Artificial Intelligence

CS3AI18 / CSMAI19

Lecture - 2/10: Problem Solving (Evolutionary Algorithms)

DR VARUN OJHA

Department of Computer Science

University of Reading

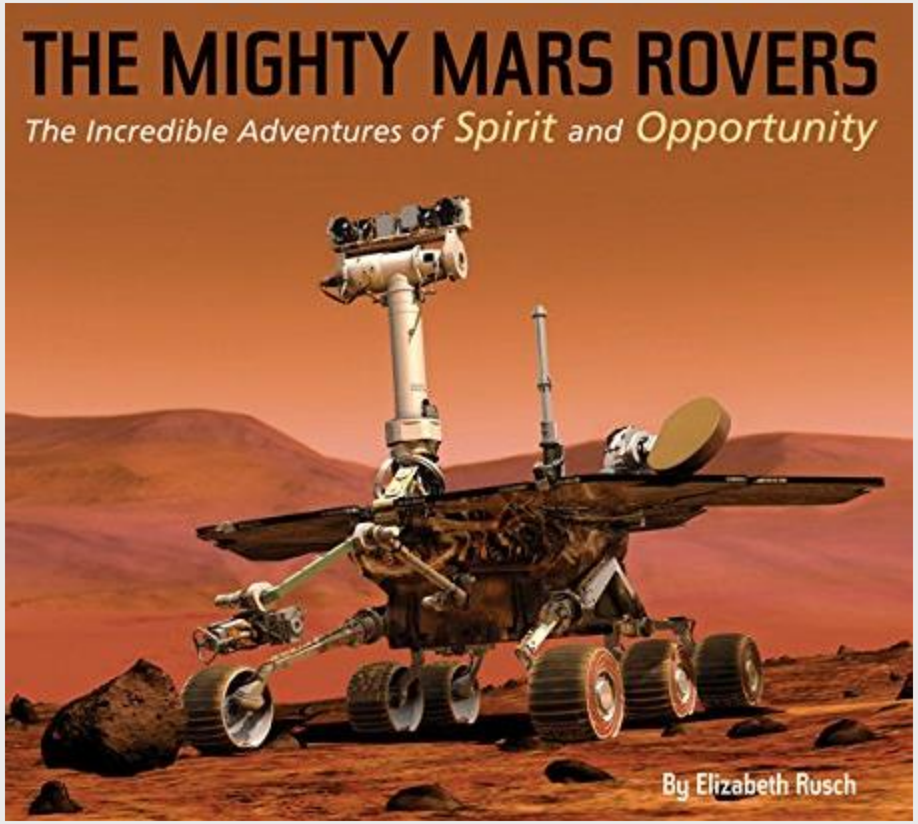
Learning Objectives

- On completion of this week, you will be able to
  - Aware of what problems AI solves
  - Understand problem solving using nature inspired intelligence
  - Aware of basic theory of the nature inspired intelligence
  - Learn algorithms (evolutionary algorithms) inspired by nature
  - Solve problem using Evolutionary algorithms

Content of this Lecture

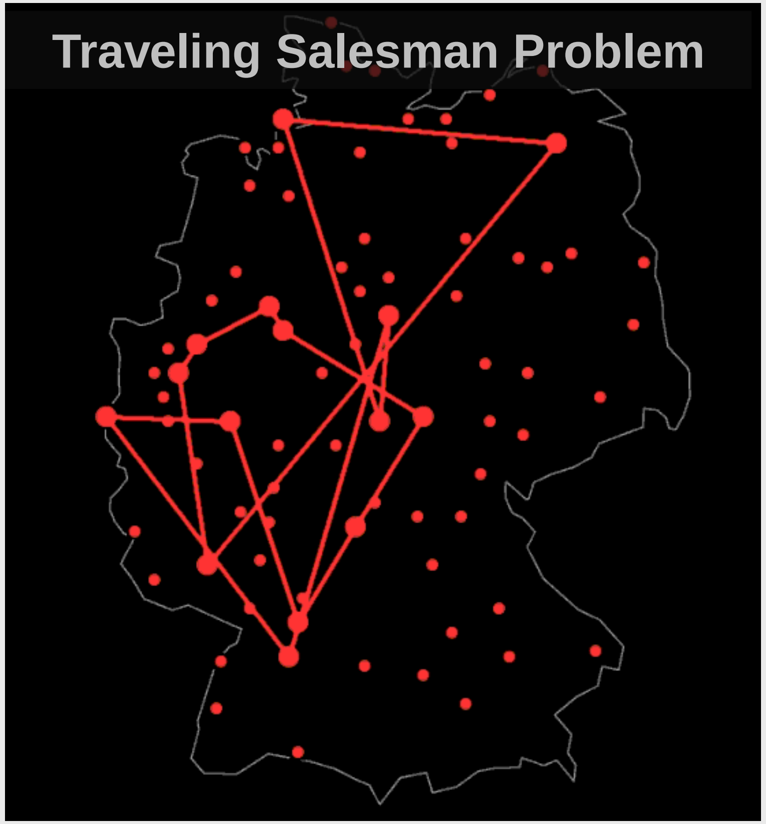
- Intro : Problems in AI
- Part – I : Nature inspired Intelligence
- Part – II : Theory of nature inspired intelligence
- Part – III : Evolutionary Algorithms
- Part – IV : Applications
- Quiz

Problem Solving and Search

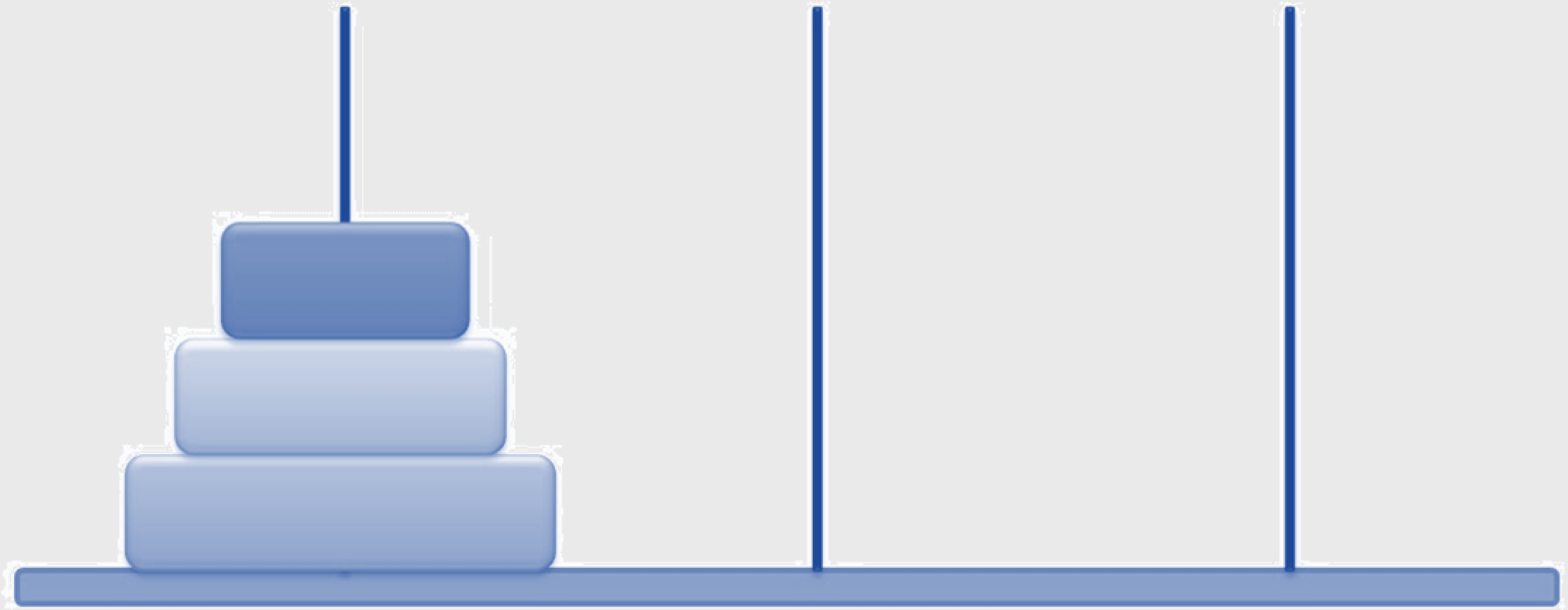


Source: Mighty Mars Rovers: The Incredible Adventures of Spirit and Opportunity by Elizabeth Rusch

Source: http://naturemoms.com/blog/2011/08/08/geocaching-a-family-treasure-hunt/



Source: http://www.mathema-ausstellung.de/nc/en/exhibition/chance/pictures49ea.html?image=4



Tower of Hanoi

2 minutes

START STOP

7:39 PM

Water-Jag Problem

2 minutes

START STOP

Problem Definition

Well-defined, i.e. what is the initial state and what is the goal state

Well-defined Goal

Objective function is known or an idea of what could be the solution looks like can be defined (method to measure)

