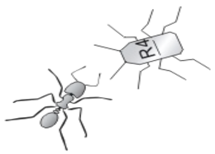
Evolutionary Computation

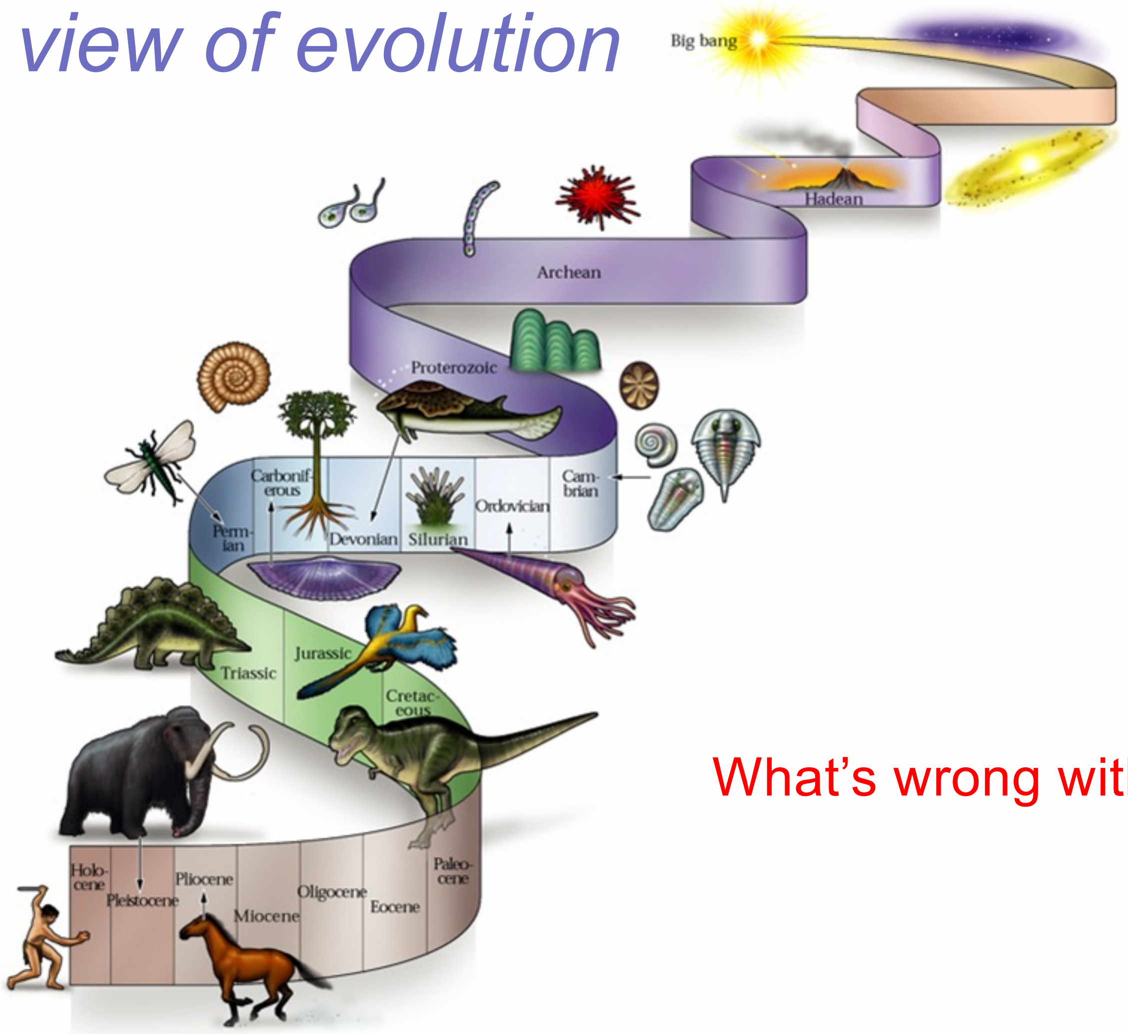
Companion slides for the book *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies* by Dario Floreano and Claudio Mattiussi, MIT Press

What you will learn today

4 pillars of evolution
Genetic basis of natural evolution
The algorithmic loop of artificial evolution
Choosing a genetic representations
Building an initial population
Devising a fitness function
Selection and reproduction methods
Mutations and crossover
Measuring evolution
Function optimization by evolution
Types of evolutionary algorithms
A simple example: evolving an antenna design

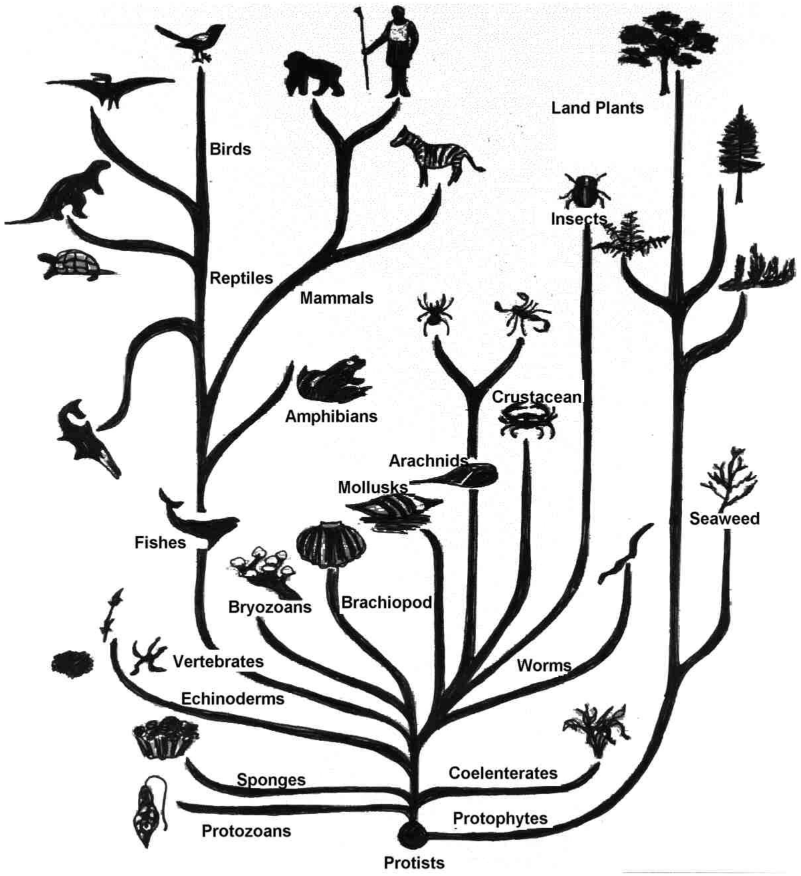


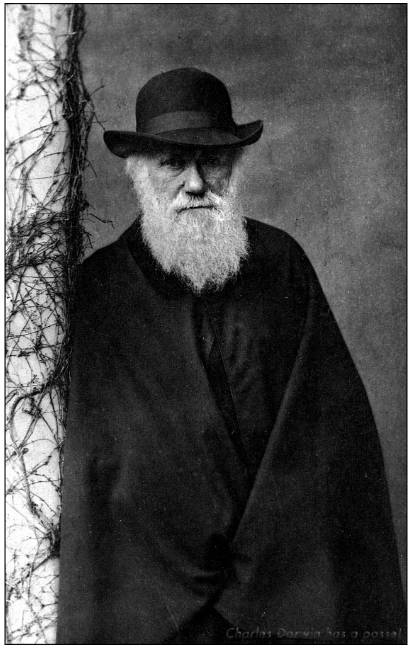
A deceptive view of evolution



What's wrong with this image?

Diversity generation





Charles Darwin, 1859
On the Origins of Species

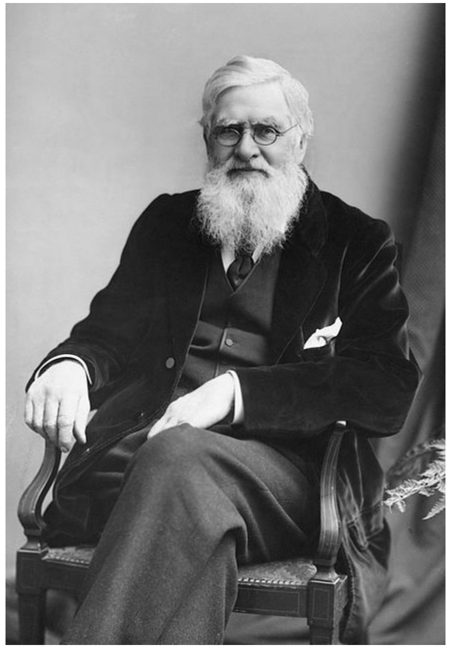
Four Pillars of Evolution

Population
Group of several individuals

Diversity
Individuals have different characteristics

Heredity
Characteristics are transmitted over generations

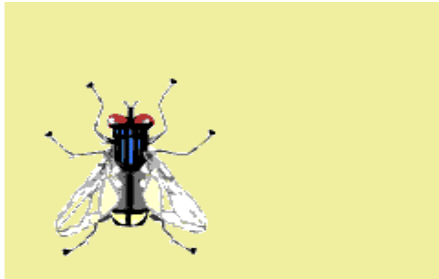
Selection
- Individuals make more offspring than the environment can support
- Comparatively better ones have higher probability of reproducing



