rechenberg & Hans-Paul Schwefel (1960s and 1970s)

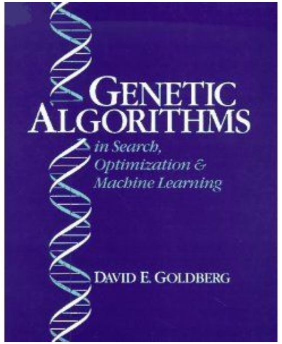
Genetic Algorithms (US)

John Holland (1975)

David Goldberg (1989)

Alan Turing. Mathematician, wartime code-breaker and pioneer of computer science. Article: "Computing Machinery and Intelligence," (1950)

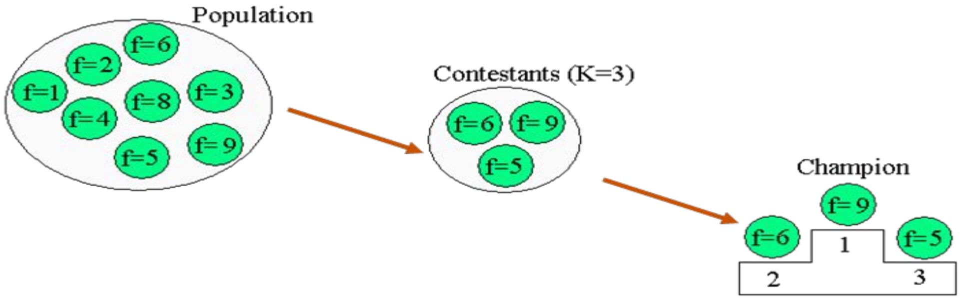
Google Scholar: 88,000 citations



Evolutionary (Genetic) Algorithms

Parent selection: Better individuals get higher chance (proportional to fitness).

- Proportional selection (roulette wheel, stochastic universal sampling)
- Scaling methods
- Rank selection
- Tournament selection
- ($\mu + \lambda$)- and (μ , λ) selection



Generate [P(0)]
t = 0
WHILE NOT *Termination_Criterion* [P(*t*)] **DO**

Evaluate [P(*t*)]
P' (*t*) = *Select* [P(*t*)]
P''(*t*) =*Apply_Variation_Operators* [P'(*t*)]
P(*t*+1) = *Replace* [P(*t*), P''(*t*)]
t = *t* + 1

END
RETURN *Best_Solution*

Tournament selection

Evolutionary (Genetic) Algorithms

Replacement (population models)

- **Generational:** each generation set of parents replaced by the offspring
- **Steady-state**: one offspring is generated per generation. One member is replaced
- **Generation gap**: a proportion of the population is replaced

Generate [P(0)]
t = 0
WHILE NOT *Termination_Criterion* [P(*t*)] **DO**

    Evaluate [P(*t*)]
    P' (*t*) = *Select* [P(*t*)]
    P''(*t*) =*Apply_Variation_Operators* [P'(*t*)]
    P(*t*+1) = *Replace* [P(*t*), P''(*t*)]
    t = *t* + 1

END
RETURN *Best_Solution*

Search operators for binary representation

Mutation

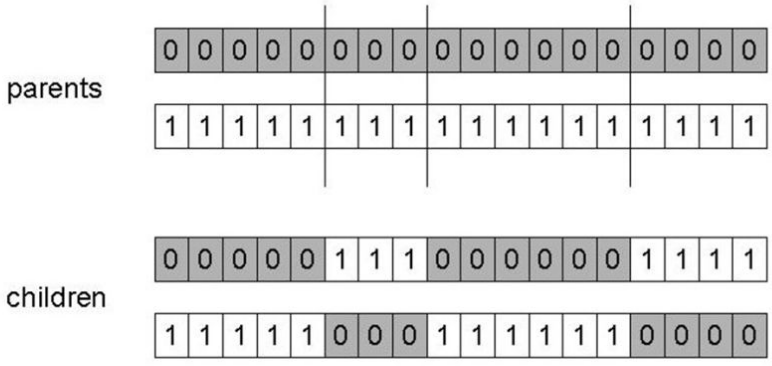
- Alter each gene independently with a probability P_m (mutation rate)
- Typically: 1/chromosome_length