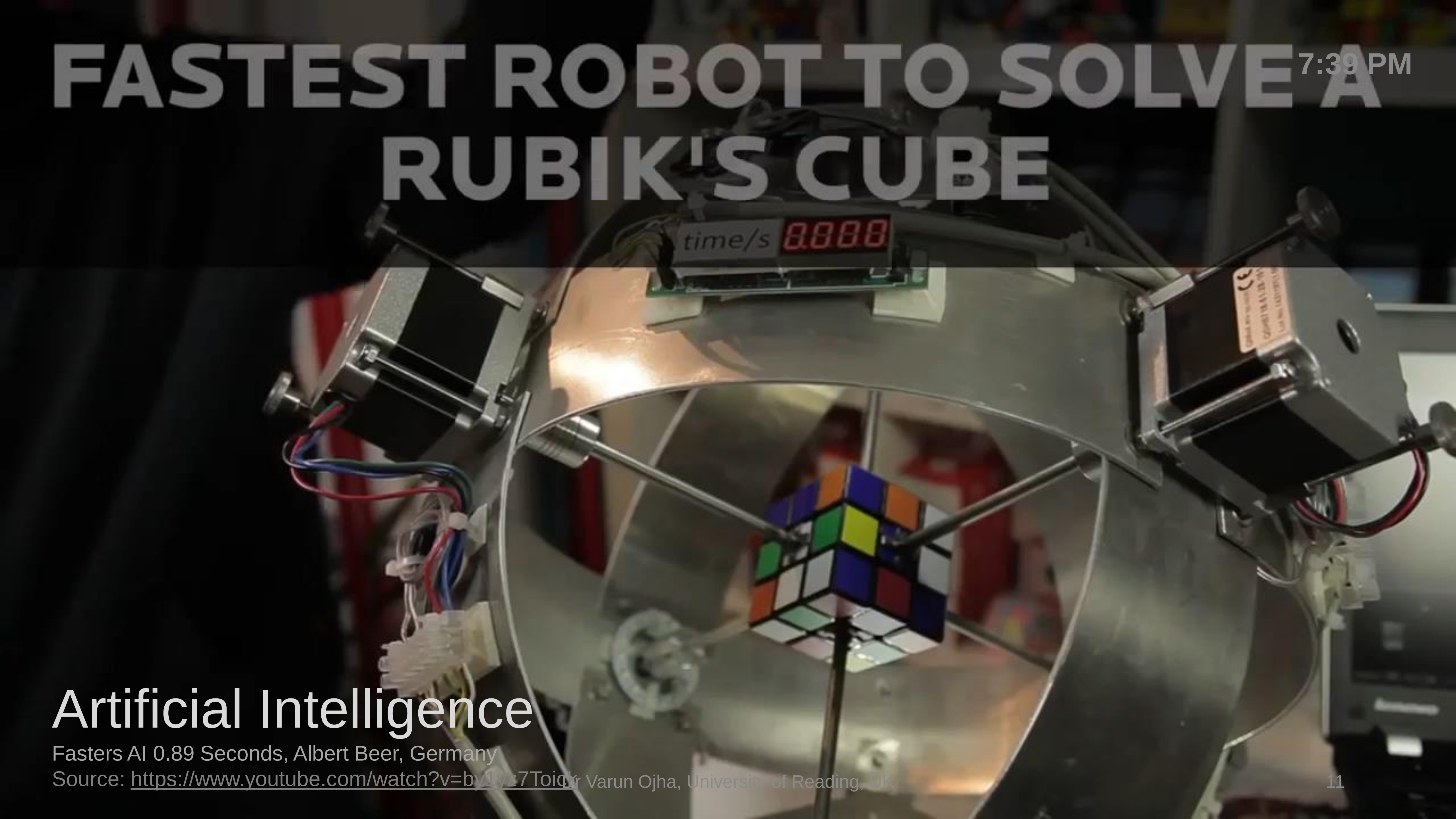
Human Vs Computer

Rubik's Cube is a search problem

Human Intelligence

Rubik's Cube World Record 4.73 Feliks Zemdegs

Source: https://www.youtube.com/watch?v=M5yjKpMXChI



Artificial Intelligence

Fasters AI 0.89 Seconds, Albert Beer, Germany

Source: https://www.youtube.com/watch?v=by1yz7Toich

Part 1
Nature's Intelligence

Nature Inspired Intelligence

Evolution

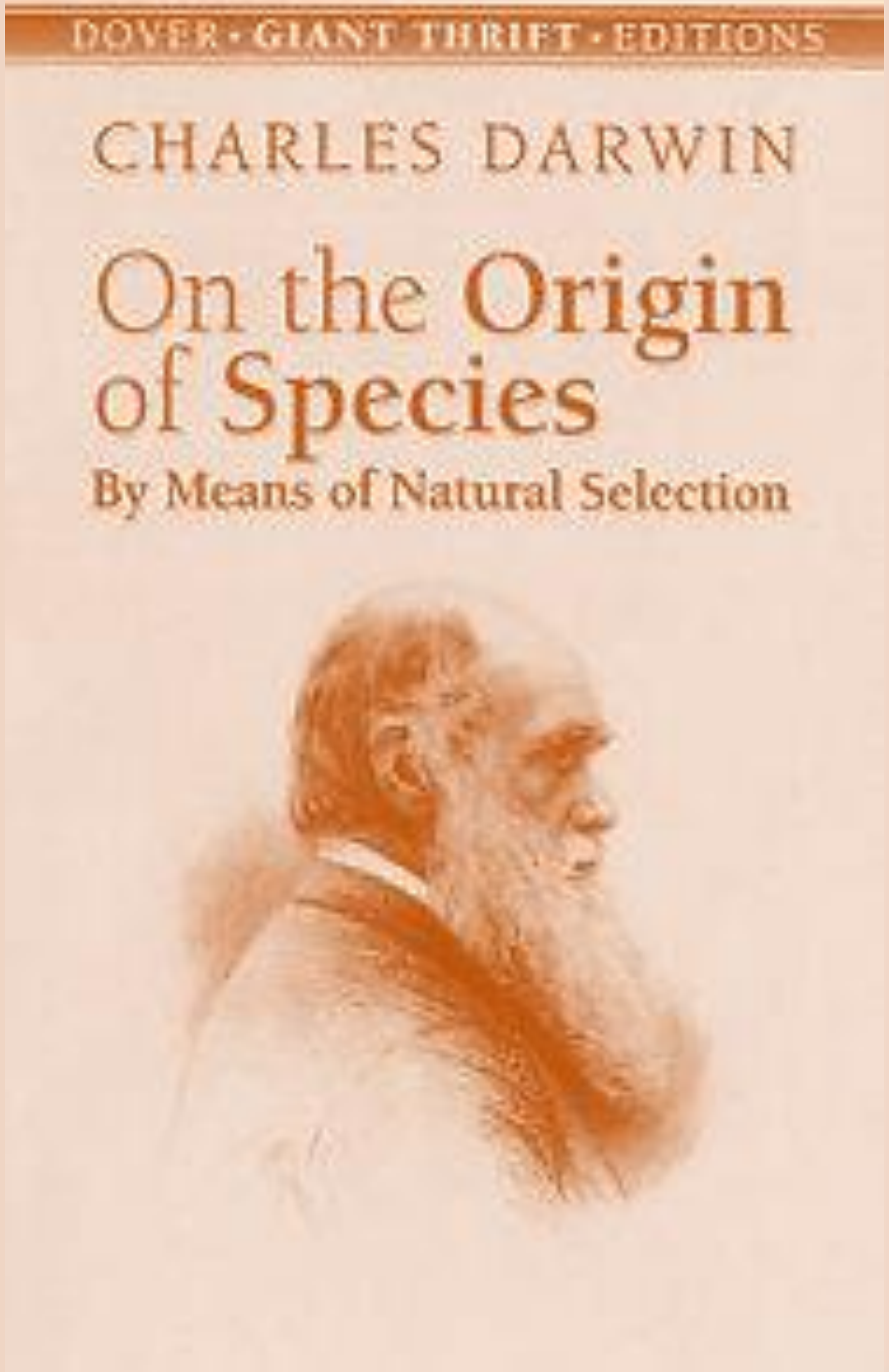
7:39 PM

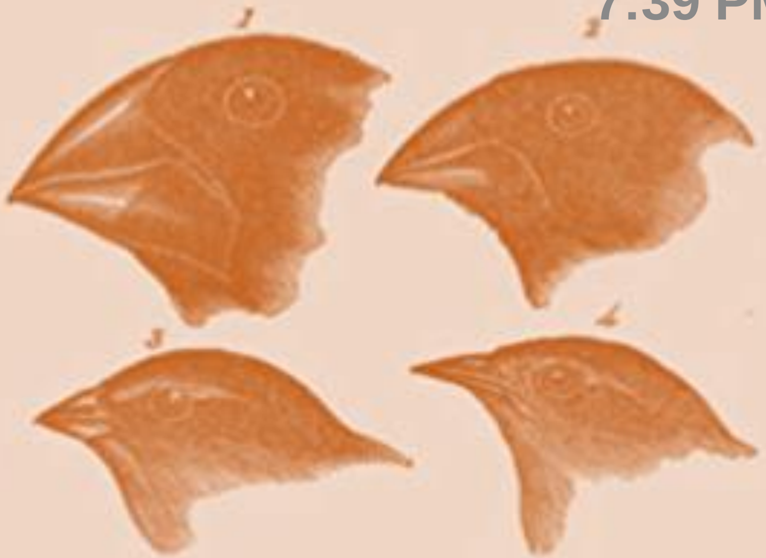
Source: Milewicz. A, (2018) How Sentient's using the principles of evolution to create AI solutions, https://tq.co/stories/sentient-evolution-ai-solutions (Accessed 18 Jan 2020)