Alfred Russel Wallace, 1858

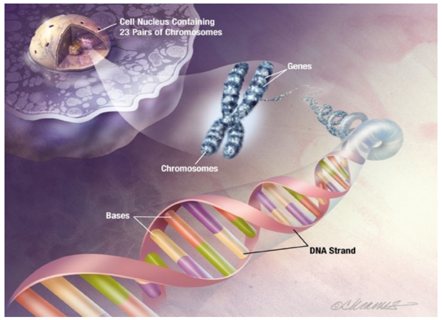
Phenotype & Genotype

Phenotype
The organism (physical instantiation, behavior, etc.)
Selection operates on phenotype
It is affected by environment, development, and learning



Gregor Mendel, 1858
Genetic basis of inheritance

Genotype
The genetic material of an organism.
Selection does not operate directly on genotype
It is affected only by mutations

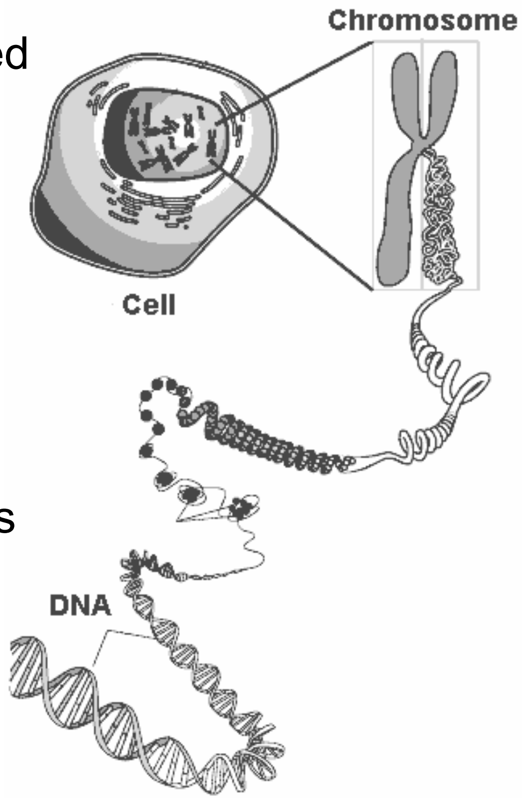


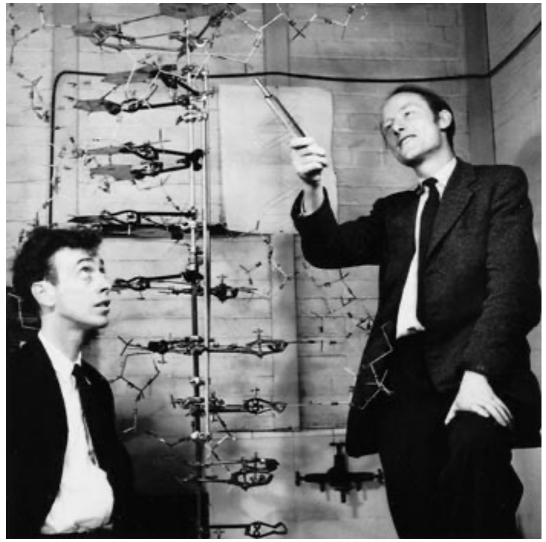
Huxley, 1940
Modern synthesis of evolution

DNA (DeoxyriboNucleic Acid)

Long molecule, twisted in spiral, and compressed

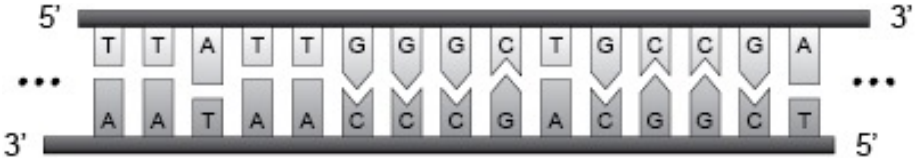
Humans have 23 pairs of DNA molecules (*chromosomes*)





Crick 1953 Watson
Discovery of DNA structure

DNA is composed of 2 complementary sequences (*strands*) of 4 nucleotides (A, T, C, G), which bind together in pairs (A-T and C-G)

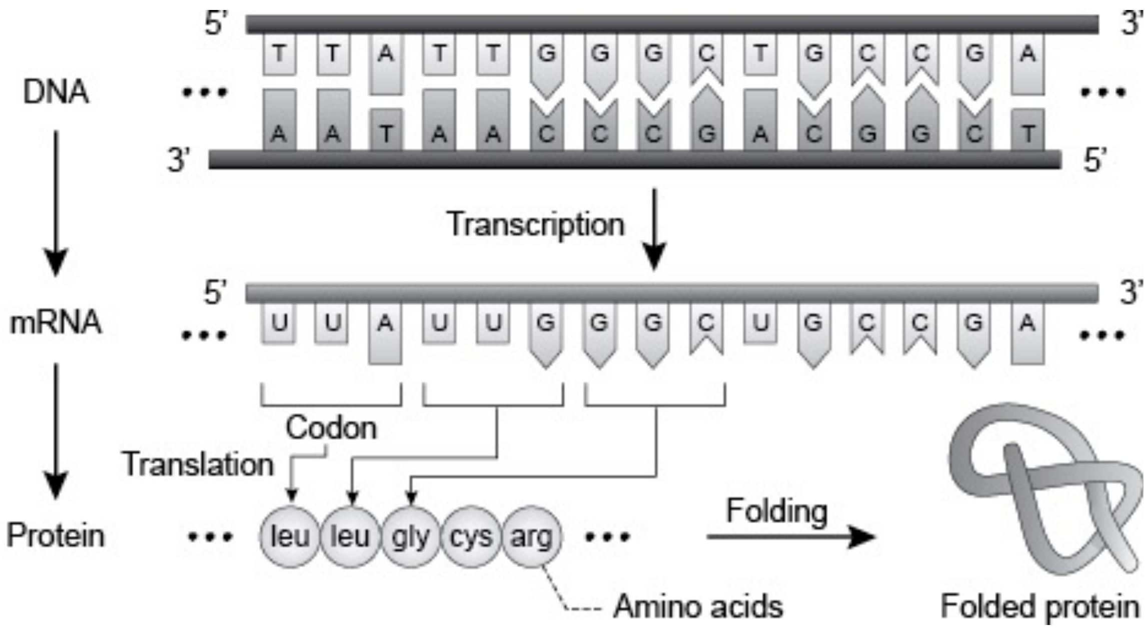


A **gene** is a sequence of several nucleotides that produce a protein

From Genes to Proteins *(gene expression)*

Proteins are folded molecule chains whose shape define the type and function of cells (some proteins affect gene expression)

The sequence of nucleotides in one strand defines the type of protein. The expression of the gene into a protein is mediated by another molecule, known as messenger RNA.

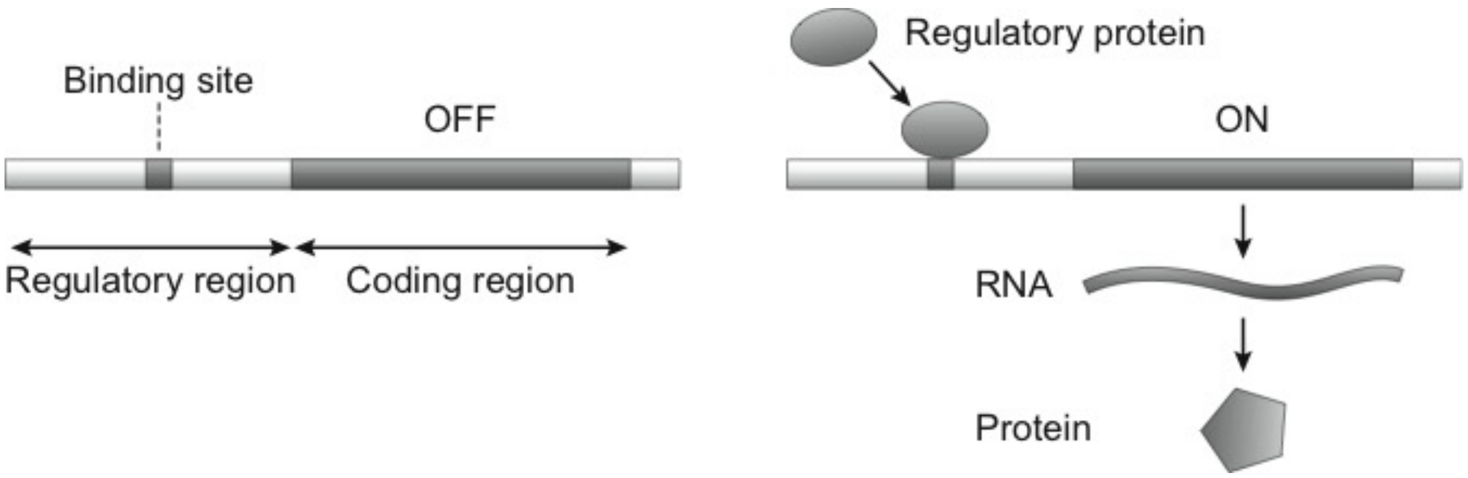


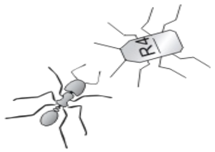
AlphaFold, 2021
Computer prediction of 3D protein folding

Regulation of gene expression

Genes are composed of a regulatory region and of a coding region.

