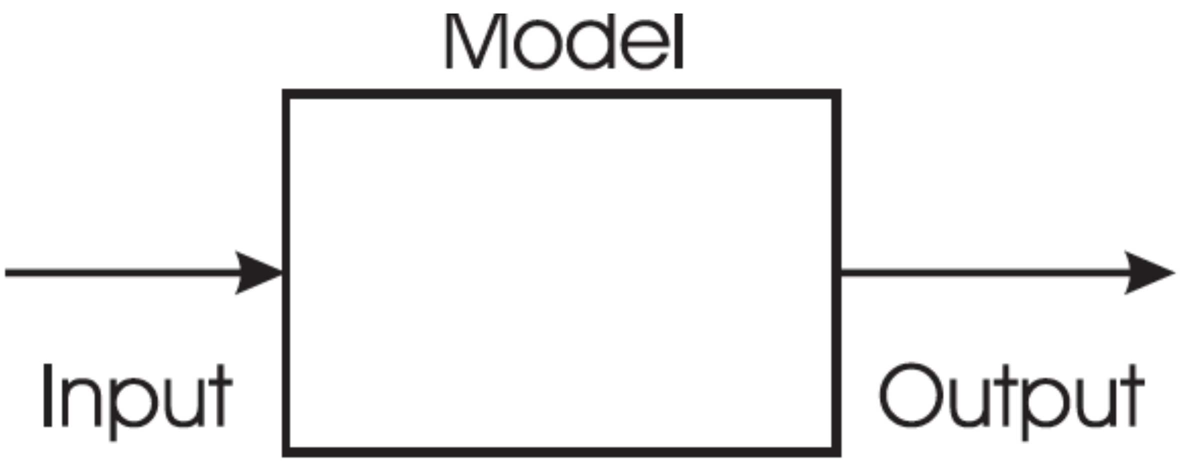
Evolutionary Systems

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

Black-Box Problems

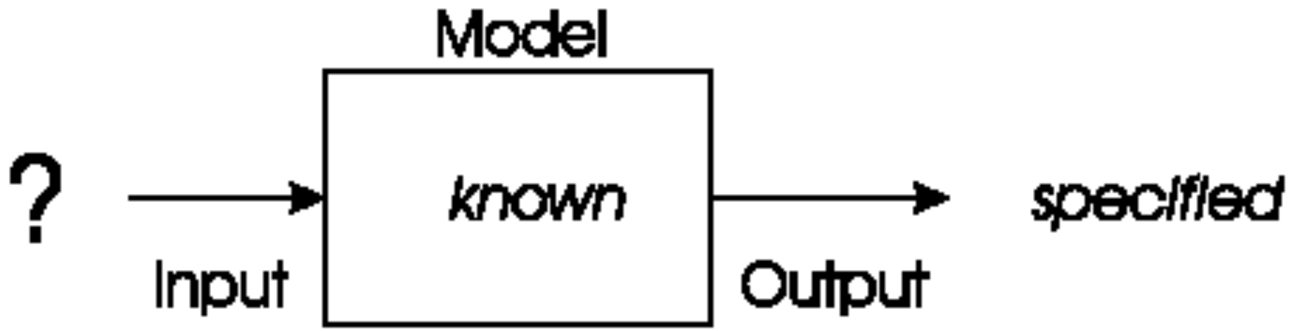


- “Black box” problems consists of 3 components
- When one component is unknown: new problem type