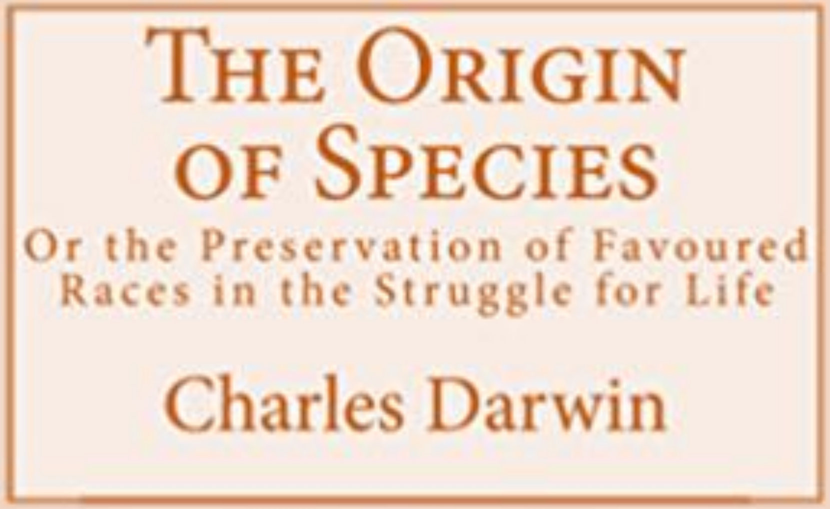
DARWINIAN EVOLUTION

NOVEMBER

1859







Darwinian Evolution

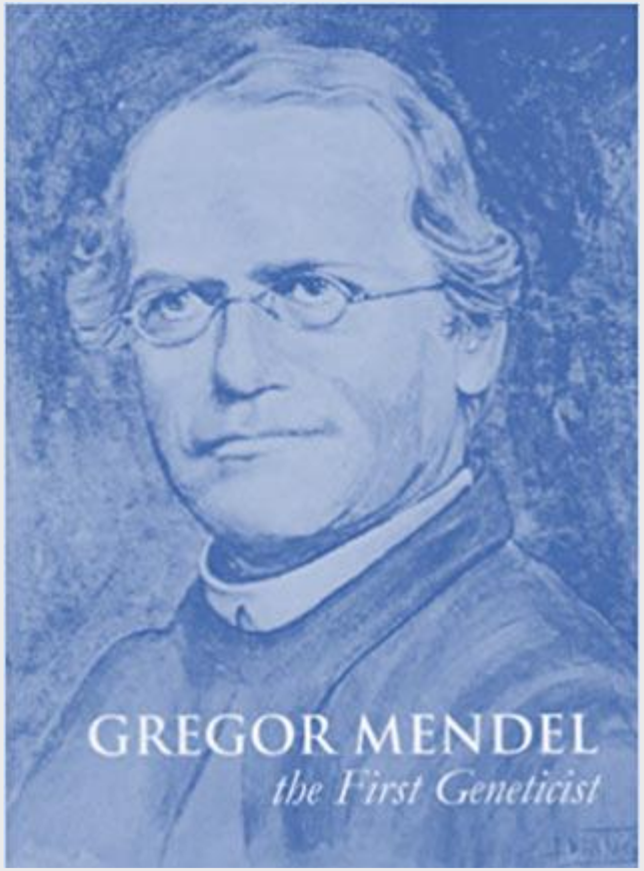
C. Darwin, On the Origin of Species by Means of Natural Selection, 1859

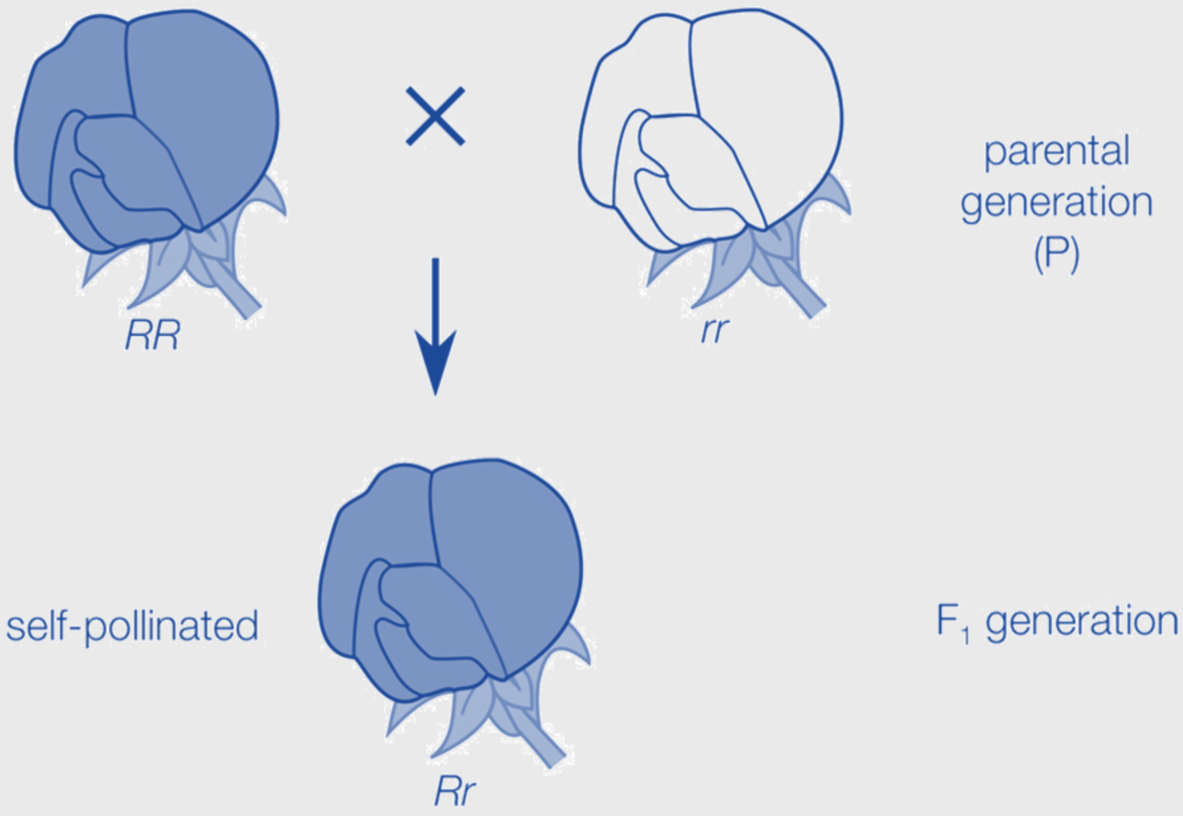
Four Postulates

1. Individuals within species are variable;
2. Some of the variations are passed on to offspring;
3. In every generation, more offspring are produced than can survive;
4. The survival and reproduction of individuals are not random: The individuals who survive and go on to reproduce, or who reproduce the most, are those with the most favourable variations. They are naturally selected.

Mendelian Genetics – 1865

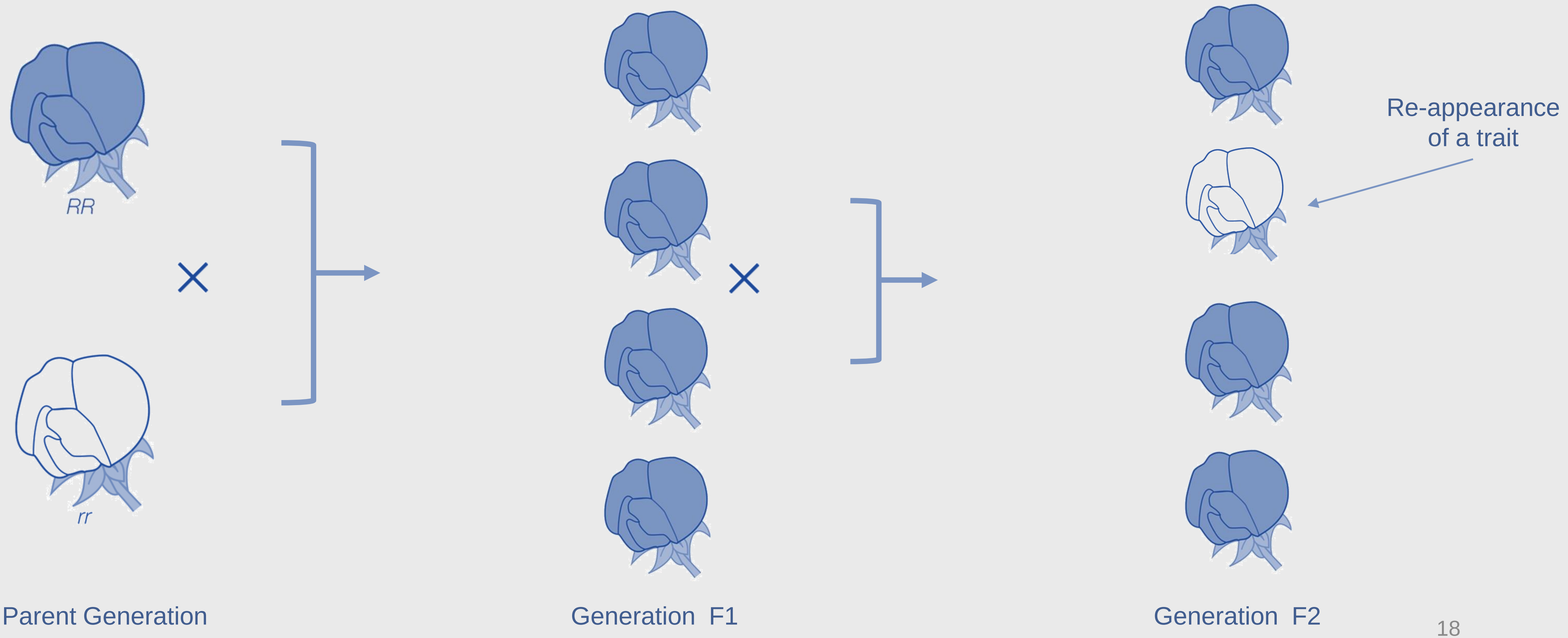
Biological programming behind all life forms