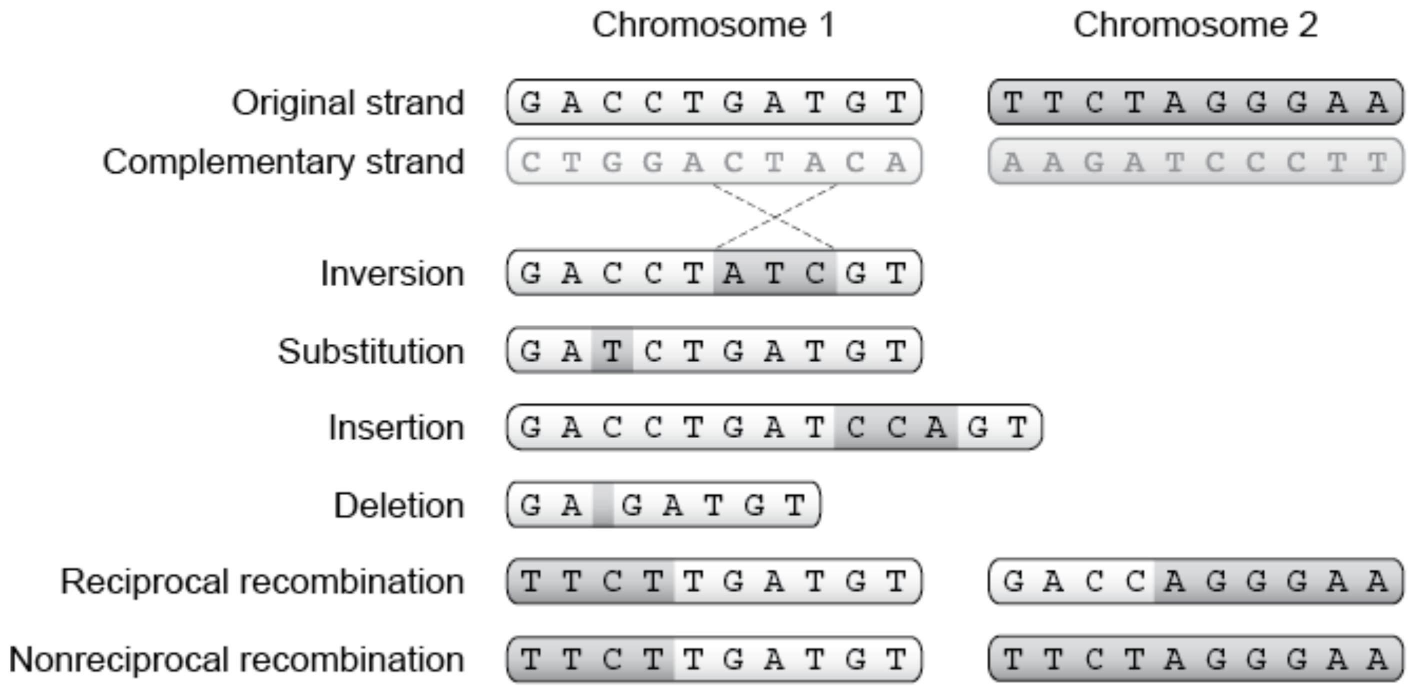
The coding region is translated into a protein if another protein binds onto the regulatory region.
Regulation can also be negative (i.e., inhibition of protein production).
Genome is a self-regulatory code





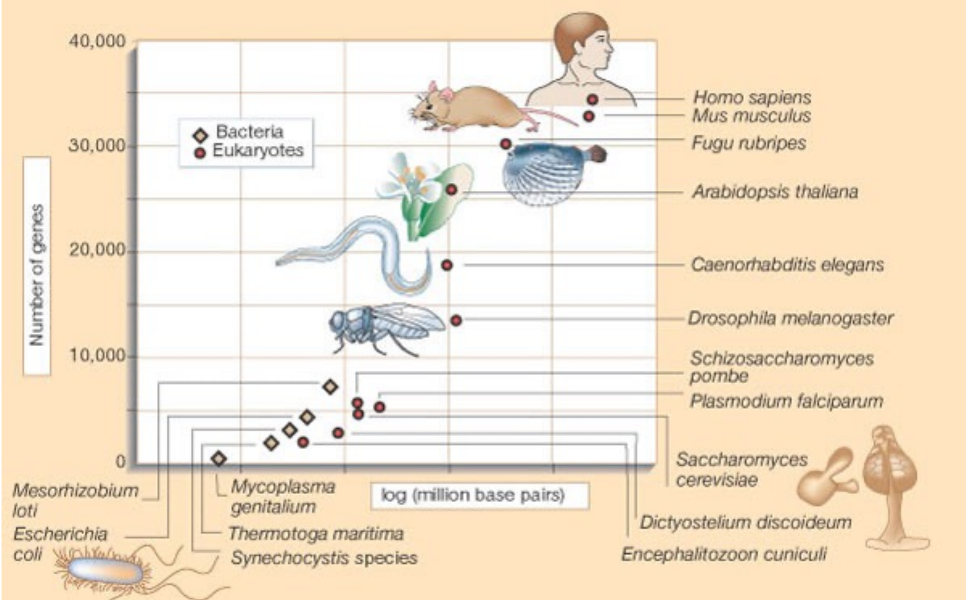
Genetic Mutations

- Genetic mutations occur during cell replication (4^{-10} per nucleotide per year)
- Those that occur in sex cells can affect evolution
- Sexual recombination is a mutation that affects two homologous chromosomes



Genome Size

Genome size within a species is constant (C-value, expressed in Mega bases), but it greatly varies across species www.genomesize.com for comparisons



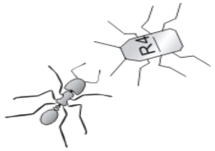
Doolittle, 2002

Genome size is not related to complexity of phenotype

Genome contains:
- **Genic** DNA
- **Nongenic** DNA

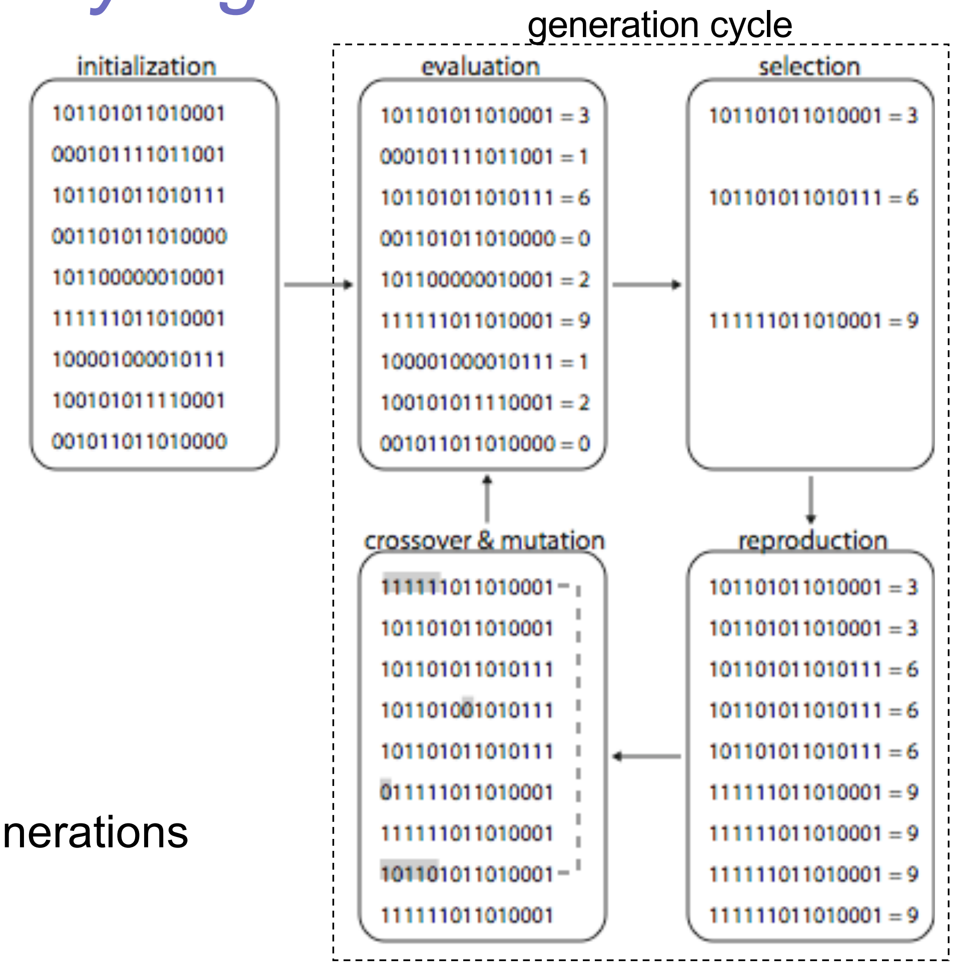
Nongenic DNA **arises** from:
- insertion/deletion mutations
- gene duplication