Adapted from http://www.evolutionarycomputation.org/slides/

Black-Box Problem: Optimisation

Model and desired output is known, task is to find inputs

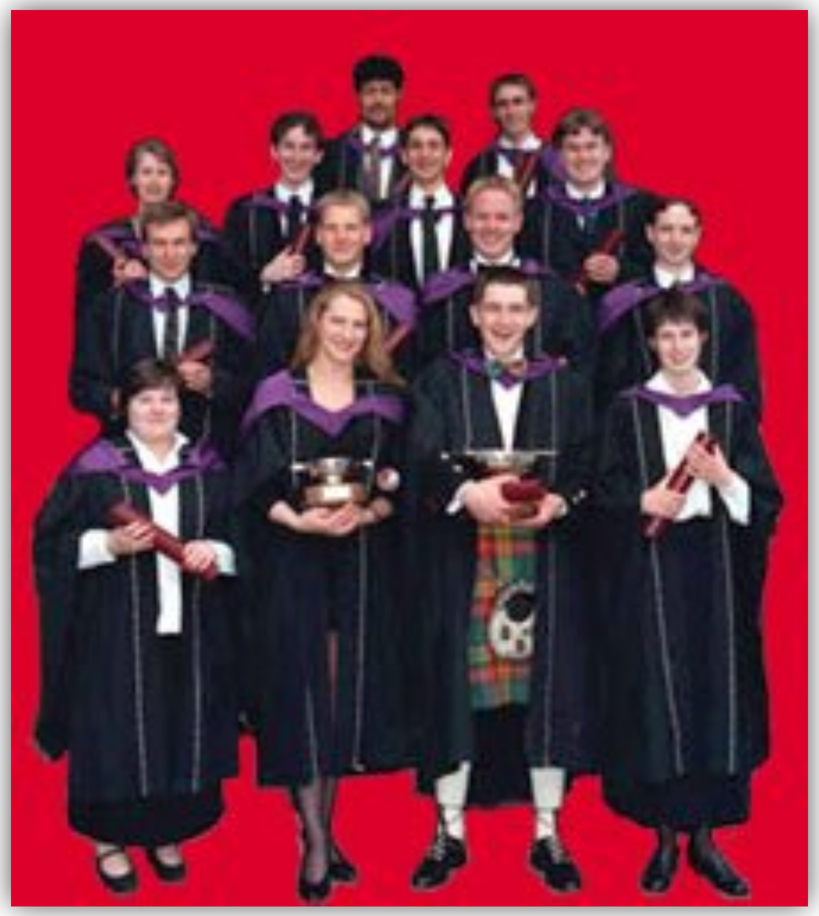


- Time tables for university, call center, or hospital
- Traveling salesman problem (TSP)
- Eight-queens problem, etc.

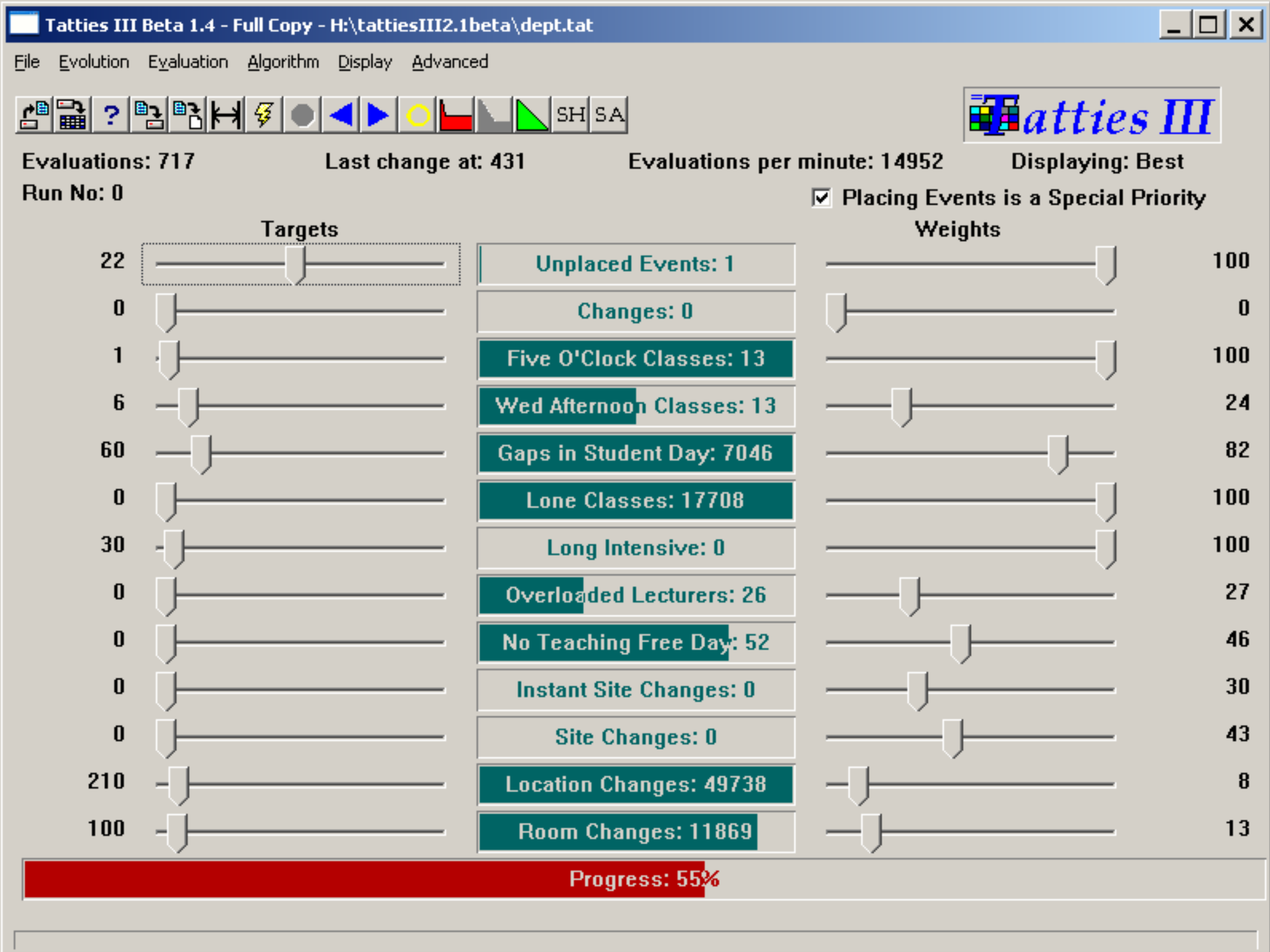
Adapted from http://www.evolutionarycomputation.org/slides/