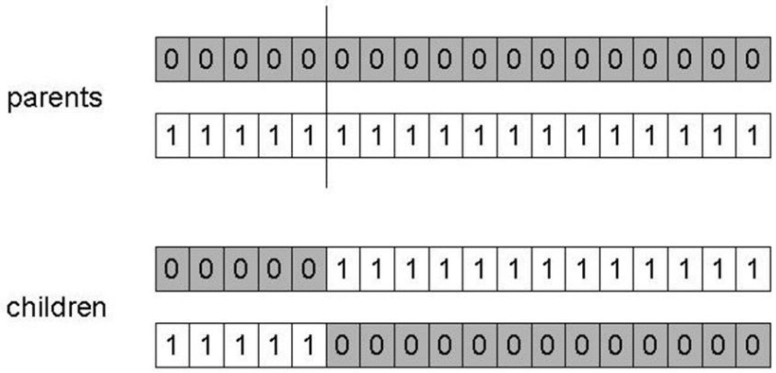
parent	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
child	0	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	0	1

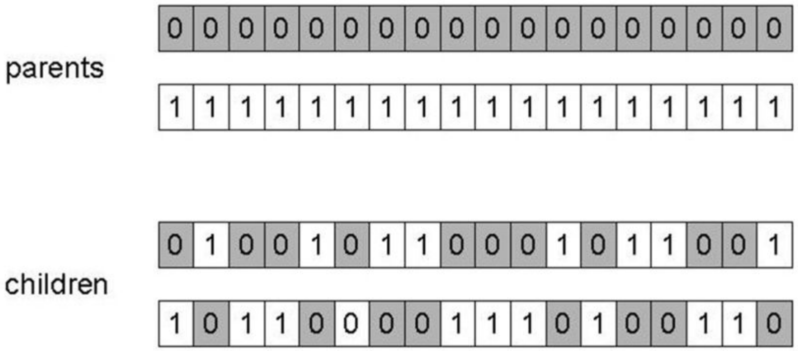
Search operators for binary representation

Recombination/crossover

- One-point
- N-point
- Uniform
- P_c typically in range (0.6, 0.9)







Search operators for permutation representation

Mutation

- 2-swap: Solutions generated by swapping two cities from a given tour
- Every solution has n(n-1)/2 neighbours
- Examples:

Move/Mutation Operators for the TSP

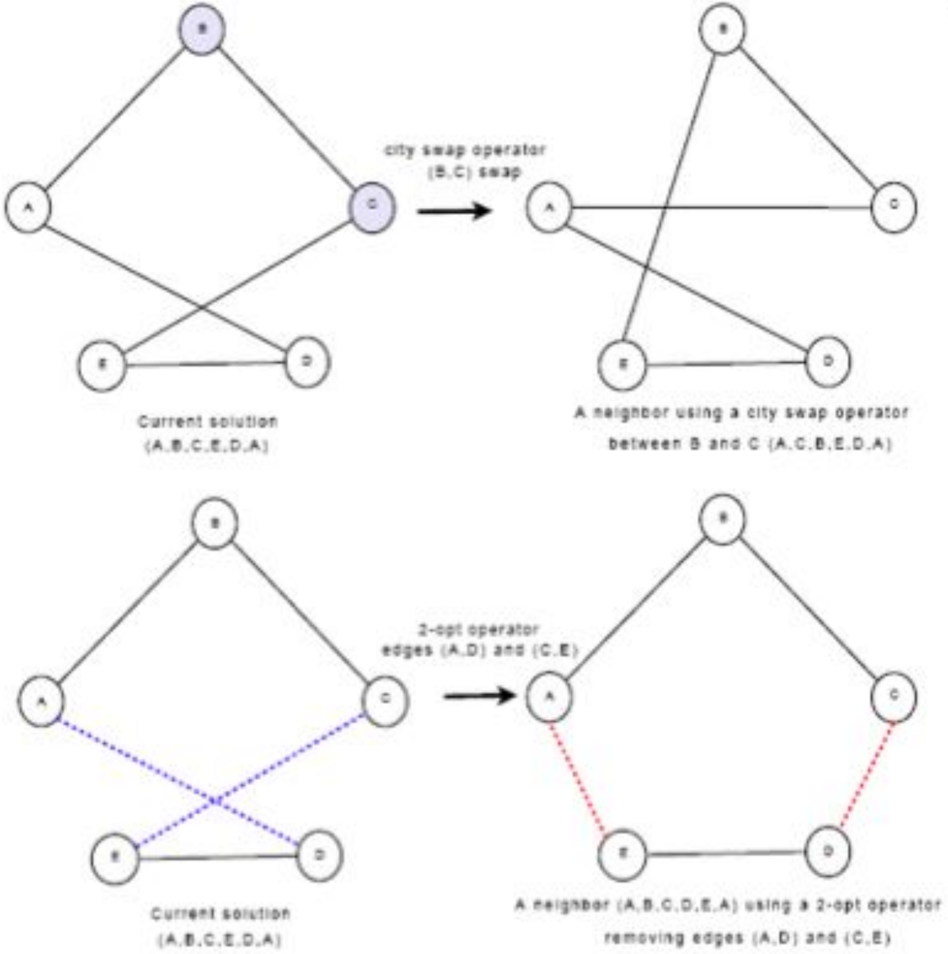


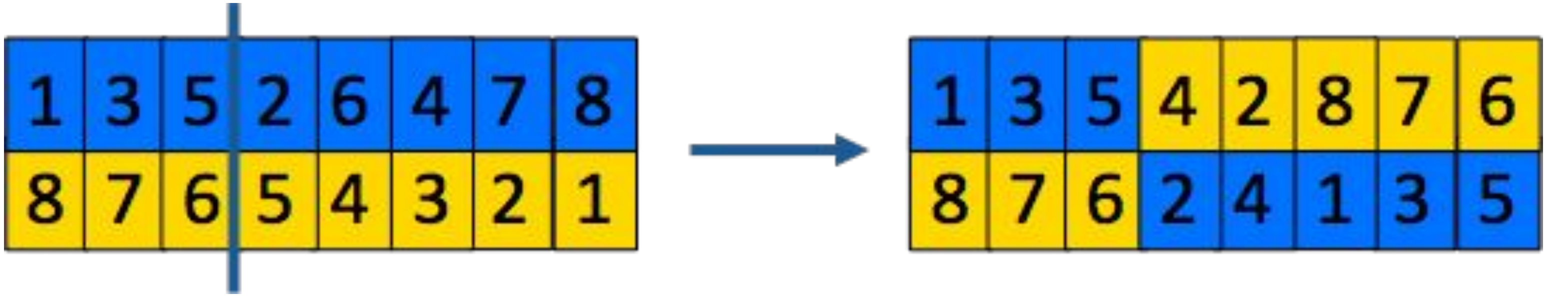
Fig. 2.5 City swap operator and 2-opt operator for the TSP.

Traveling Salesman Problem Visualization

Search operators for permutation representation

Example recombination: Combining two permutations into two new permutations

- choose random crossover point
- copy first parts into children
- create second part by inserting values from other parent:
  - in the order they appear there
  - beginning after crossover point
  - skipping values already in child

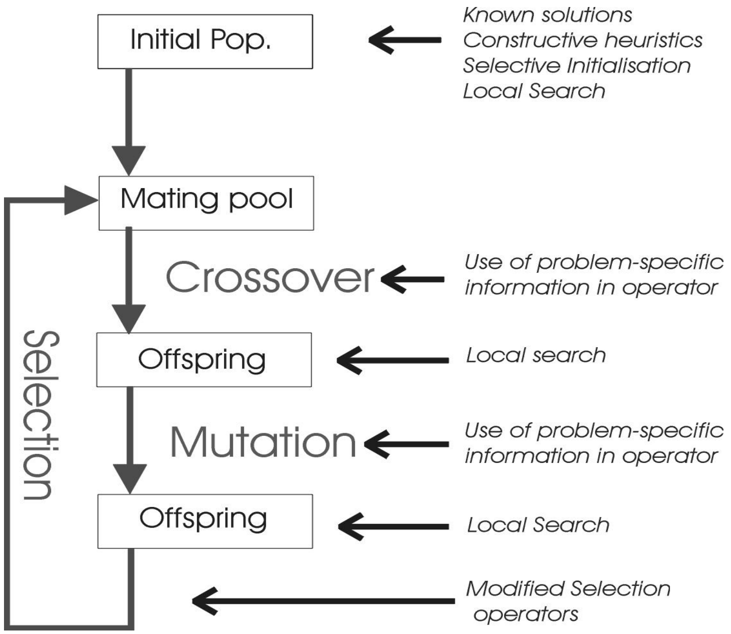


Memetic (hybrid) algorithms

Combination of GAs with local search operators, or GAs that use instance specific knowledge in operators

Could be faster and more accurate than GAs, and are the “state-of-the-art” on many problems