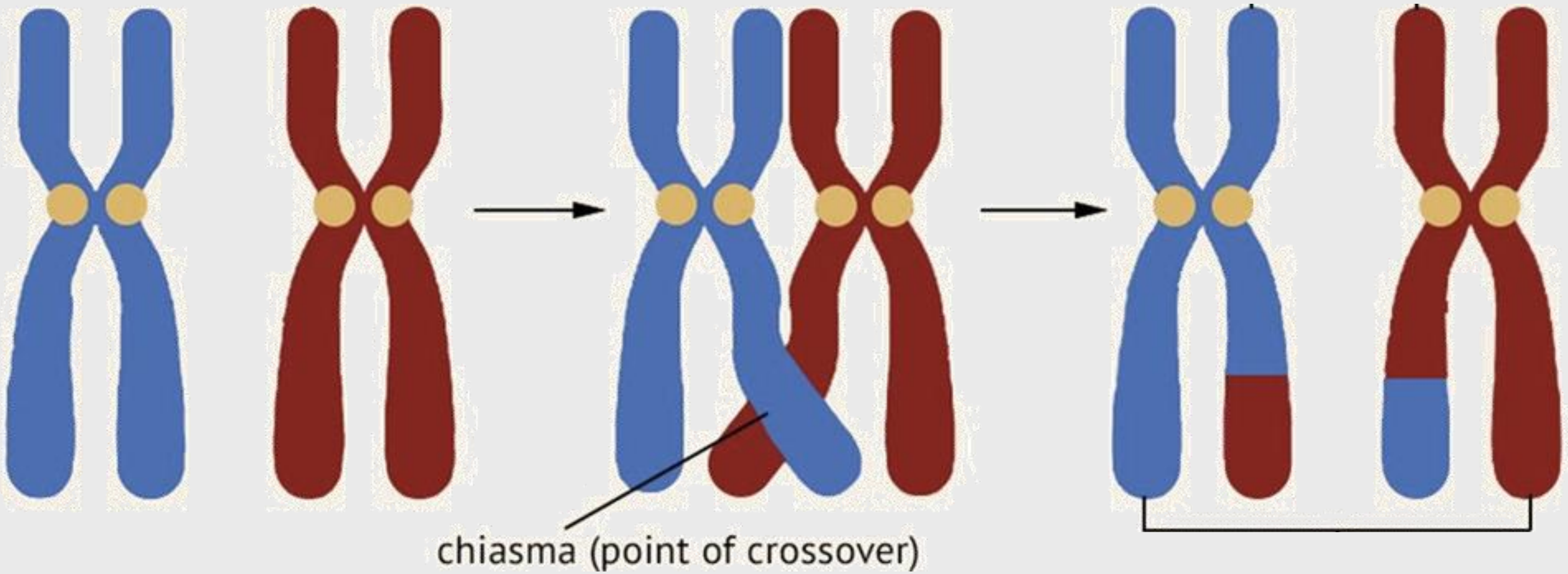
Mendelian Genetics – 1865

Biological programming behind all life forms



Biological Programming

- Genotype - set of genes an organisms carries
- Phenotype – organisms physical appearance / characteristics



SURVIVAL OF THE FITTEST

The fittest phenotype propagates to next generation

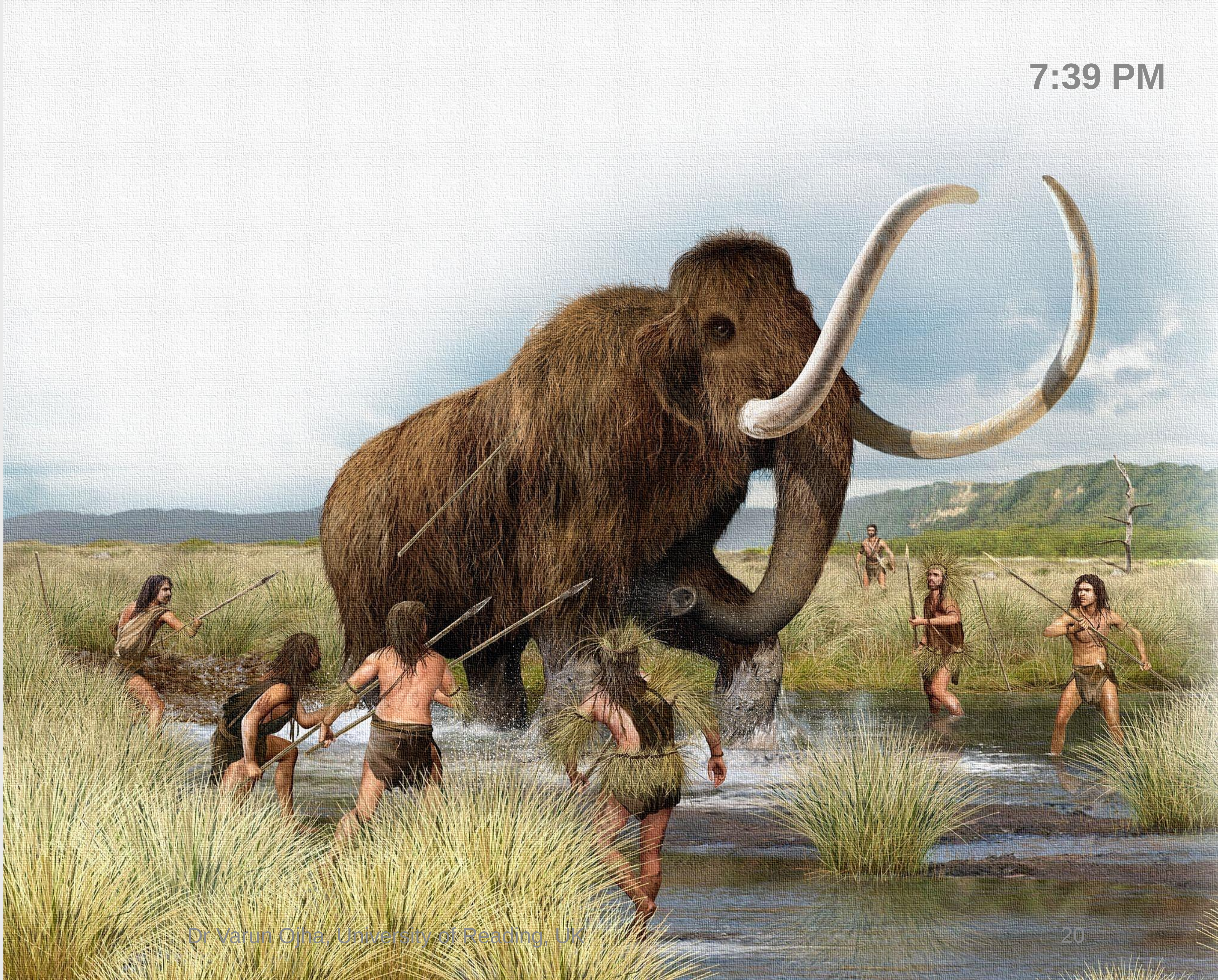


Image Source
https://en.wikipedia.org/wiki/File:Hunting_Woolly_Mammoth.jpg

Nature of Natural Selection

Freeman, S. & Herron, J.. C. Evolutionary Analysis, 2nd ed., Prentice-Hall 2001.
Stearns, S. C. and Hoekstra, R. F. Evolution. An Introduction, Oxford University Press, 2000

Natural Evolution acts

- On Individuals, but the Consequences occur in the population
- On Individuals, not groups
- On Phenotypes, but evolution consist of changes in the Genotype
- On existing traits, but can produce new traits

Evolution

- Is backward looking
- Is not perfect
- Is non-random
- Is not progressive

Why are we Interested ?

Results of Evolution are

- 'Creative'
- 'Surprising'
- 'Unexpected'
- 'Highly adapted' to 'Environmental Niches'

Why are we Interested ? (Contd..)

Evolutionary process

- Unsupervised !
- No 'conscious' design
- No knowledge involved apart from reproductive fitness
- But ! - Natural Evolution had an extremely long time (3.7 Billion Years!)
- Natural Evolution acts in parallel

Why are we Interested ? (Contd..)

Demo: https://rednuht.org/genetic_cars_2/ (max 2 min)

Can a program 'create things like this' ?

Part 2
Theory

Evolutionary Computation

Is the study of computational systems which use ideas and get inspirations from **natural evolution** and other **biological systems** (Artificial Ants, Artificial Cells, Artificial Immune Systems)

Evolutionary Algorithms

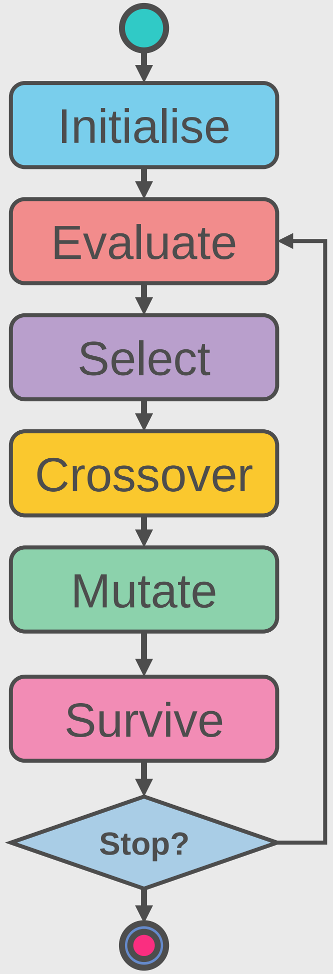
Evolutionary algorithms (EA) are biological evolution inspired algorithms.

Evolutionary algorithms:

- are a subset of Evolutionary Computation (EC)
- use natural selection inspired operators such as **selection**, **recombination,** and **mutation.**
- are closely linked to AI techniques, especially search techniques.
- can be regarded as **population-based** **stochastic generate-and-test** algorithms.