Optimisation problem: University timetables

- Enormously big search space
- Timetables must be *good*
- "Good" is defined by a number of competing criteria
- Timetables must be feasible
- Vast majority of search space is infeasible

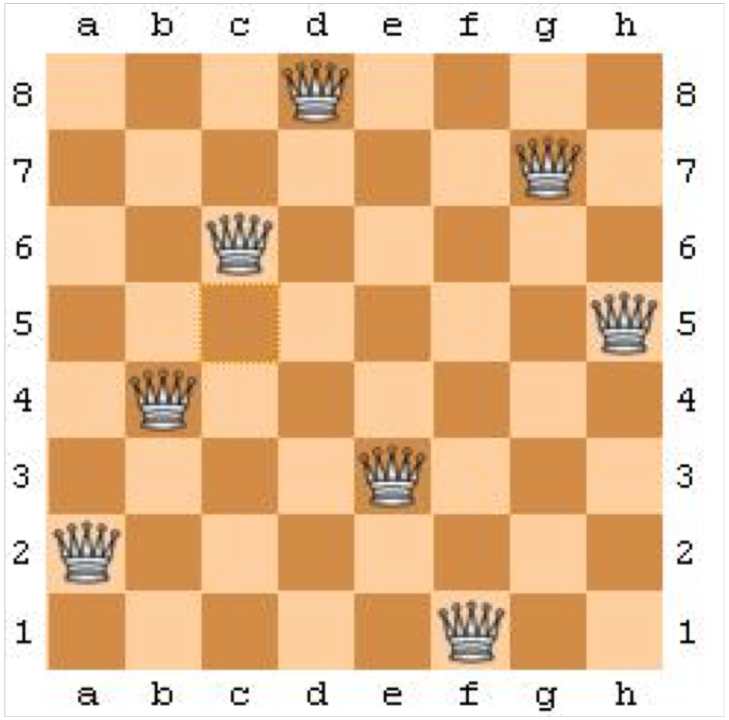


Adapted from http://www.evolutionarycomputation.org/slides/


Tatties III Beta 1.4 - Full Copy - H:\tattiesIII2.1beta\dept.tat
File Evolution Evaluation Algorithm Display Advanced
SH SA
Tatties III
Evaluations: 717
Last change at: 431
Evaluations per minute: 14952
Displaying: Best
Run No: 0
Placing Events is a Special Priority
Targets
Weights
22 Unplaced Events: 1 100
0 Changes: 0 0
1 Five O'Clock Classes: 13 100
6 Wed Afternoon Classes: 13 24
60 Gaps in Student Day: 7046 82
0 Lone Classes: 17708 100
30 Long Intensive: 0 100
0 Overloaded Lecturers: 26 27
0 No Teaching Free Day: 52 46
0 Instant Site Changes: 0 30
0 Site Changes: 0 43
210 Location Changes: 49738 8
100 Room Changes: 11869 13
Progress: 55%