- The term *meme* was coined by R. Dawkins (1976)
- The term memetic algorithms by P. Moscato (1989)
- The idea of hybridisation in GAs is older



(Eiben, Smith, 2003)

Other evolutionary algorithms

Evolution Strategies

- Specialised in continuous search spaces: min. $f : R^n \to R$
- Rechenberg & Schwefel in the 60s, Technical University of Berlin. Applied to hydrodynamic shape optimisation
- Special feature: self-adaptation of mutation parameters

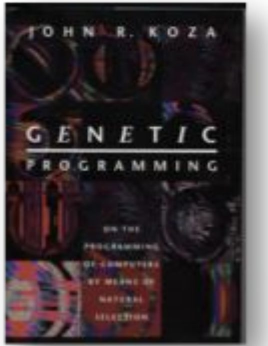
Genetic Programming

- Evolve a population of computer programs
- Applied to: machine learning tasks (prediction, classification…)
- Representation
  - Non-linear genomes: trees, graphs
  - Linear genomes: grammatical evolution

Human competitive results of Evolutionary Computation

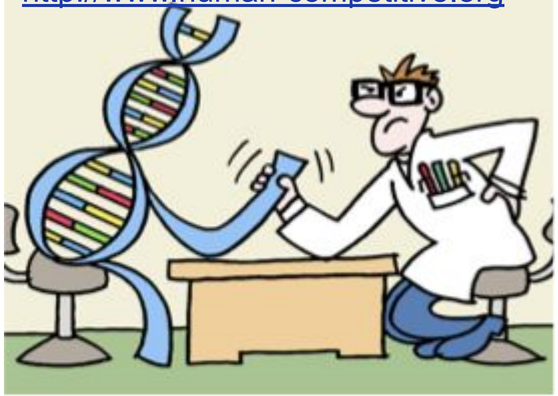
Awards program (2004 – to date) HUMIES ($10,000.00) each year

Real-world applications of EC

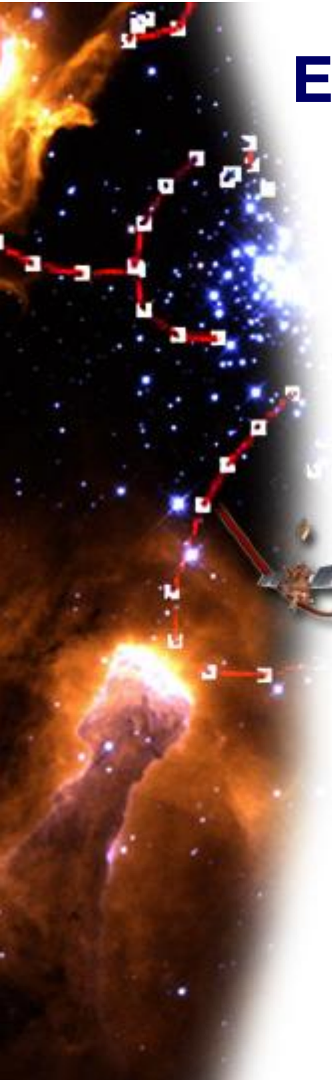


John R. Koza
Scientist and businessman. Popularised Genetic Programming, millionaire, co-inventor of rub-off instant lottery game ticket

http://www.human-competitive.org



Quantum computing circuits, analogue electrical circuits, antennas, mechanical systems, controllers, game playing, , image recognition, optical lens systems, bioinformatics, robotics, scheduling, software repair, communication protocols, ..., etc.



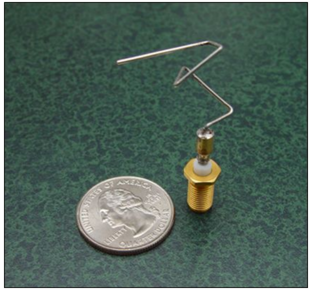
Evolved Antennas for Deployment on NASA's Space Technology 5 Mission

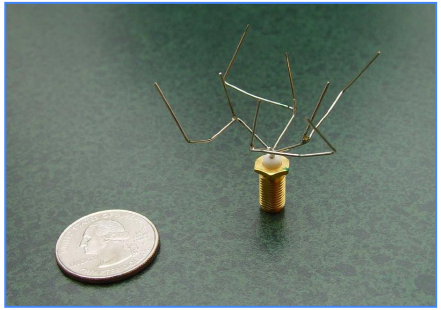
J. D. Lohn, G. S. Hornby, D. S. Linden, Evolvable Systems Group, Computational Sciences Division