Nongenic DNA may have an **adaptive value**:
- pseudogenes may be re-activated
- pseudogenes may transform into new genes by several neutral mutations



Elements of an evolutionary algorithm

- Devise genetic representation
- Build a population
- Design a fitness function
- Choose selection method
- Choose crossover & mutation
- Choose data analysis method



<u>Repeat</u> generation cycle until:

- maximum fitness value is found
- solution found is good enough
- no fitness improvement for several generations

Artificial and Natural Evolution

Similarities between natural and artificial evolution:

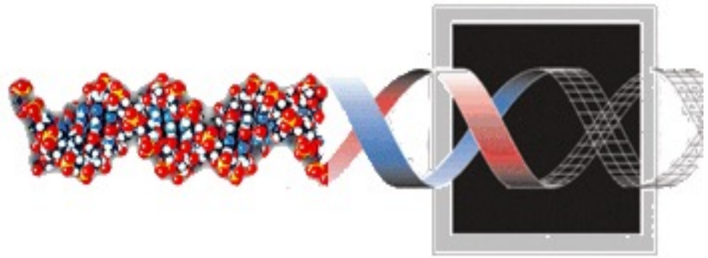
- Phenotype (computer program, object shape, electronic circuit, robot, etc.)
- Genotype (genetic representation of the phenotype)
- Population
- Diversity
- Selection
- Inheritance

Differences between natural and artificial evolution:

- Fitness is measure of performance of the individual solution to the problem
- Selection of the best according to performance criterion (fitness function)
- Expected improvement between initial and final solution

Genetic Representation

Coding of the phenotype (function variables, network weights, body parts, etc.) into a string



1001101010001

Simplification of biology:

- Single stranded sequence of elements
- Fixed length along generations, only genic
- Haploid structure and one chromosome
- Often one-to-one direct correspondence between genotype and phenotype

Types of representations:

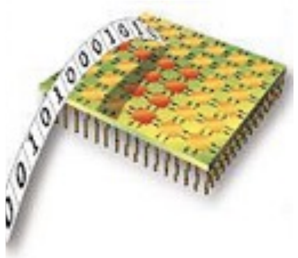
- Discrete
- Real-valued
- Sequence
- Tree-based

Discrete Representations

A sequence of *l* discrete values drawn from alphabet with cardinality *k*

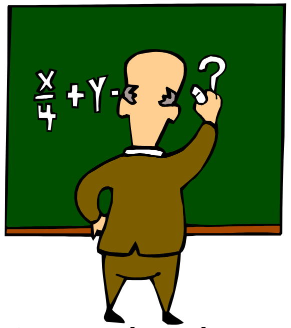
- E.g., binary string of 8 positions (l=8, k=2): 01010100
- Can be mapped into several phenotypes:

to configuration string of
FPGA electronic circuits



01010100

to integer *i* using
binary code

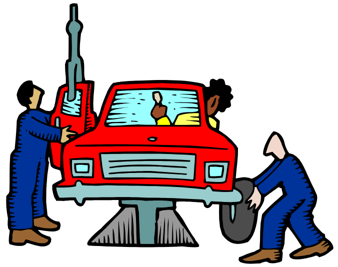


84

0.328125

to real value *r* in range [min, max]:
r = min + (i/255)(max-min)

Job	A.M.	P.M.
1	x	
2		x
3	x	
4		x
5	x	
6		x
7	x	
8	x	



to job schedule:

- job=gene position
- time=gene value

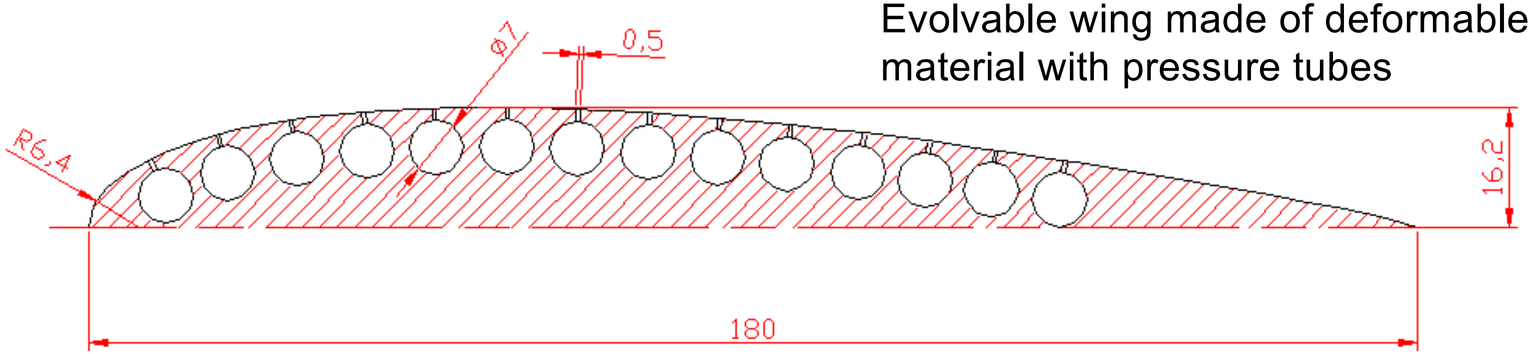
Real-Valued Representation

A vector of real values that represent function parameters

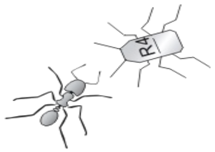
- Used when high-precision parameter optimization is required
  - Variables of multi-dimensional function to be optimized
  - Connection weights of neural networks
  - Parameters of experiment

- Example: representation of wing profile for shape optimization

Evolvable wing made of deformable material with pressure tubes

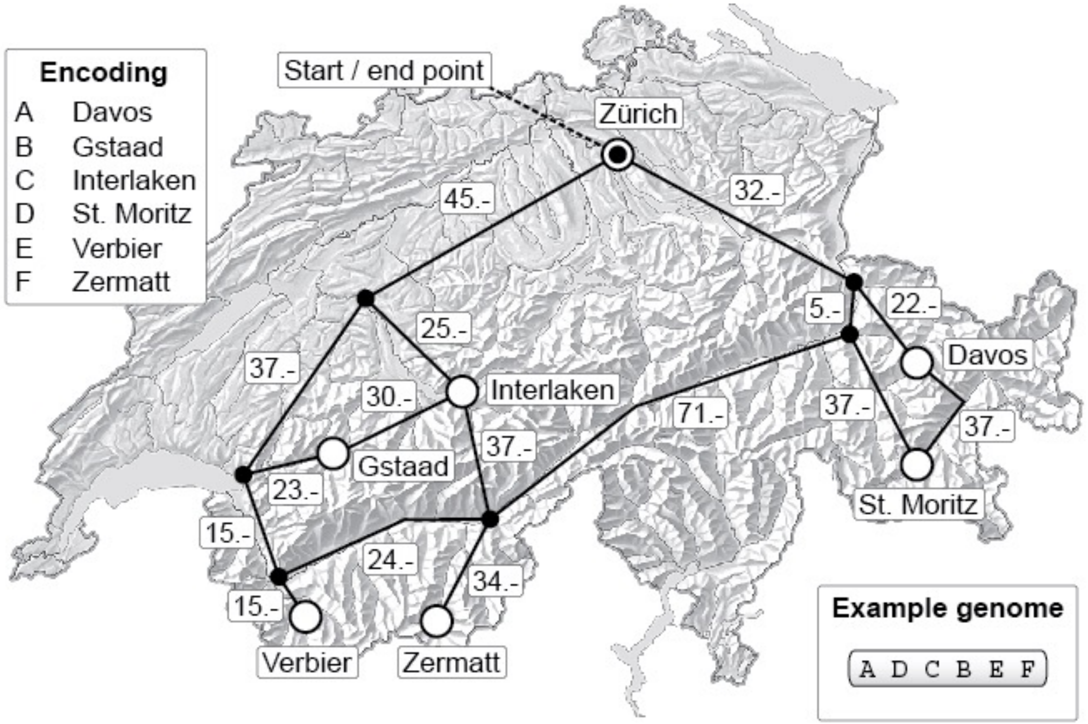


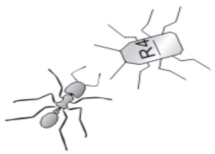
<u>Genotype</u>= pressure values of 14 tubes



Sequence Representation

A specific case of discrete representations used for Traveling Salesman Problems (plan a path to visit n cities under some constraints). E.g., planning ski holidays with lowest transportation costs





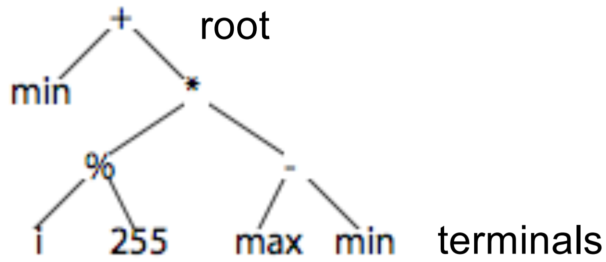
Tree-based Representation

A nested list describing a tree with branching points and terminals
Suitable for encoding hierarchical structures, such as computer programs and robot body plans

For example, a computer program is made of:
- Function set: multiplication, If-Then, Log, etc.
- Terminal set: constants, variables, sensor readings, etc.