Optimisation problem: Queen-positioning

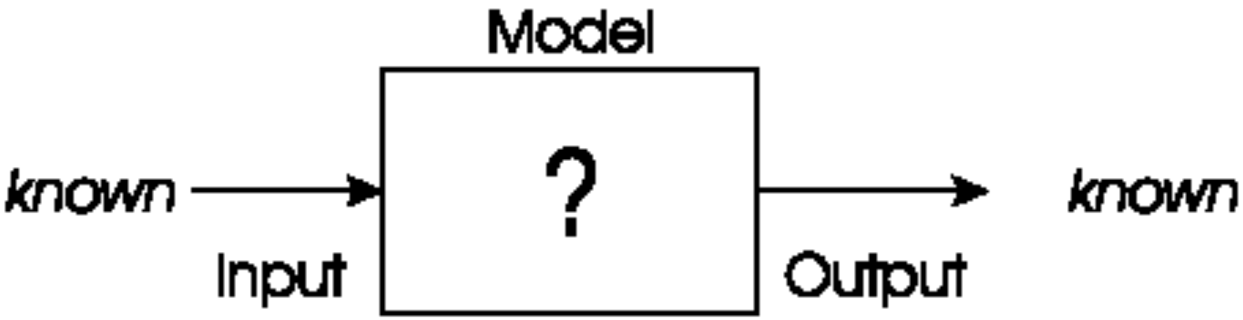
- Given an 8-by-8 chessboard and 8 queens
- Place the 8 queens on the chessboard without any conflict
- Two queens conflict if they share same row, column or diagonal
- Can be extended to an n queens problem (n>8)



Adapted from http://www.evolutionarycomputation.org/slides/

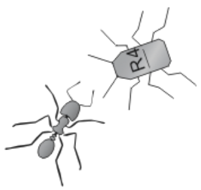
Black-Box Problem: Modelling

We have corresponding sets of inputs & outputs and seek a model that delivers correct output for every known input



- Evolutionary machine learning
- Predicting Customer Behavior
- Predicting stock exchange

Adapted from http://www.evolutionarycomputation.org/slides/



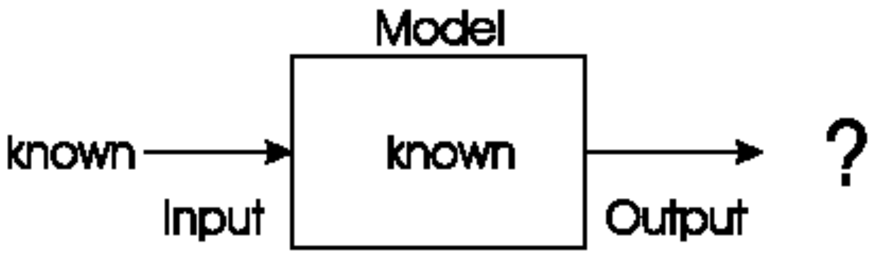
Modelling problem: Loan Paying Behaviour

- British bank evolved creditability model to predict loan paying behavior of new applicants
- Evolving: prediction models
- Fitness: model accuracy on historical data

Adapted from http://www.evolutionarycomputation.org/slides/

Black-Box Problem: Simulation

We have a given model and wish to know the outputs that arise under different input conditions