GECCO-2004

- Winner of the 2004 Hummies award
- Evolved antenna that is now flying space
- One of the top evolvable hardware results to date

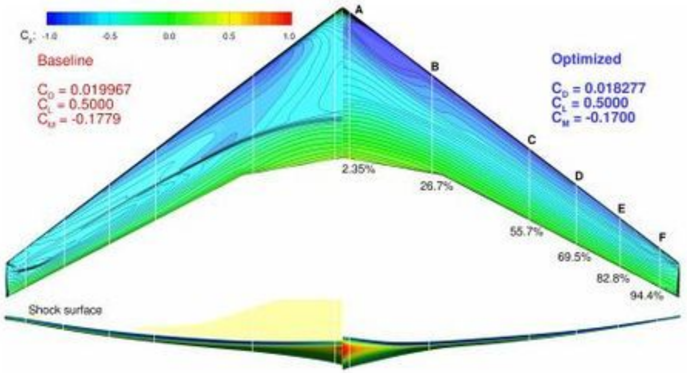
Previous human designed antenna (helical). Did not meet the requirements.





Evolved Antennas: Better than conventional design.

Aerodynamic Shape Optimisation



Wing and Airfoil Optimisation

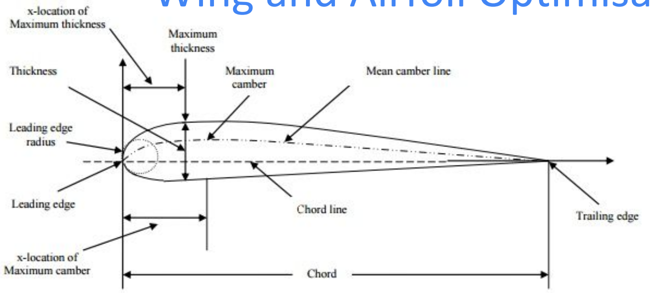


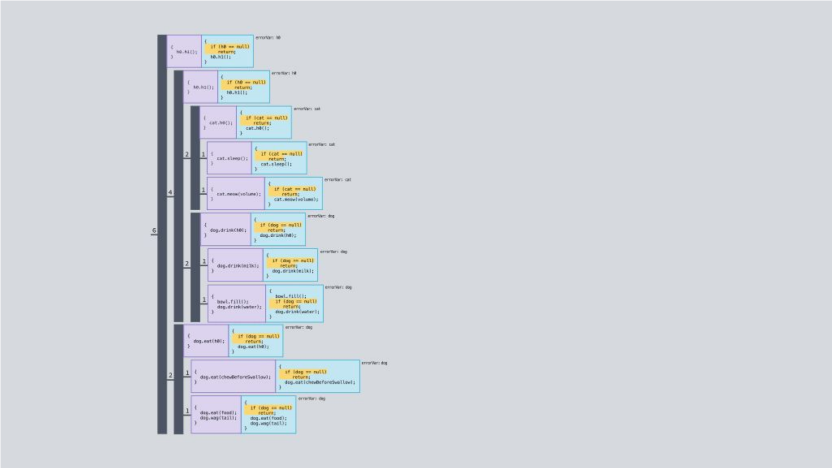
Figure 5.5. Airfoil geometric parameters

Japanese bullet trains (Shinkansen) Nose Shape Optimisation

Evolutionary Software Testing at Facebook

Sapienz: Automated software testing at scale

Getafix: Fixing bugs automatically



Engineering manager Mark Harman wins 2019 IEEE Harlan D. Mills award

Simulated Evolution of Virtual Block Creatures

Video

Virtual creatures (Karl Sims, 1994)

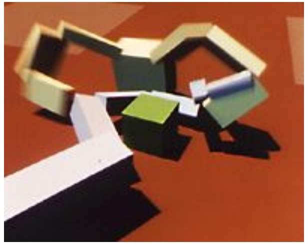
Representation: coded instructions for their growth and locomotion