Evolutionary Algorithms - STEPS

1. **t** := **0**; // Generation 0
2. Generate **Initial Population** $\mathbf{P^{(t)}}$ at random;
3. **Evaluate the fitness** of each individual in $\mathbf{P^{(t)}}$;
4. **Until** (termination condition **not** met) **do**
    1. **Select parents**, $\mathbf{Pa^{(t)}}$ from $\mathbf{P^{(t)}}$ based on their fitness in $\mathbf{P^{(t)}}$;
    2. Apply **crossover (recombination)** to create offspring from parents: $\mathbf{Pa^{(t)}} \rightarrow \mathbf{O^{(t)}}$
    3. Apply **mutation** to the offspring: $\mathbf{O^{(t)}} \rightarrow \mathbf{O^{(t)}}$
    4. **Evaluate the fitness** of each individual in $\mathbf{O^{(t)}}$;
    5. **Survive population** $\mathbf{P^{(t+1)}}$ from current offspring $\mathbf{O^{(t)}}$ and parents $\mathbf{P^{(t)}}$;
    6. **t** := **t** + **1**; // Next generation
5. **end-do**



What’s in favour of EAs?

- There is **no restriction on the fitness** (objective) function.
- It can be **non-differentiable** or **even discontinuous**.
- There is no need to know the exact form of the objective function.
- Simulation can be used to derive a fitness value.

- The **initial population does not have to be generated randomly**
- You can use existing knowledge to seed population
- The representation does not have to be binary

- There are **alternative options for all steps of the algorithm**
- Genetic operators, selection, termination

Components of EAs

1. Representation of individuals: Coding (**Binary**, **Integer**, **Real**).
2. Evaluation method for individuals: **Fitness**.
3. Initialization procedure - 1st generation (**random** or **ad-hoc**).
4. Definition of variation operators (**mutation** and **crossover**).
5. Parent (**mating**) selection mechanism.
6. Survivor (environmental) **selection** mechanism.
7. Technical parameters (e.g. **mutation rates**, **population size**).

Biological Programming

Evolutionary Algorithm

7:39 PM

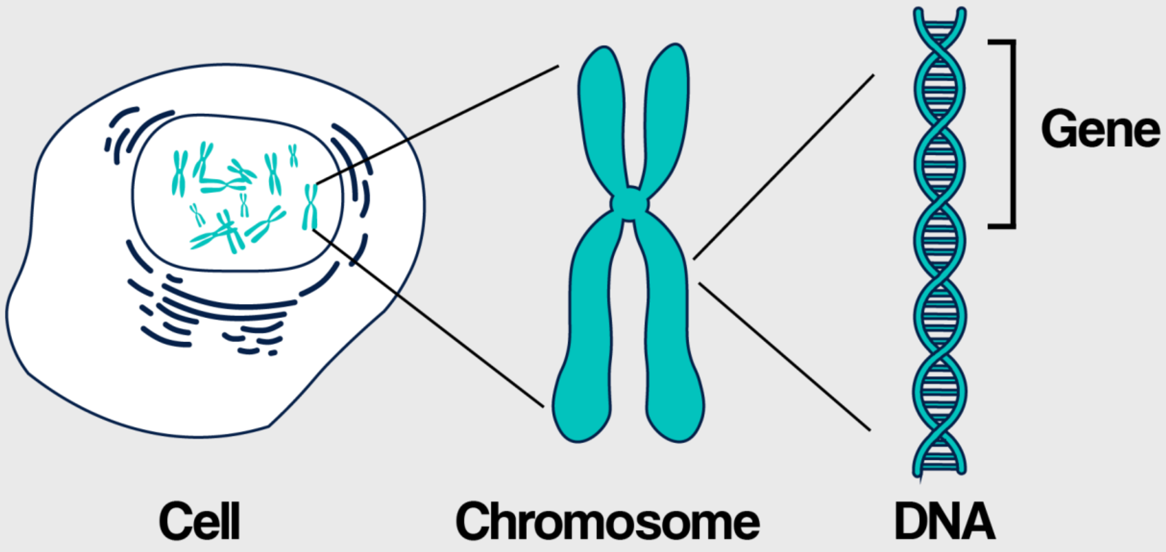


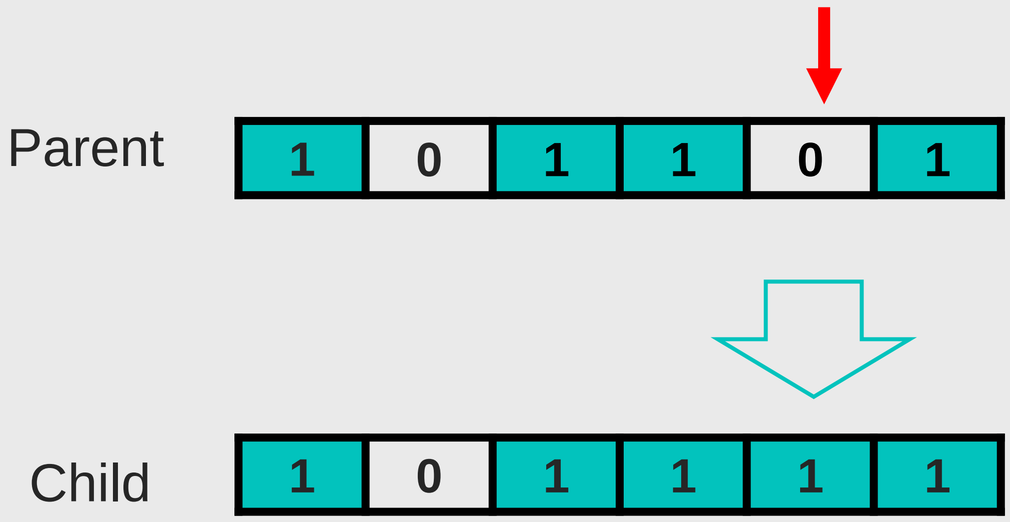
Image Source: https://kintalk.org/genetics-101/

Binary **Chromosome**	1	0	1	1	0	1
Integer **Chromosome**	2	1	1	3	9	7
Real **Chromosome**	2.5	1.5	3.7	8.1	1.5	0.7
String **Chromosome**	A	E	C	B	Z	X

Gene

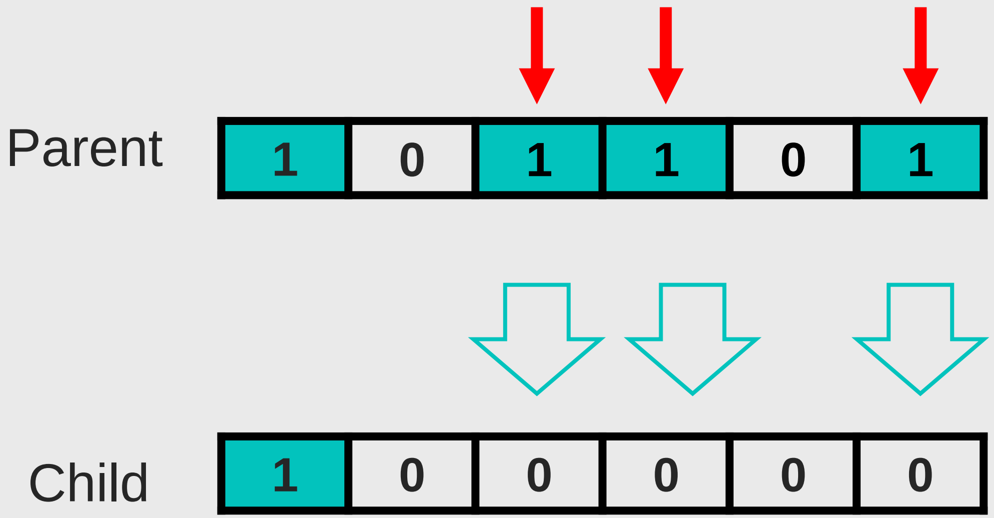
Mutation

Random selection of a gene



Single point mutation: Bit flip

Random selection of multiple genes

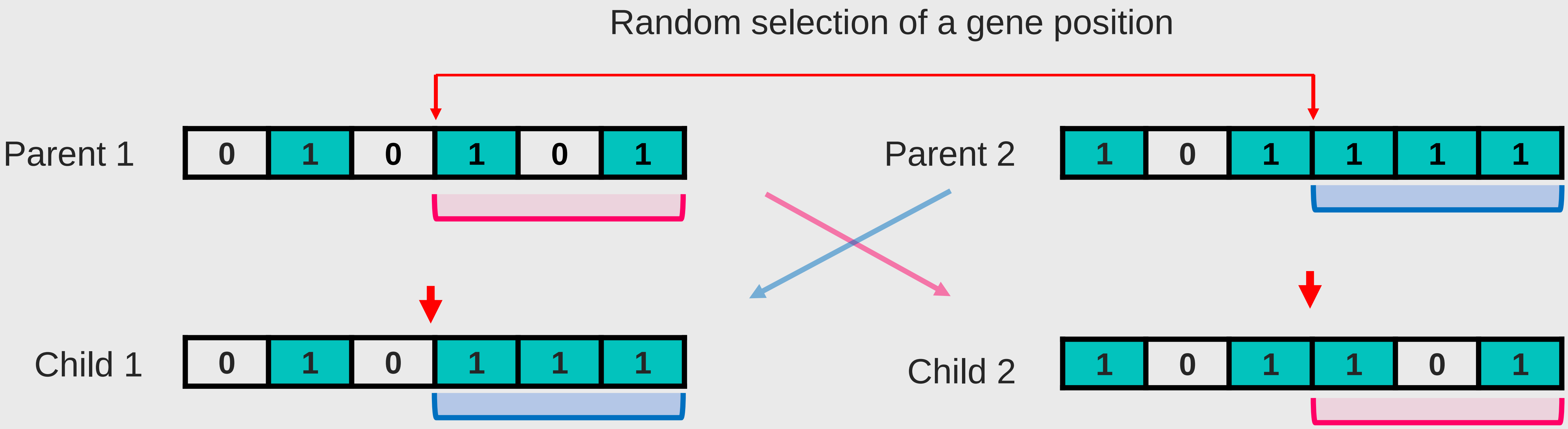


Multi-point mutation: Bit flip

Mutation Operator (Remarks)