Expression r=min+(i/255)(max-min)

Nested list (+, min, (*, (/, i, 255), (-, max, min)))



- <u>Closure</u>: all functions must accept all terminals in Terminal set and outputs of all functions in Function set (e.g., protected division %)
- <u>Sufficiency</u>: elements in Function and Terminal sets must be sufficient to generate program that solves the problem

Build Initial Population

How large? Sufficiently large to cover problem space (!), but sufficiently small for evaluation costs (typical size found in the literature: between 10s and 1000s individuals)

Uniform sample of search space:

- Binary strings: 0 or 1 with probability 0.5
- Real-valued representations: uniform on a given interval if bounded phenotype (e.g., +2.0, -2.0); otherwise best guess
- Sequence: position all elements at random locations of each string
- Trees are built recursively starting from root:
  - root is randomly picked from function set
  - set maximum depth of tree
  - for every branch, randomly pick from all elements of function set and of terminal set
  - if a terminal is picked, it becomes a leaf (end of the branch)

Clone and mutate previously evolved genotype or hand-designed genotype; possible dangers:

–Small genetic diversity

–Unrecoverable bias

Fitness Function

Evaluates **performance** of phenotype with a numerical score

- Choice of components; e.g., lift and drag of wing
- Combination of components; e.g. (lift + 1/drag) or (lift - drag)
- Extensive test of each phenotype
- Warning! You Get What You Evaluate (example in application, later)

Subjective fitness: select phenotype by visual inspection

- Used when aesthetic properties cannot be quantified objectively
- Can be combined with objective fitness function

"A-Volve", Sommerer and Mignonneau,
NTT ICC Tokyo Opera House, www.ntticc.or.jp

Selection

A method to make sure that better individuals make comparatively more offspring

Some methods:

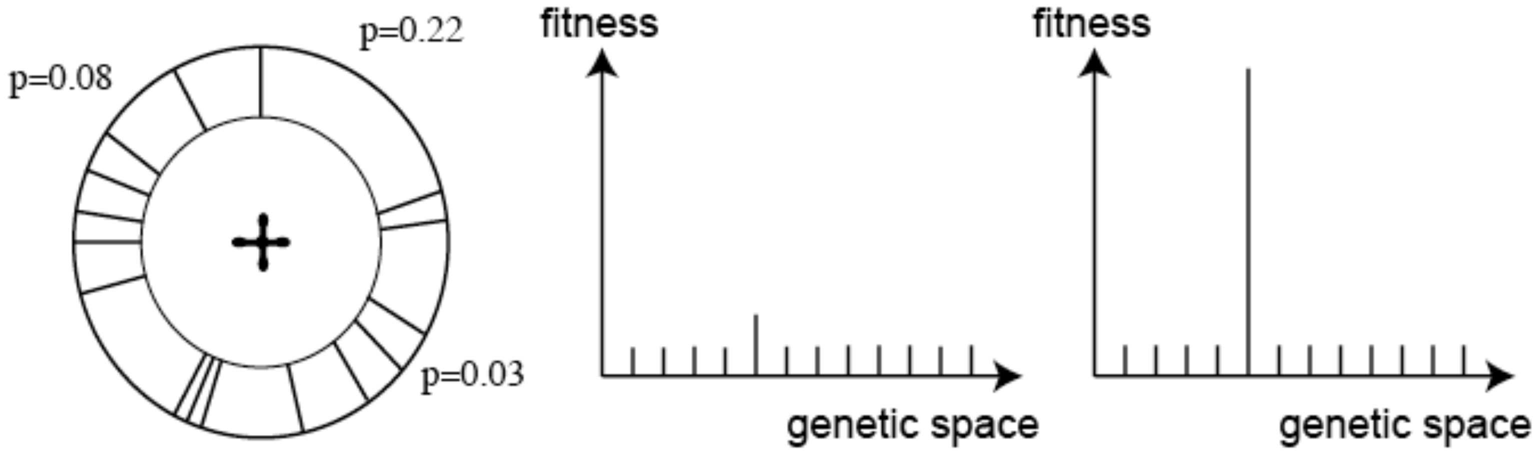
- Proportionate selection
- Rank-based selection
- Truncated rank-based selection
- Tournament selection

- Selection pressure is inversely proportional to percentage of selected individuals
- High selection pressure = rapid loss of diversity and premature convergence
- Make sure that also less performing individuals have a chance to reproduce

Proportionate Selection

The probability that an individual makes an offspring is proportional to how good its fitness is with respect to the population fitness: **$p(i) = f(i)/\Sigma f(i)$**

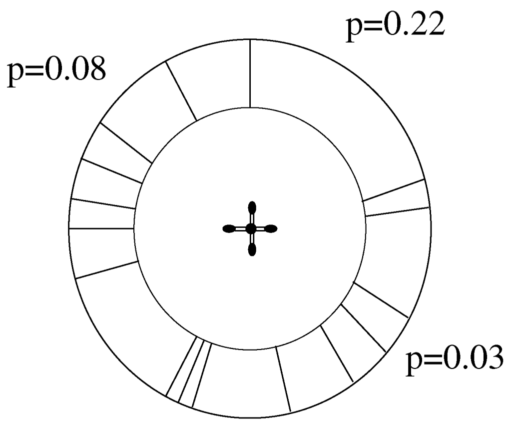
Also known as Roulette Wheel selection



Problems:
Uniform fitness values = random search
Few high-fitness individuals = high selection pressure

Rank-based Selection

- Individuals are sorted on their fitness value from best to worse. The place in this sorted list is called the rank r.
- Instead of using the fitness value of an individual, the rank is used to select individuals: **p(i) = 1 - r(i)/Σr(i)**
- Use roulette wheel



individual	fitness	rank
A	5	5
B	7	3
C	8	2
D	2	8
E	3	7
F	9	1
G	7	4
H	4	6

Truncated Rank-based Selection

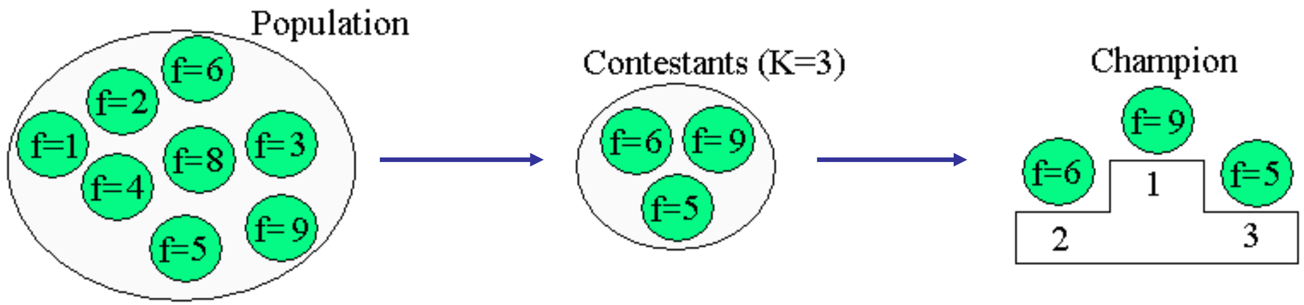
- Only the best **x** individuals are allowed to make offspring and each of them makes the same number of offspring: **N/x**, where **N** is the population size.
- E.g., in population of 100 individuals, make 5 copies of 20 best individuals

individual	fitness	rank	list
A	5	5	F
B	7	3	C
C	8	2	B
D	2	8	G
E	3	7	A
F	9	1	H
G	7	4	E
H	4	6	D

Tournament Selection

For every offspring to be generated:

- Pick randomly ***k*** individuals from the population
- Choose the individual with the highest fitness and make a copy
- Put all individuals back in the population

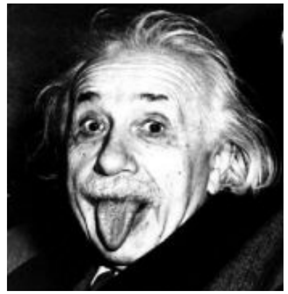


k is the tournament size (larger size = larger selection pressure)

Generational Replacement

All individuals are replaced by their offspring in the new generation

Population size: constant over generations



Problem: mutations or poor fitness assessment may lead to loss of good individuals

Elitism: insert ***n*** best individuals from previous generation and randomly remove **n** individuals

Generational rollover

Generate and insert one offspring at a time in the population and let it compete with other individuals