Fitness Function: ability to perform a given task

Swimming

Hopping

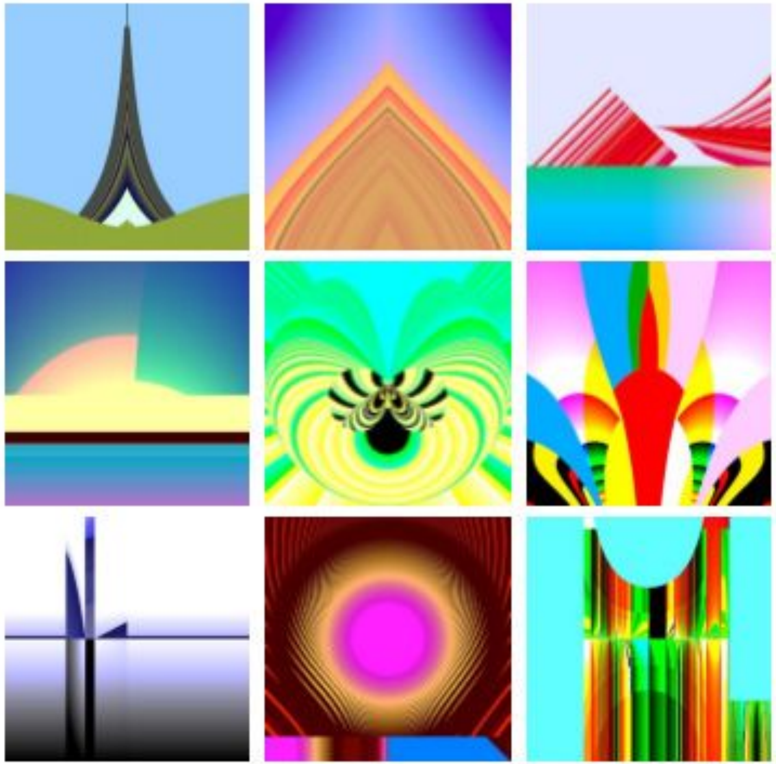
Following



Competing

Karl Sims - Evolved Virtual Creatures, Evolution Simulation, 1994

Evolutionary Art



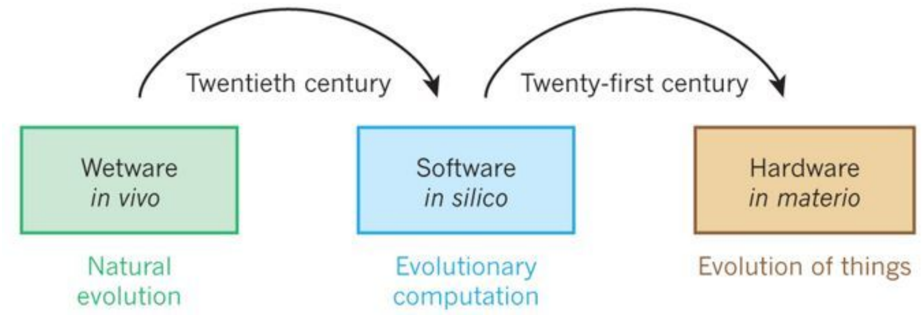
Automated Artist-Critic Coevolution

Starting from images of landmark buildings and natural objects, a virtual evolutionary artist competes against a deep convolutional neural network art-critic producing intriguing results.

Author: Alan Blair
https://PickArtSo.com

Evolutionary Robotics & Embodied Evolution

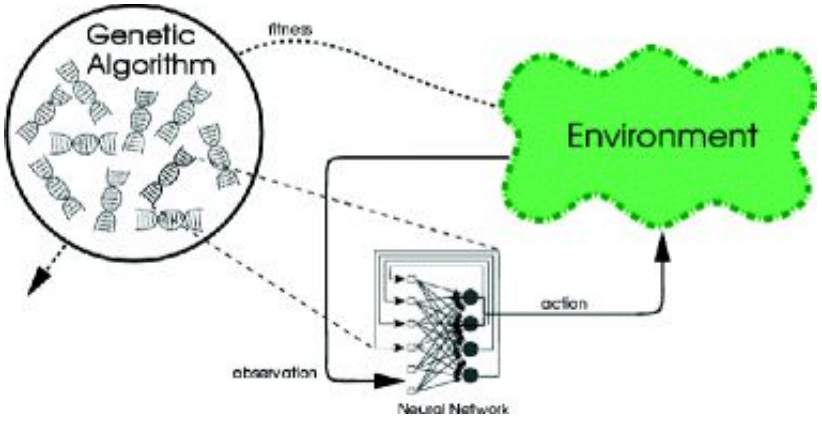
“Given the fact that evolution can produce intelligence, it is plausible that Artificial Evolution can produce Artificial Intelligence”. Gusz Eiben



From evolutionary computation to the evolution of things, *Nature volume 521, pages 476–482 (28 May 2015)*, by A. Eiben & Jim Smith https://www.nature.com/articles/nature14544

Neuroevolution

- EAs used to generate NN weights, topologies, rules or ensembles
- Fitness function measures performance in the task
- Can be applied more widely than supervised learning algorithms



Main Motivation: to train NNs in sequential decision tasks with sparse reinforcement information.