- **Types**
  - Bit-Flipping
  - Random bit assignment
  - Multiple bit mutations
  - Inversely proportional hypermutation (Artificial Immune System)
- **Mutation rate**: the probability of applying mutation
- **Per-chromosome mutation rate vs. per-gene (bit) mutation rate**
- **Custom variations** for non-binary representations

Crossover (Recombination)



Single point crossover

Crossover Operator (Remarks)

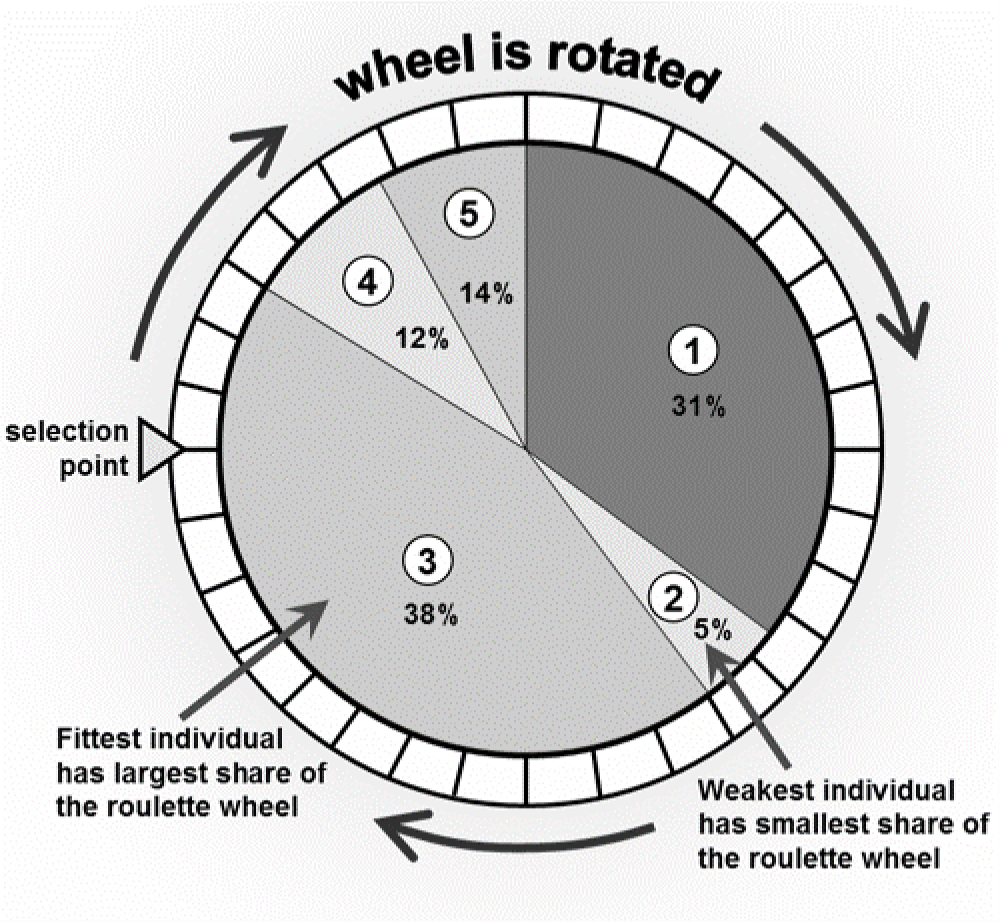
- **Type**
  - One-Point Crossover
  - K-Point Crossover
  - Uniform Crossover
- **Crossover rate:** the probability of applying crossover
- **Mutation rate and Crossover rate trade-offs**
- **Custom variations** for nonlinear representations

Selection

Parent selection from a population of 5 individuals

No.	Chromosome	Value_{10}	X	Fitness $f(x)$	% of Total
1	0001101011	107	1.05	6.82	31
2	1111011000	984	9.62	1.11	5
3	0100000101	261	2.55	8.48	38
4	1110100000	928	9.07	2.57	12
5	1110001011	907	8.87	3.08	14
Totals				22.05	100

Example population of 5 for: $f(x) = -\frac{1}{4}x^2 + 2x + 5$



Read the source for details:
Source: http://www.edc.ncl.ac.uk/highlight/rhjanuary2007g02.php

Selection (Remarks)

- **Type**
  - Roulette wheel selection
  - Fitness proportional selection
  - Rank-based selection
  - Tournament selection
- **Aging Operator** like more complex operator concepts
- **The Selection Pressure**: Different selection operators produce different behaviour

Exploration vs. Exploitation

Search Bias!

Some offspring tend to be more likely to be generated than others.

- **Bias depends** on representation and operators
- **Crossover bias:** e.g. One-point crossover vs. Uniform crossover
- **Mutation bias:** e.g. 1-bit-flip vs. K-bit-flip
- **Remarks:** Search operators are applied to individuals. It is very important to realise the **interdependency between operators and the representation** of individuals.

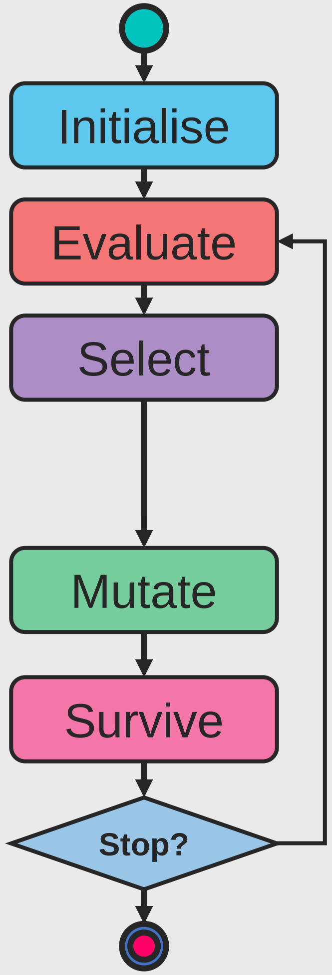
Part 3
Algorithms

Evolutionary Algorithm

- Evolutionary Programming
- Evolutionary Strategy
- Genetic Algorithm
- Genetic Programming
- Immunological Algorithms

Evolutionary Programming (EP)

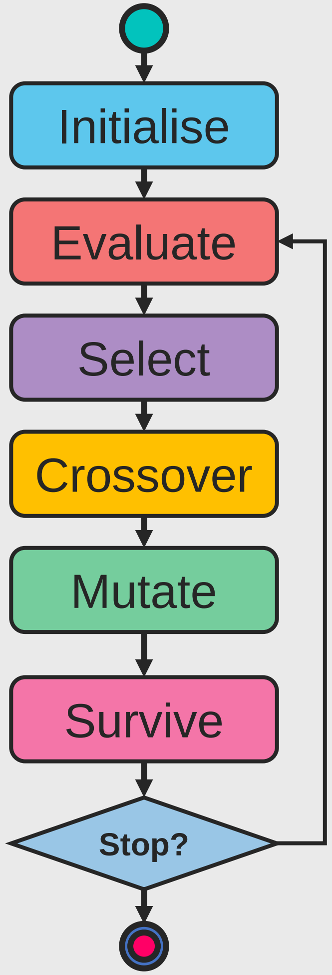
L. Fogel, 1966



- ✓ **Individuals:** The individuals are Finite State Machine (FSM) representation, real-valued vectors, ordered lists, graphs.
- ✓ **Selection:** All N individuals are selected to be parents, and then they are mutated, producing N children.
- ✓ **Operators:** Mutation is based on the representation used, and is often adaptive. For example, when using a real-valued vector, each variable within an individual may have an adaptive mutation rate that is normally distributed
- ✓ **No recombination**
- ✓ **Survival:** Children are **evaluated** and **N** survivors are chosen from the **2N** individuals, using a probabilistic function based on fitness (individuals with a greater fitness have a higher chance of survival).

Evolutionary Strategy (ES)

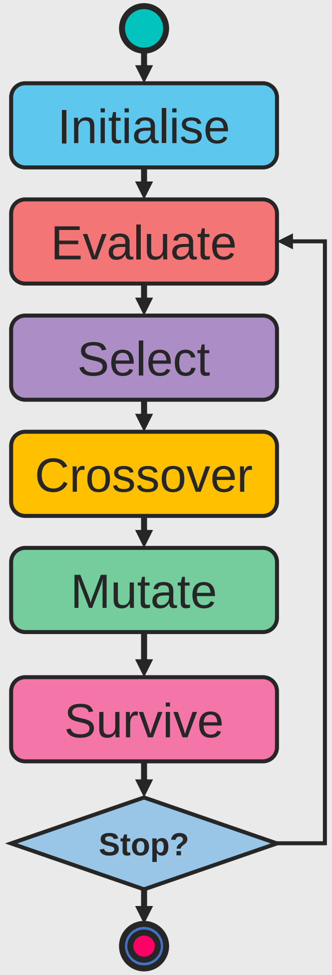
Rechenberg & Schwefel, 1973



- ✓ **Individuals:** typically uses real-valued vectors.
- ✓ **Selection:** uniformly randomly.
- ✓ **Operators:** Pairs of parents produces children via recombination. The number of children created is greater than N.
- ✓ **Survival is deterministic:**
  - ✓ ES allows the N best children to survive, and replaces the parents with these children.
  - ✓ ES allows the N best children and parents to survive.
- ✓ **Like EP, adapting mutation.**
- ✓ Unlike EP, recombination does play an important role in ES, especially in adapting mutation.