Often used to answer "what-if" questions in evolving dynamic environments

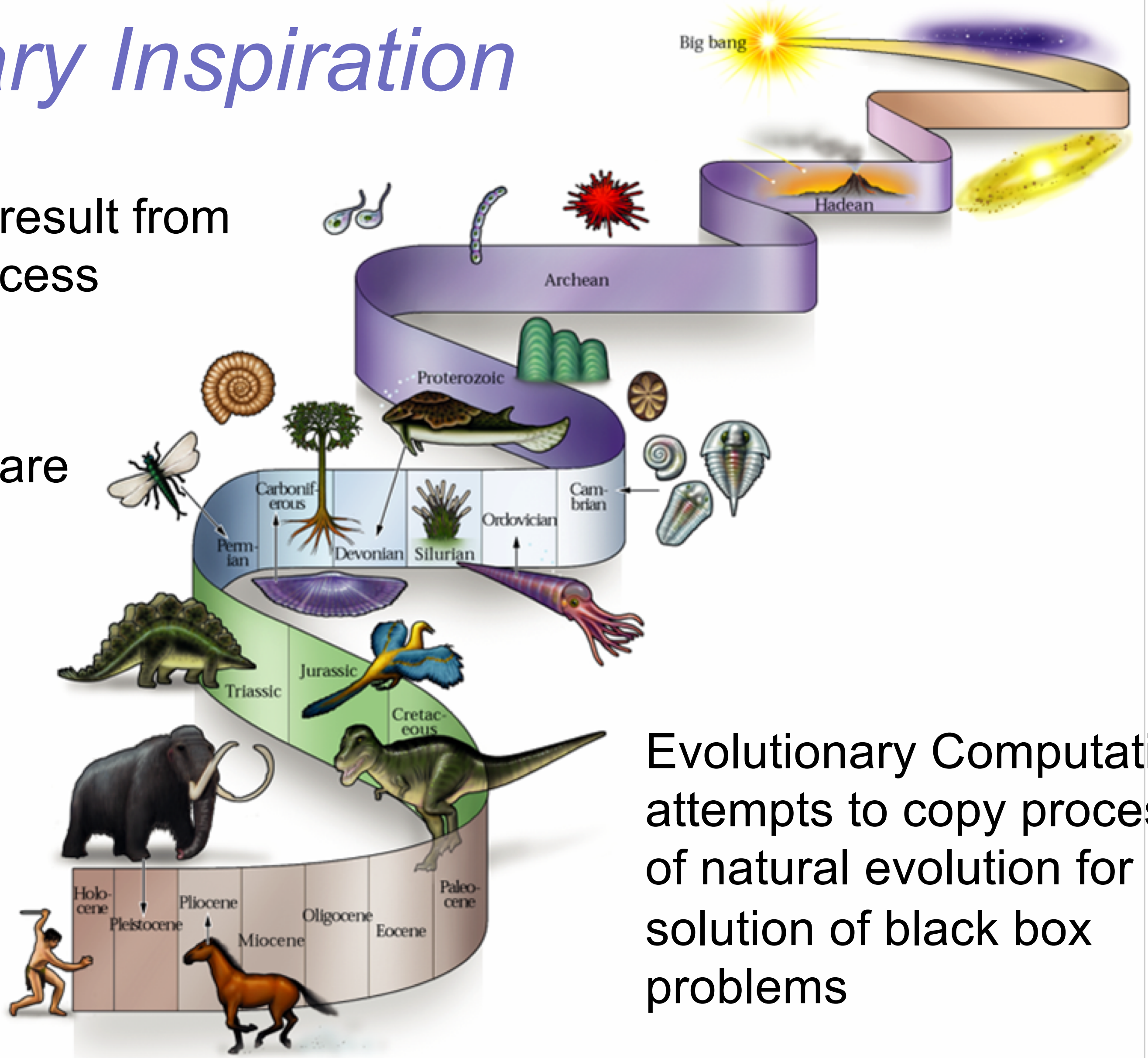
- Evolutionary economics, Artificial Life
- Weather forecast system
- Impact analysis new tax systems

Adapted from http://www.evolutionarycomputation.org/slides/

Evolutionary Inspiration

Biological systems result from an evolutionary process

Biological systems are
- robust
- complex
- adaptive



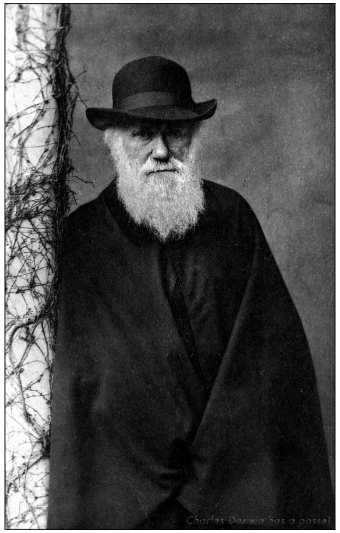
Evolutionary Computation attempts to copy process of natural evolution for solution of black box problems