Methods to maintain population size

Remove individual at random

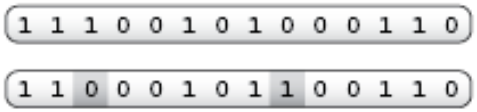
Remove oldest individual

Remove worst individual

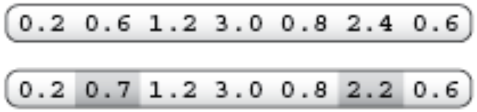
Mutation

Applied to each gene in the genetic string with probability p_m

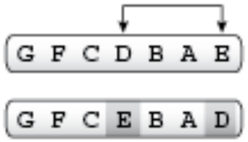
Binary genotypes



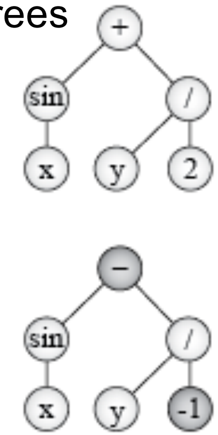
Real-valued genotypes
(uniform mutation)



Sequence genotypes

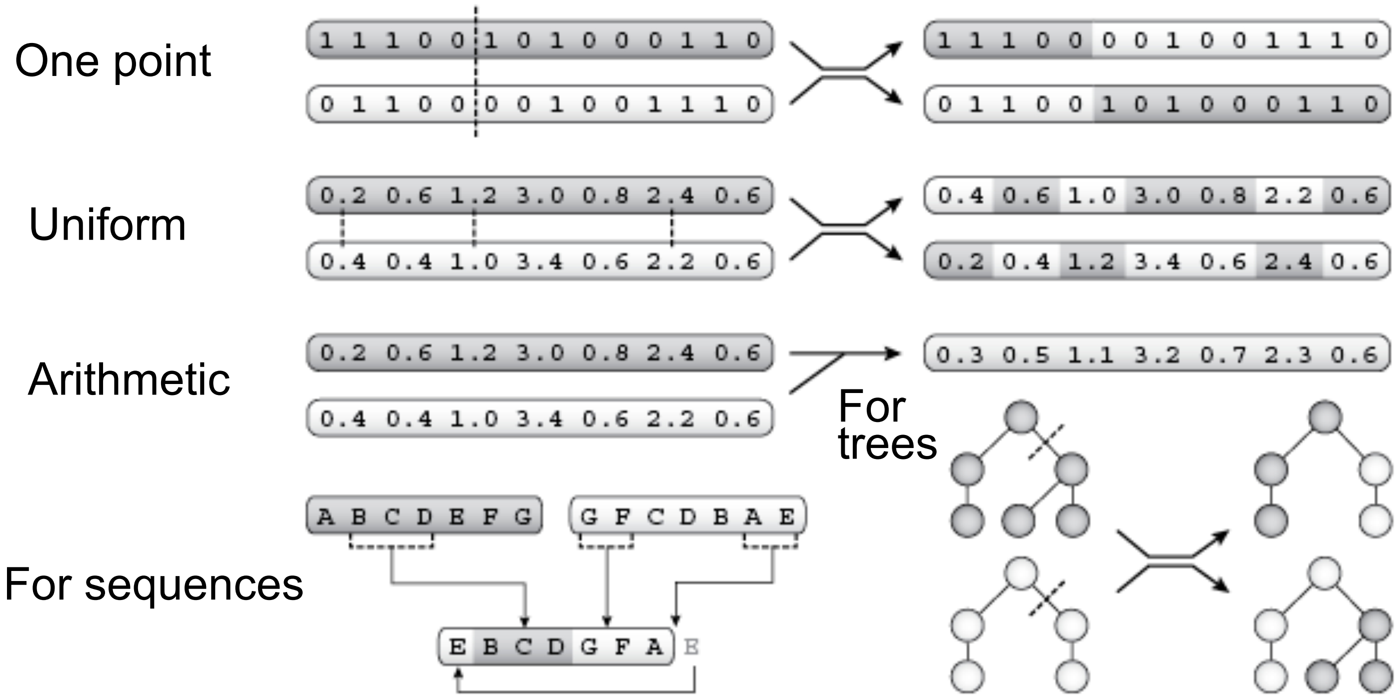


For trees



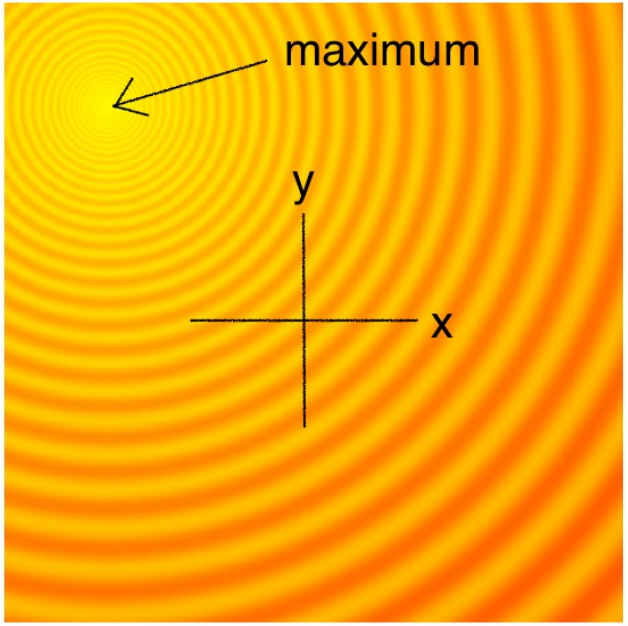
Crossover

Emulates recombination of genetic material from two parents during meiosis
Exploitation of synergy of sub-solutions (building blocks) from parents
Applied to randomly paired offspring with probability p_c(pair)

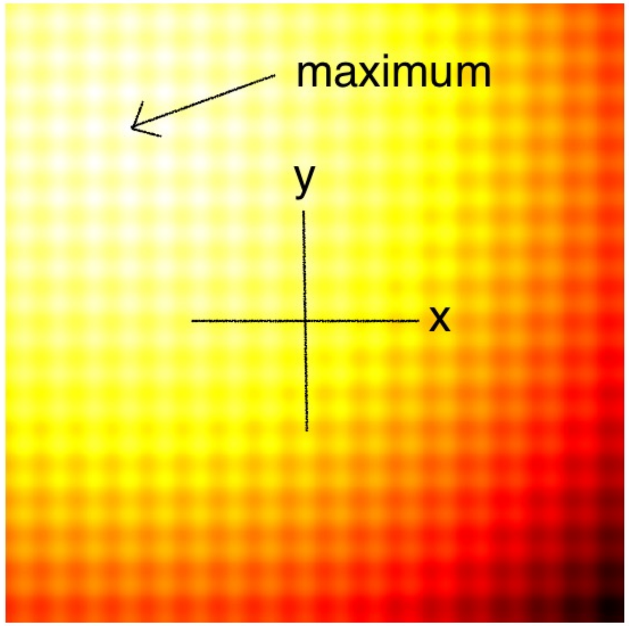


Artificial landscapes

Shifted Schaffer-2D function



Shifted Rastrigin-2D function

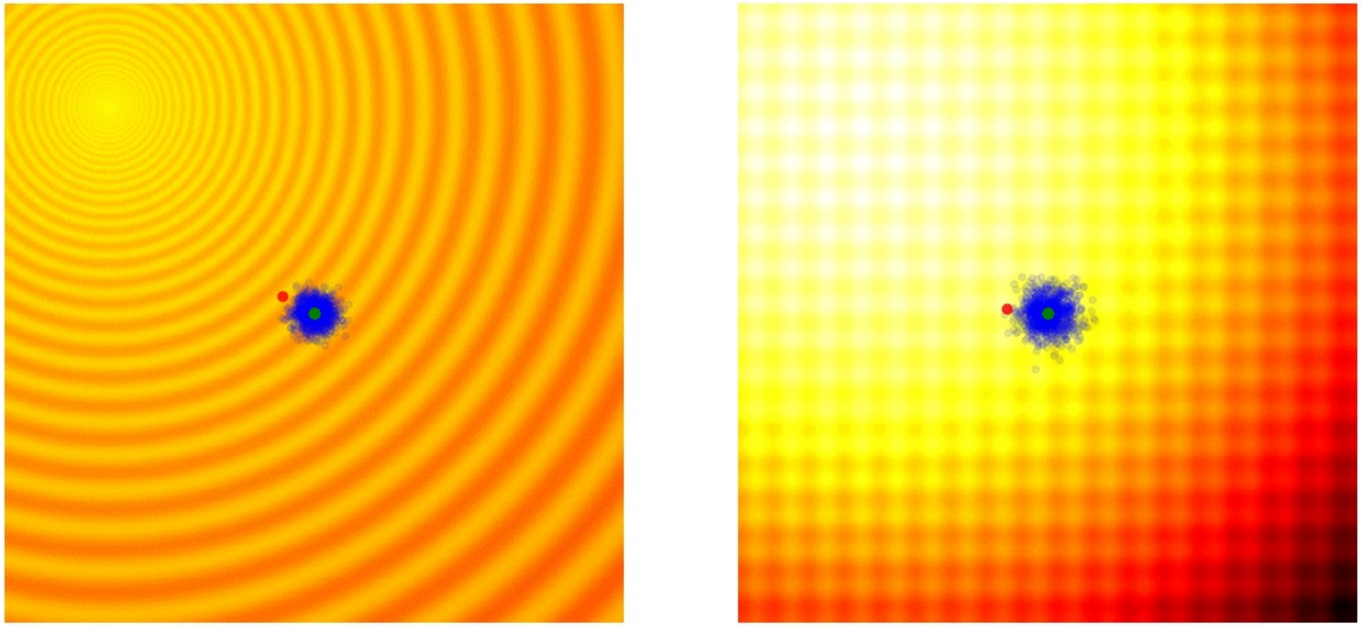


Goal: find a set of *parameters* (*x*,*y*),
such that *F*(*x*,*y*) is as close as possible to the global maximum