Genetic Algorithm (GA)

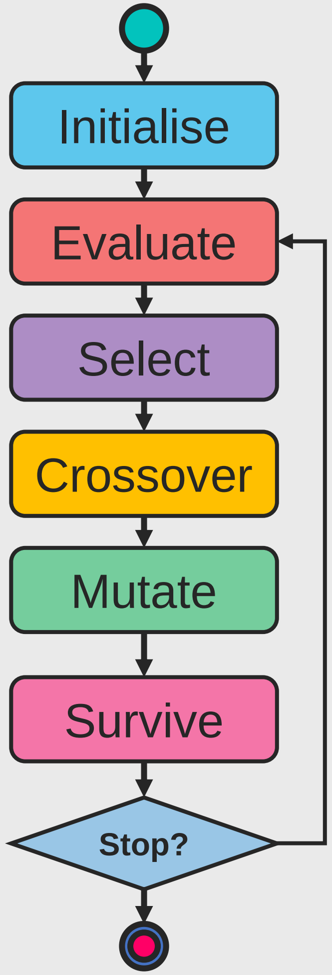
John Holland, 1975



- ✓ **Individuals:** traditionally use a more domain independent representation, namely, bit-strings.
- ✓ **Selection:** Parents are selected according to a probabilistic function based on relative fitness.
- ✓ **Operators:** N children are created via recombination from the N parents.
  - ✓ Crossover is important: the primary search operator
  - ✓ Mutation flips bits with some small probability (background operator).
- ✓ **Survival:**
  - ✓ The N children are mutated and survive, replacing the N parents in the population.

Genetic Programming (GP)

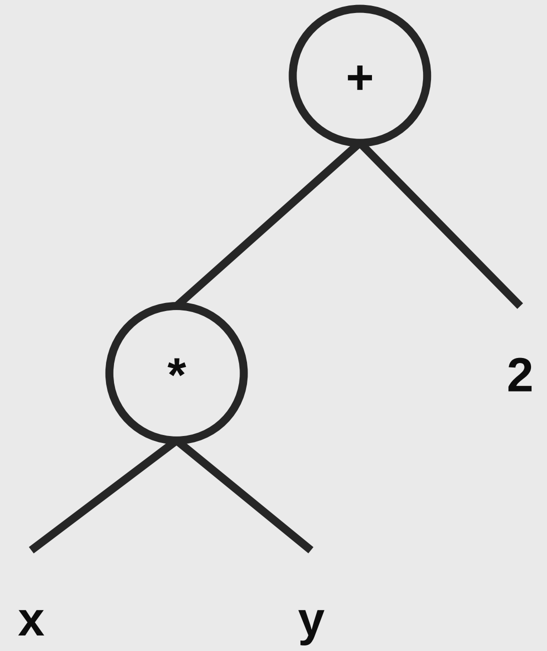
John Koza, 1992



- ✓ **Individuals:** tree representation of program or expression.
- ✓ **Selection:** Parents are selected according to a probabilistic function based on relative fitness.
- ✓ **Operators:** N children are created via recombination from the N parents.
  - ✓ Crossover is important: the primary search operator
  - ✓ Mutation flips bits with some small probability (background operator).
- ✓ **Survival:**
  - ✓ The N children are mutated and survive, replacing the N parents in the population.

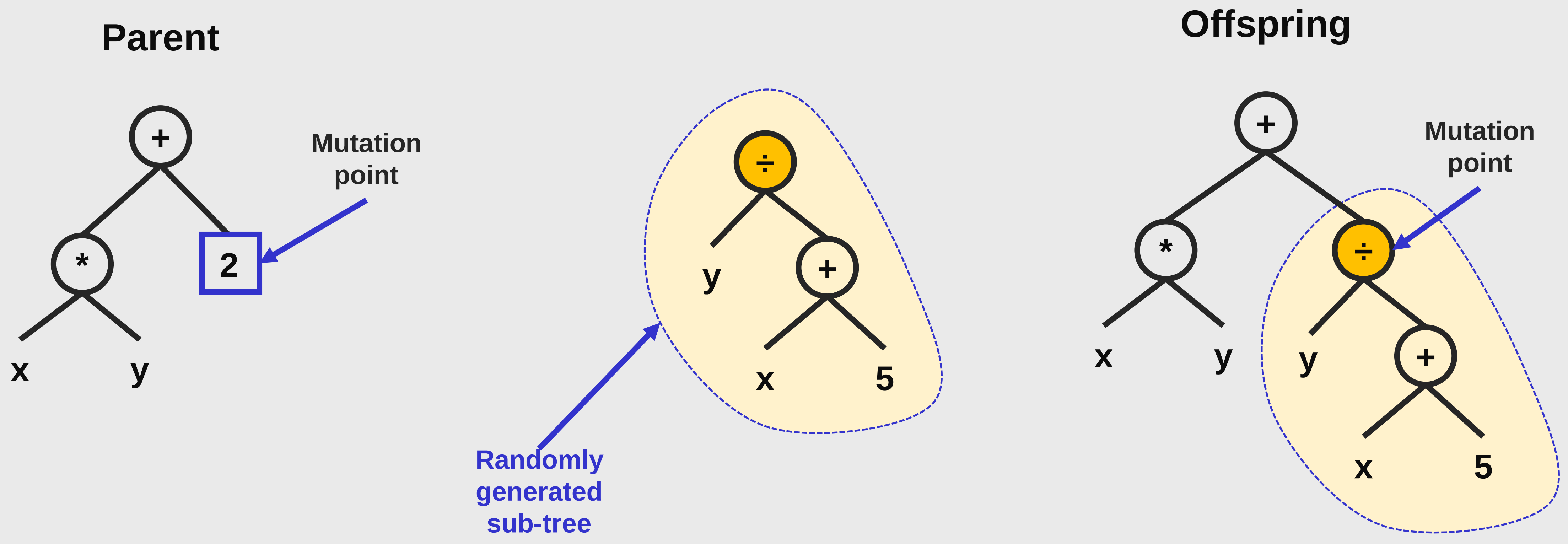
Expression → Tree

X * Y + 2 →



Mutation Operator (GP)

Mutation by **punning a sub-tree** and **replacing** by **randomly generated-sub-tree**



Mutation Operator (GP)

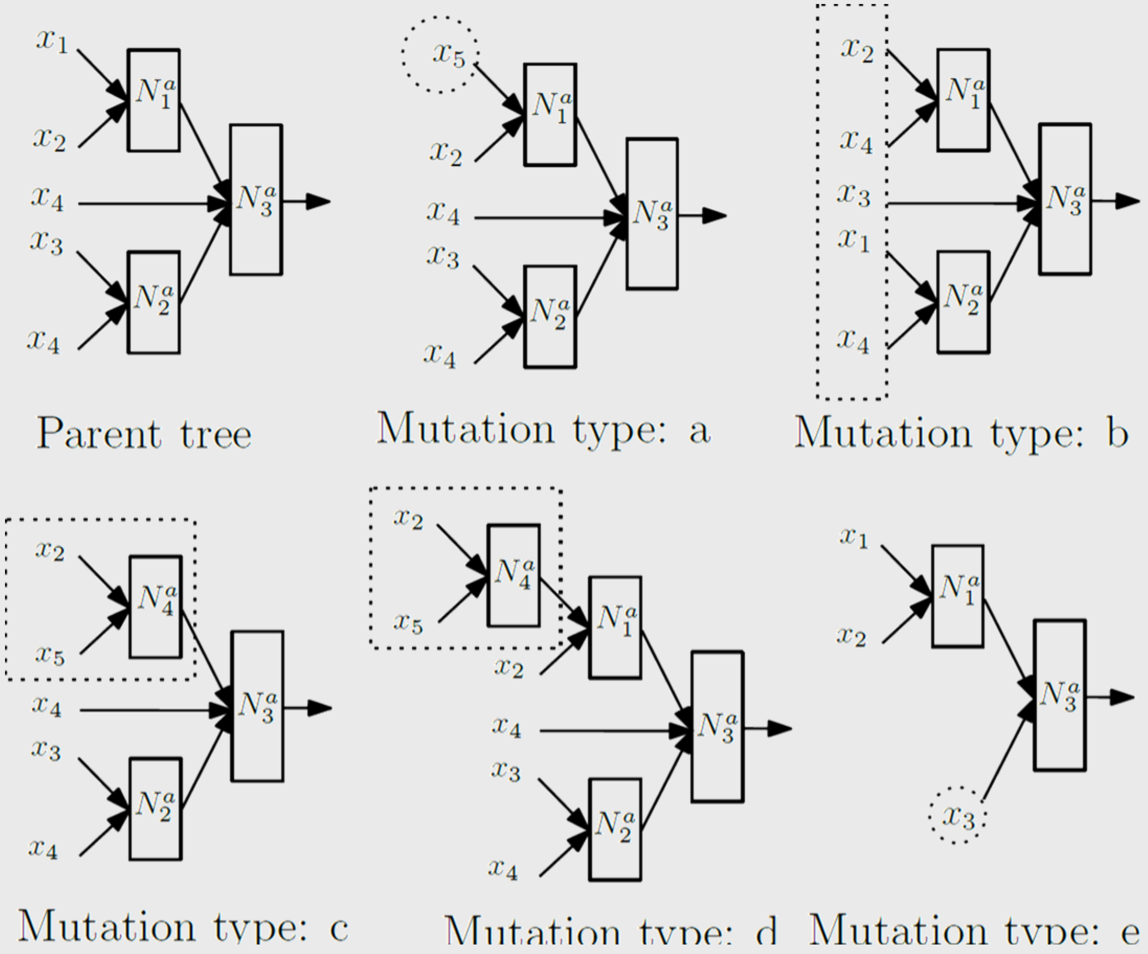
a) Mutation at a **single leaf** node.