Robotics, Game Playing, ALife

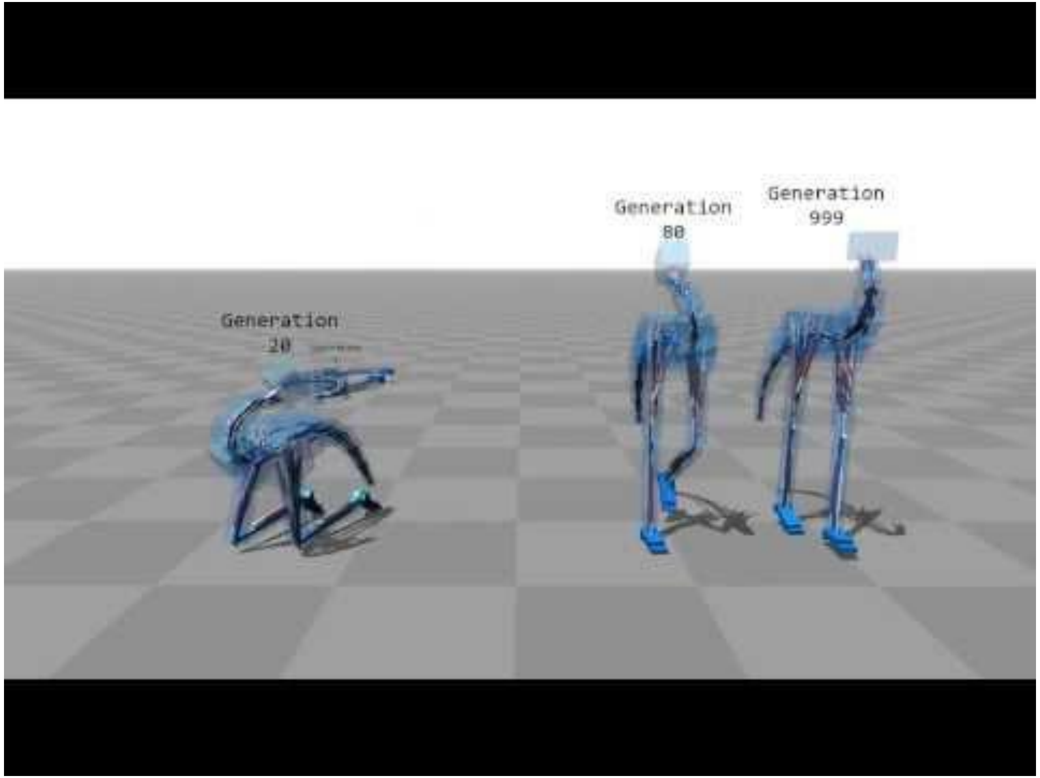
http://nn.cs.utexas.edu/index.php

Optimisation skeletal & muscle models

‘Teach’ creatures to walk.

Skeletal and muscle model has many parameters. These can be optimised.

Here, using an evolutionary algorithm (CMA-ES).



http://www.cs.ubc.ca/~van/papers/2013-TOG-MuscleBasedBipeds/2013-TOG-MuscleBasedBipeds.pdf

Other population-based algorithms: the social behaviour metaphor

Ant colony optimisation (ACO)

- Dorigo, Di Caro & Gambardella (1991).
- Inspired by the behaviour of real ant colonies
- A set of software agents artificial ants search for good solutions
- Ants build solutions incrementally by moving on the graph

Particle Swarm Optimization (PSO)

- Eberhart & Kennedy, 1995
- Inspired by social behaviour of bird flocking or fish schooling
- Solutions (called particles) fly through the search space by following the current optimum particles
- At each iteration they accelerate towards the best locations

Summary

Optimisation Problems

The travelling salesman problem

Vehicle routing

Other ‘fun’ and practical examples

Population-based algorithms

Inspired by evolution by natural selection

Main variants (GA, Hybrid, ES, GP)

Other population based approaches