More test functions: https://en.wikipedia.org/wiki/Test_functions_for_optimization

Source: http://blog.otoro.net/2017/10/29/visual-evolution-strategies/

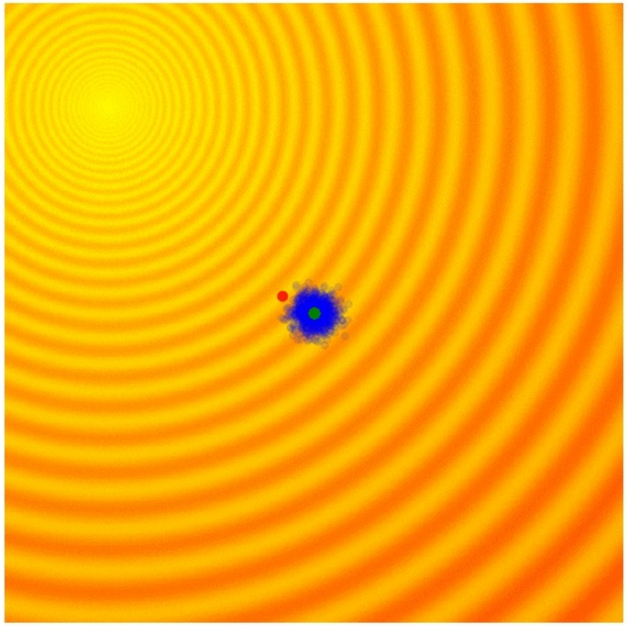
A Simple Evolutionary Algorithm

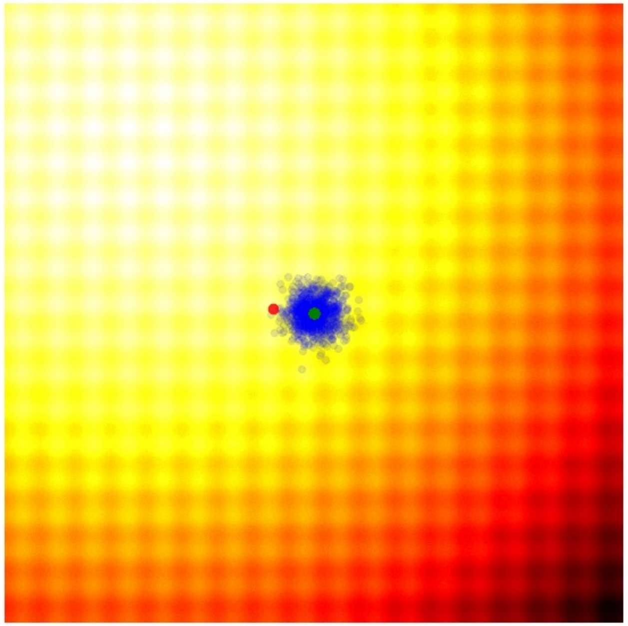


1. Sample initial population from Normal distribution, with mean $\mu=(\mu x,\mu y)$ and standard deviation $\sigma=(\sigma x,\sigma y)$ set at the axis origin
2. Select best 10% and make copies to create new population
3. Crossover and mutate by adding Gaussian noise with fixed σ
4. Repeat steps 2&3 until satisfactory solution is found

Source: http://blog.otoro.net/2017/10/29/visual-evolution-strategies/

20 generations



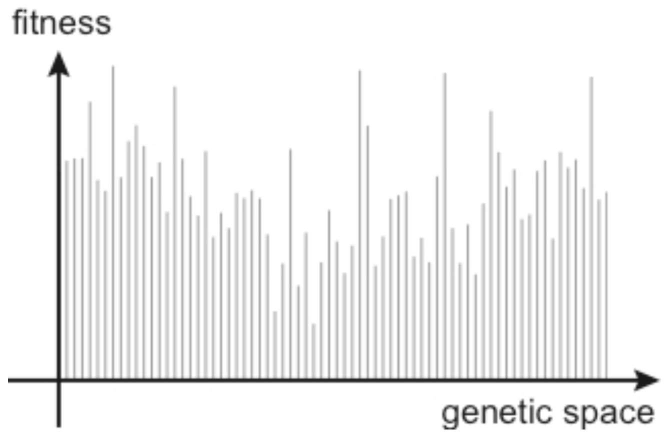


Blue dots show the individuals of the current generation
Red dot shows the best individual of the current generation
Green dots show the selected parents of the previous generation

Source: http://blog.otoro.net/2017/10/29/visual-evolution-strategies/

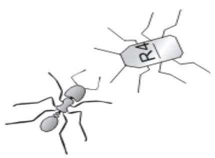
Measuring evolvability: Fitness Landscape

Fitness landscape is a theoretical plot of fitness values associated to all genotypes
Landscape ruggedness helps identify population size, selection pressure, mutation
Notion of landscape can be deceptive



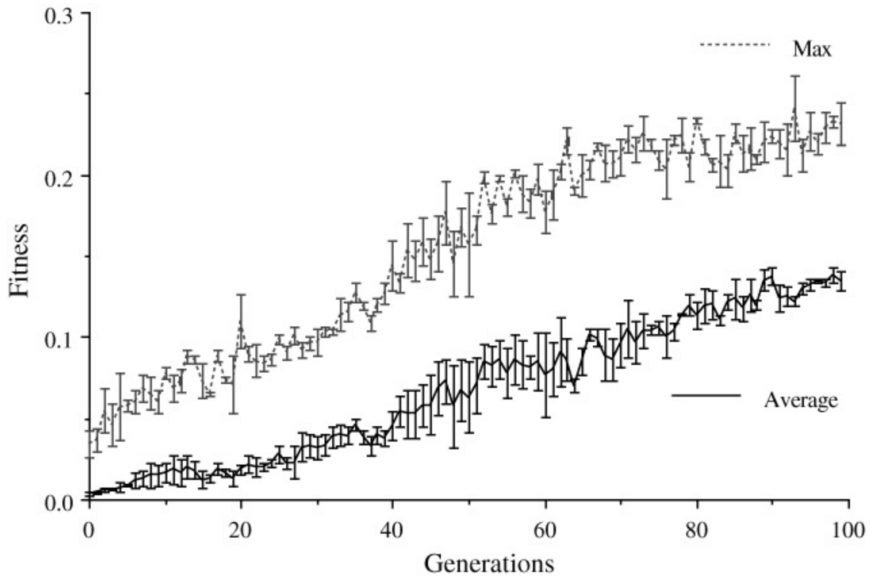
Estimating ruggedness of real landscape:

• Sample random genotypes: if flat, use large populations
• Explore surroundings of individual by applying genetic operators in sequence for fixed number of times: the larger the fitness improvement, the smaller the population size



Measuring Performance: Fitness Graph

Track best and population average fitness of each generation
Multiple runs are necessary: plot average data and standard error

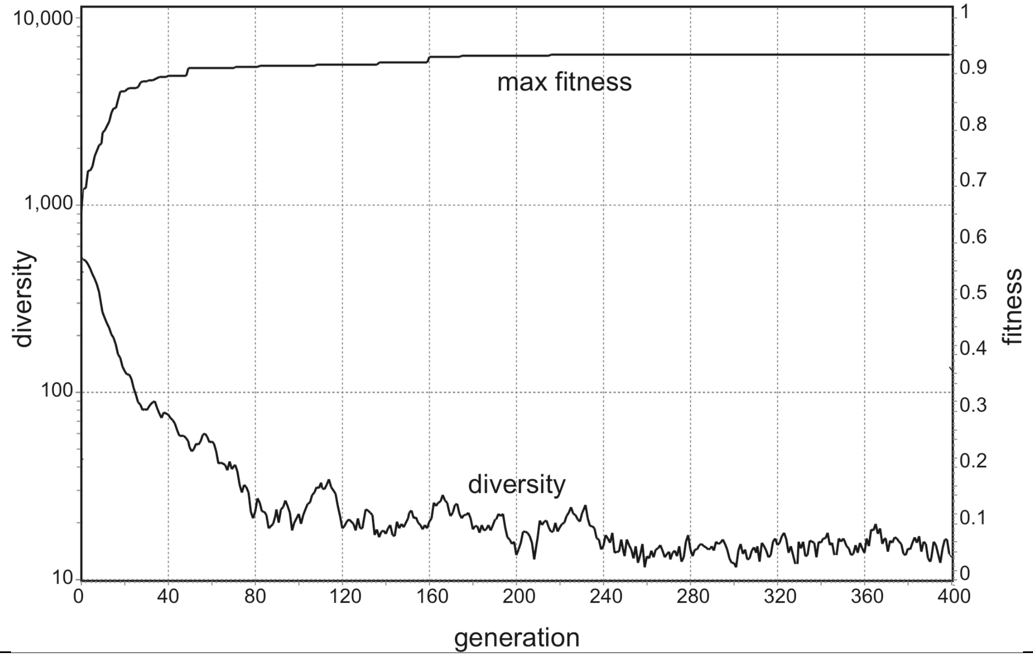


- Fitness graphs are meaningful only if the problem is stationary
- Stagnation of fitness function may mean best solution found or premature convergence