b) Mutation at **all leaf nodes**

c) Mutation by **punning a sub-tree** and replace by randomly generated-Sub-tree

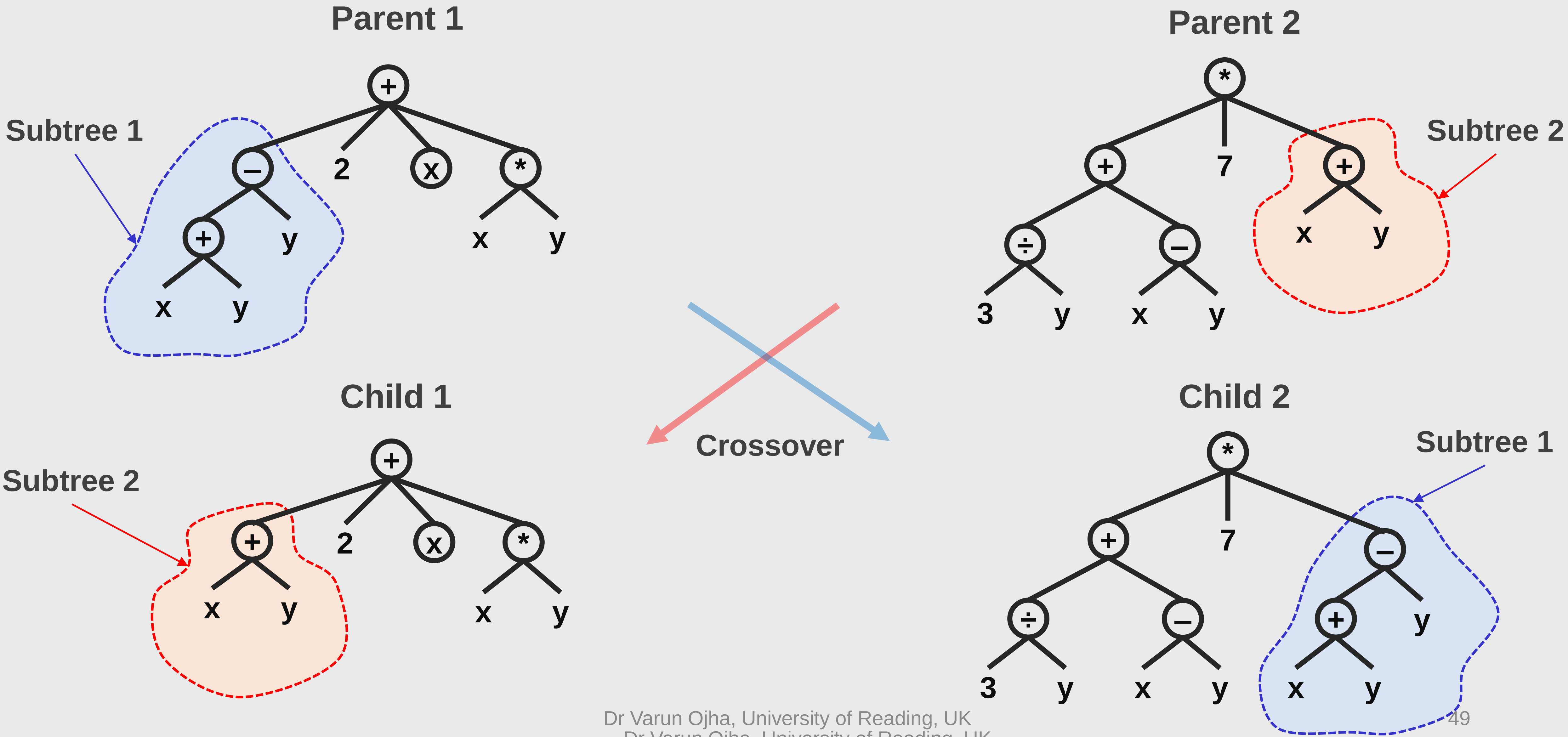
d) Mutation by **growing a tree**/appending a randomly generated sub-tree

e) Replace a subtree by a leaf node



Ojha, V.K., Snášel, V. and Abraham, A., 2017. Multi-objective programming for type-2 hierarchical fuzzy inference trees. *IEEE Transactions on Fuzzy Systems*, *26*(2), pp.915-936.

Crossover Operator (GP)



Dr Varun Ojha, University of Reading, UK

Advantages of EAs

- **Widely applicable**, also in cases where no (good) problem specific techniques are available:
- Multimodalities, discontinuities, constraints.
- Can **work for noisy objective functions**.
- Multiple criteria decision making problems.
- **No presumptions** with respect to the problem space.
- Low development costs; i.e. costs to adapt to new problem spaces.
- The solutions of EA's have **straightforward interpretations**.
- They **can be run interactively** (online parameter adjustment).

Drawbacks of EAs

- **No guarantee for finding optimal solutions** within a finite amount of time.
- **No complete theoretical basis** (yet), but much progress is being made (*X.Yao, J. He. Artificial Intelligence, 145, 2003*).
- **Parameter tuning** is largely based on **trial and error** (genetic algorithms); solution: Self-adaptation (evolution strategies).
- Often **computationally expensive**: Parallelism.

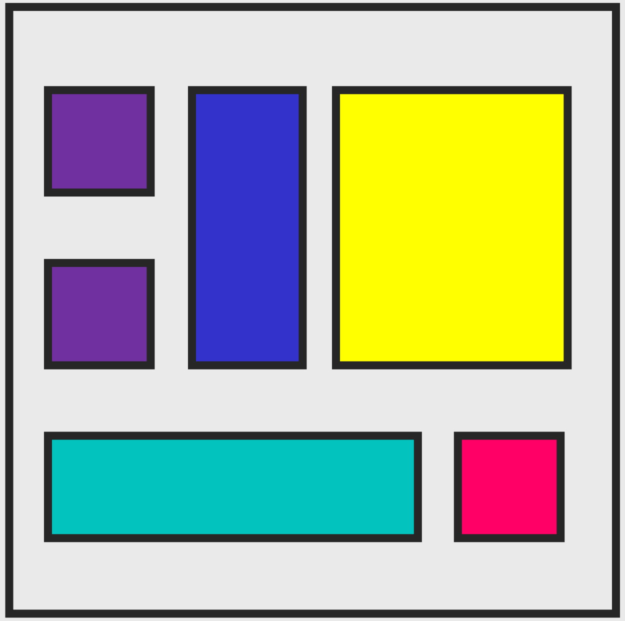
Part 4
Applications

Areas EA Application

- Optimization and Problem Solving
  - (J. Foster, Evolutionary Computation, Nature Genetics Reviews, 2001).
- NP-Complete Problem, e.g., e.g., the traveling salesman problem
  - (Lemmon & Milinkovitch, PNAS, 99(16), 2002)
- Protein Folding
  - (M. Damsbo et al., PNAS, 101(19), 2004;G. Nicosia, V. Cutello, Narzisi, Royal Society Interface, 2005).
- Automated Synthesis of Analog Electrical Circuits;
  - (J. Koza et al., Genetic Programming III, Morgan Kaufmann, 1999).
- Control and Modelling
  - (Assion et al, Science, 282, 1998).

Applications of EAs

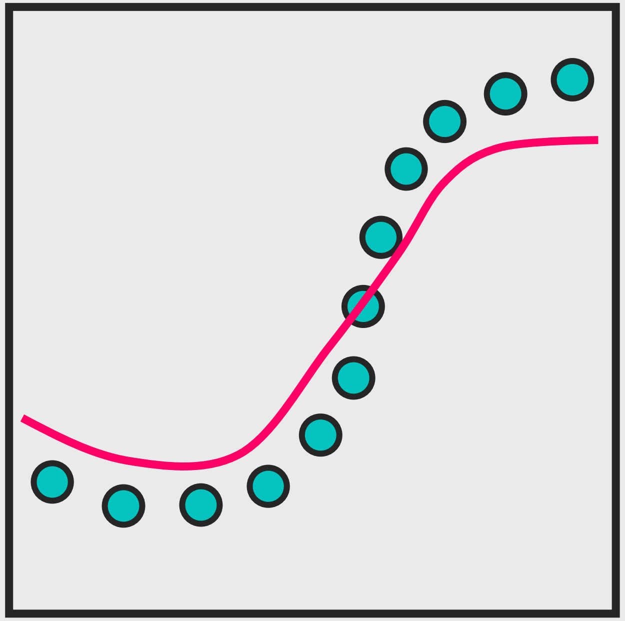
COMBINATORIAL FUNCTION OPTIMIZATION



Bin packing

A fun example is: Phrase solver
https://subprotocol.com/system/genetic-hello-world.html

CONTINUOUS FUNCTION OPTIMIZATION



Curve fitting

An example is Curve fitting
https://subprotocol.com/system/genetic-regression-curve.html

Turing's Three Approaches for Intelligence Machines

1. **Logic-driven**
2. **Knowledge-based**
3. **Evolutionary Search**

Turing’s Third Approaches for Intelligence Machines

In 1950 Turing described how evolution and natural selection might be used to create an intelligent program:

“... we cannot expect to find a good child machine at first attempt. One must experiment with teaching one such machine and see how well it learns. One can then try another and see if it is better or worse”.

--Turing A. M., Computing machinery and intelligence,Mind,1950.

Turing's Third Approaches for Intelligence Machines

Structure of Machine:
Individual in EAs

"... we cannot expect to find a good **child machine** at first attempt. One must **experiment with teaching** one such machine and see how well it learns. One can then try another and see if it is **better or worse**".

Change in machine:
Mutation

Judgment of the experiment:
Natural Selection