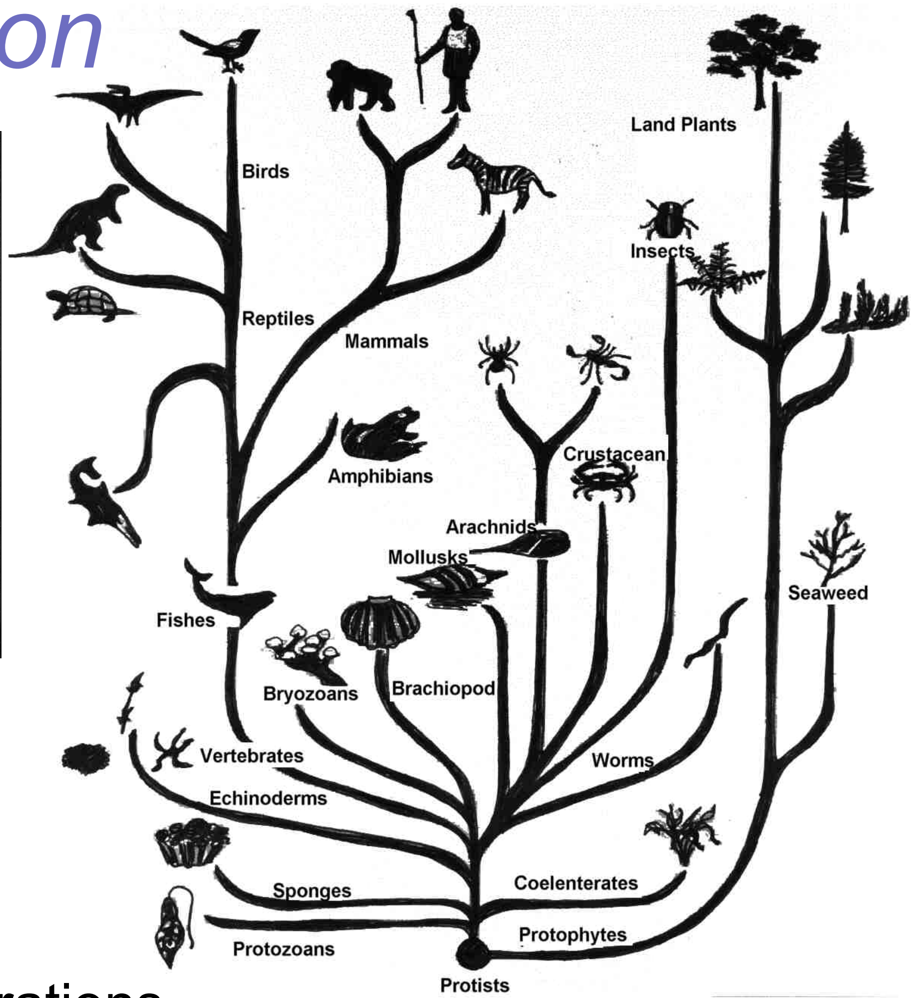
Natural evolution does not necessarily generate increasingly complex systems