Measuring Diversity: Genotype distance

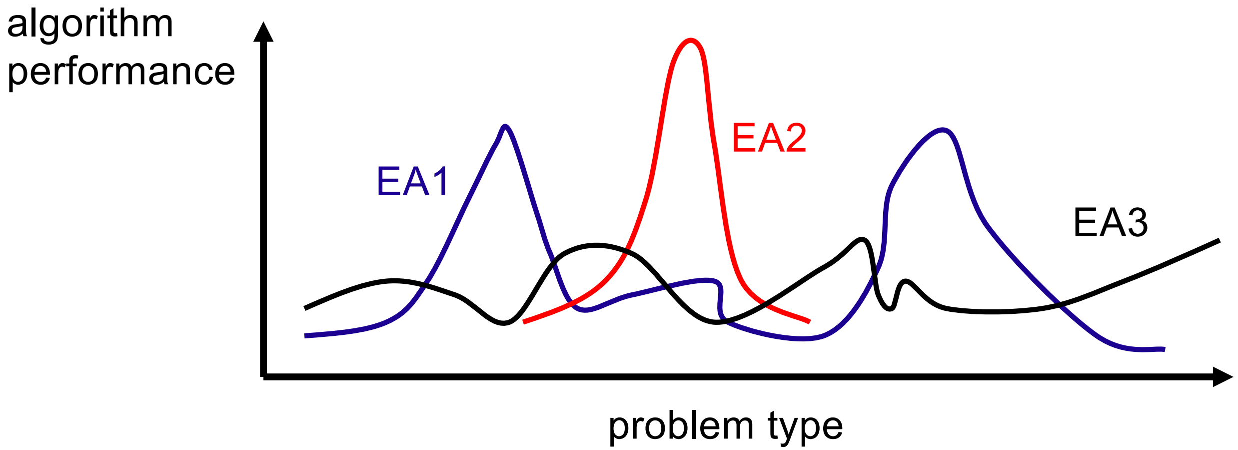
Diversity tells whether the population has potential for further evolution
Measures of diversity depend on genetic representation
E.g., for binary and real valued, use sum of Euclidean or Hamming distances

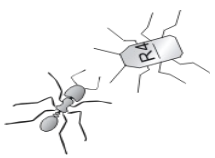
$$D_a(P) = \sum_{i,j \in P} d(g_i, g_j)$$



Applicability

- Evolutionary algorithms can be used for any problem
- Different problems may require different algorithms
- Knowledge of problem domain can help to choose best algorithm





Examples of Evolutionary Algorithms

- **Genetic Algorithms** (GA) - Holland, 1975

Binary genotypes, crossover and mutation

- **Genetic Programming** (GP) - Koza, 1992

Tree-based genotypes, crossover and mutations

- **Steady-State GA** (SSGA) – Whitley et al., 1988

Gradual replacement: Best individuals replace replace worst individuals

- **Differential Evolution** (DE) – Storn & Prince, 1996

As SSGA, but with differential factor

- **Evolutionary Strategies** (ES) - Rechenberg, 1973

Real-valued genotypes, mutation step(s) encoded in genotype

- **Covariance Matrix Adaptation ES** (CMA-ES) – Hansen & Ostermeier, 2001

Evolutionary Strategies with correlated and adaptive mutations

- **Non-dominated Sorting GA** (NSGA)– Srinivas, Deb, 1998

Multi-objective evolutionary optimization

- **Viability Evolution** (ViE)– Maesani, Mattiussi, Floreano, 2014

Evolution without fitness ranking and diversity preservation

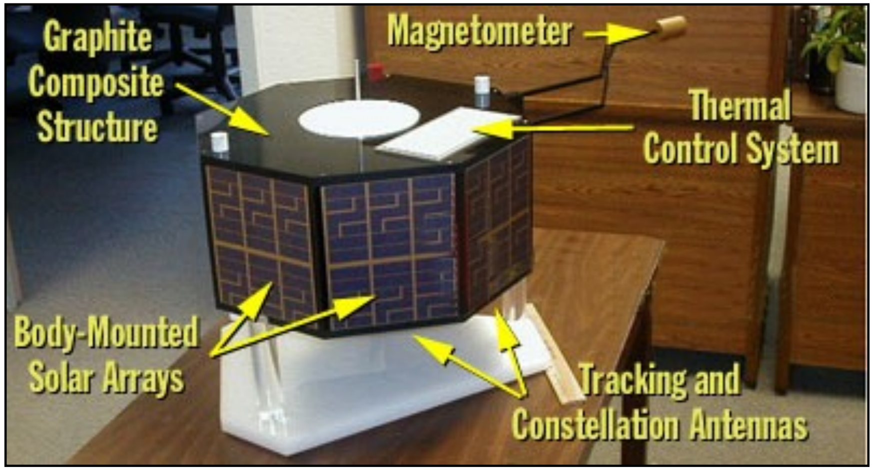
- **MAP Elites** – Mouret and Clune, 2015

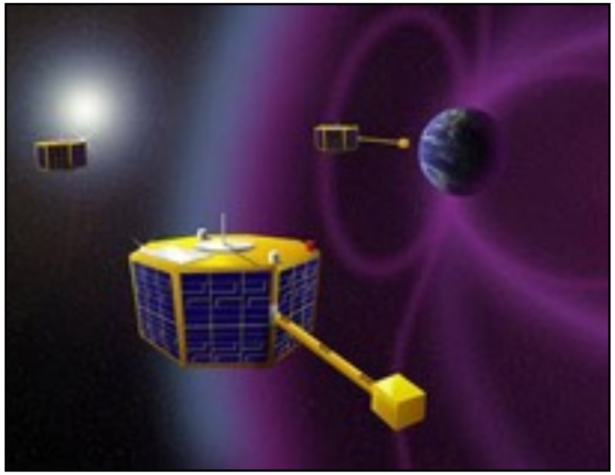
Preserve diversity by making similar solutions compete with each other

Antenna for Nanosatellites

ST5 mission: Measure effect of solar activity on the Earth's magnetosphere
3 nanosatellites (50 cm)
Design of antenna to send data to ground station

[Lohn, Hornby, Linden, 2004]

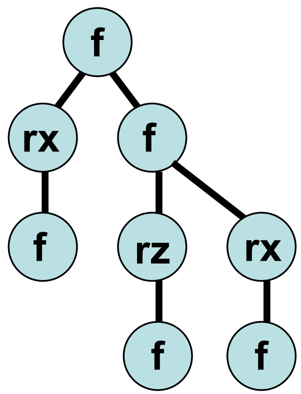


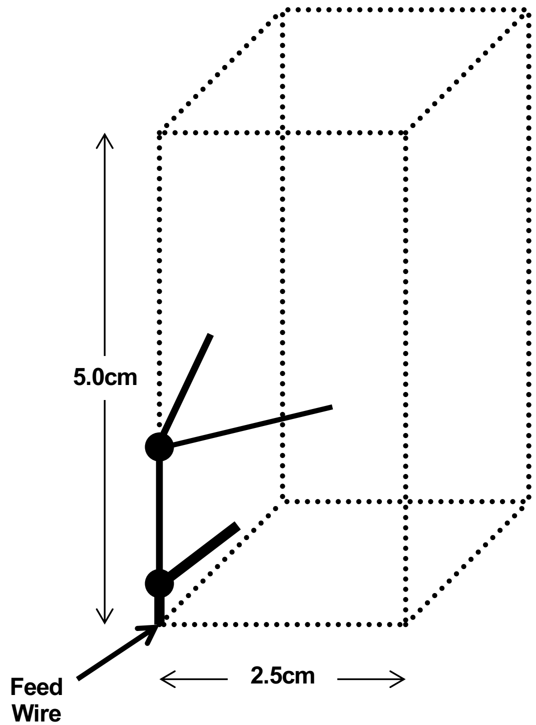


Genetic Representation of antennas

Tree-based Encoding with Genetic Programming
Evaluate fitness in simulation
Build best and test in anechoic chamber

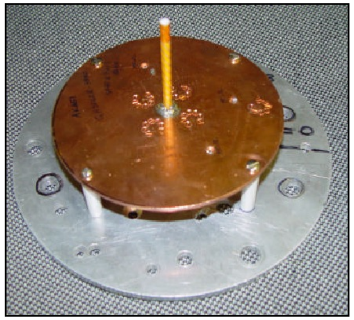
Function Set
f = forward (length, radius)
r x/y/z = rotate x/y/z
Terminal Set
Length, radius, x, y, z
Constraint:
max. 3 branches for each f node





Comparison human/evolved

Human



Evolved