The 4 Pillars of Evolution

All species derive from common ancestor

Charles Darwin, 1859
On the Origins of Species

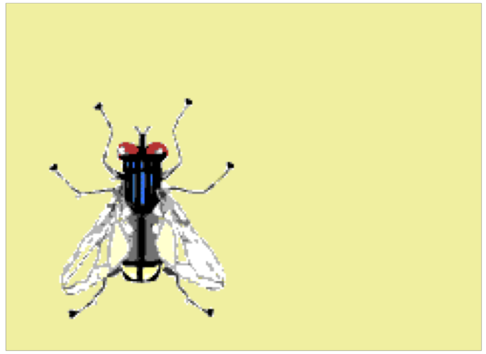




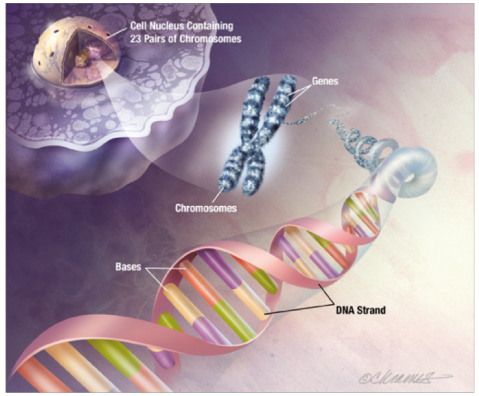
Population
Group of several individuals
Diversity
Individuals have different characteristics
Heredity
Characteristics are transmitted over generations
Selection
- Individuals make more offspring than the environment can support
- Better at food gathering = better at surviving = make more offspring

Phenotype & Genotype

Phenotype = The organism (appearance, behavior, etc.).
Selection operates on it;
It is affected by environment, development, and learning



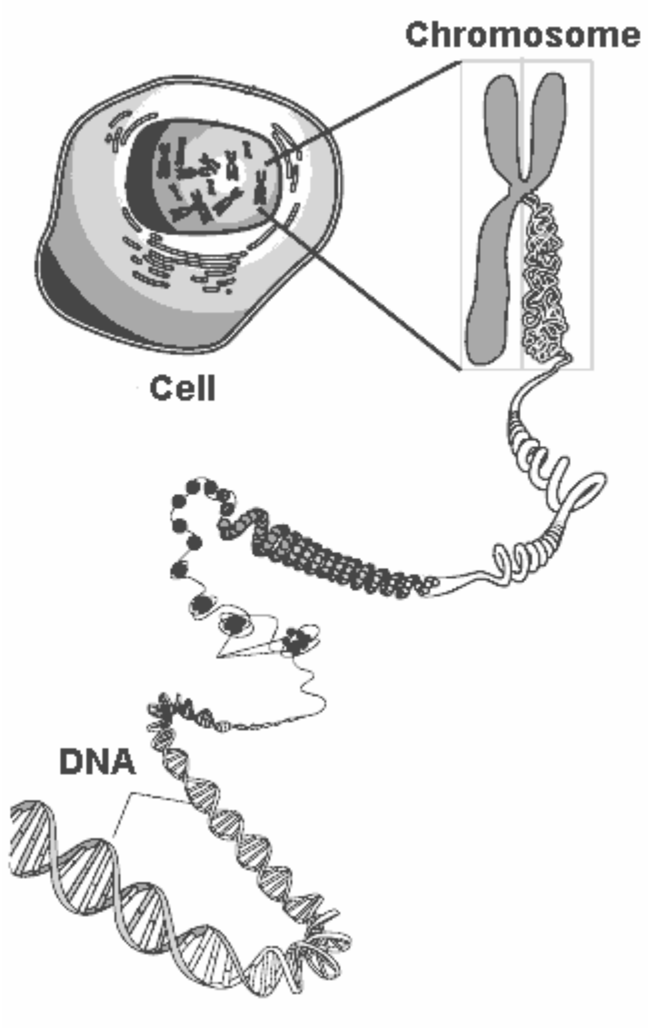
Genotype = the genetic material of that organism.
Selection does not operate directly on it
It is affected only by mutations



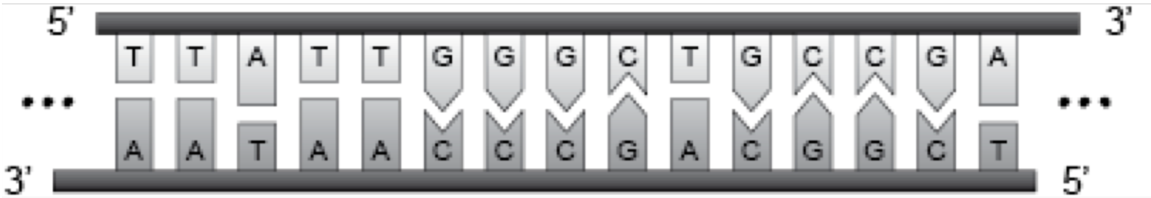
DNA (DeoxyriboNucleic Acid)

Long molecule, twisted in spiral, and compressed

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



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)

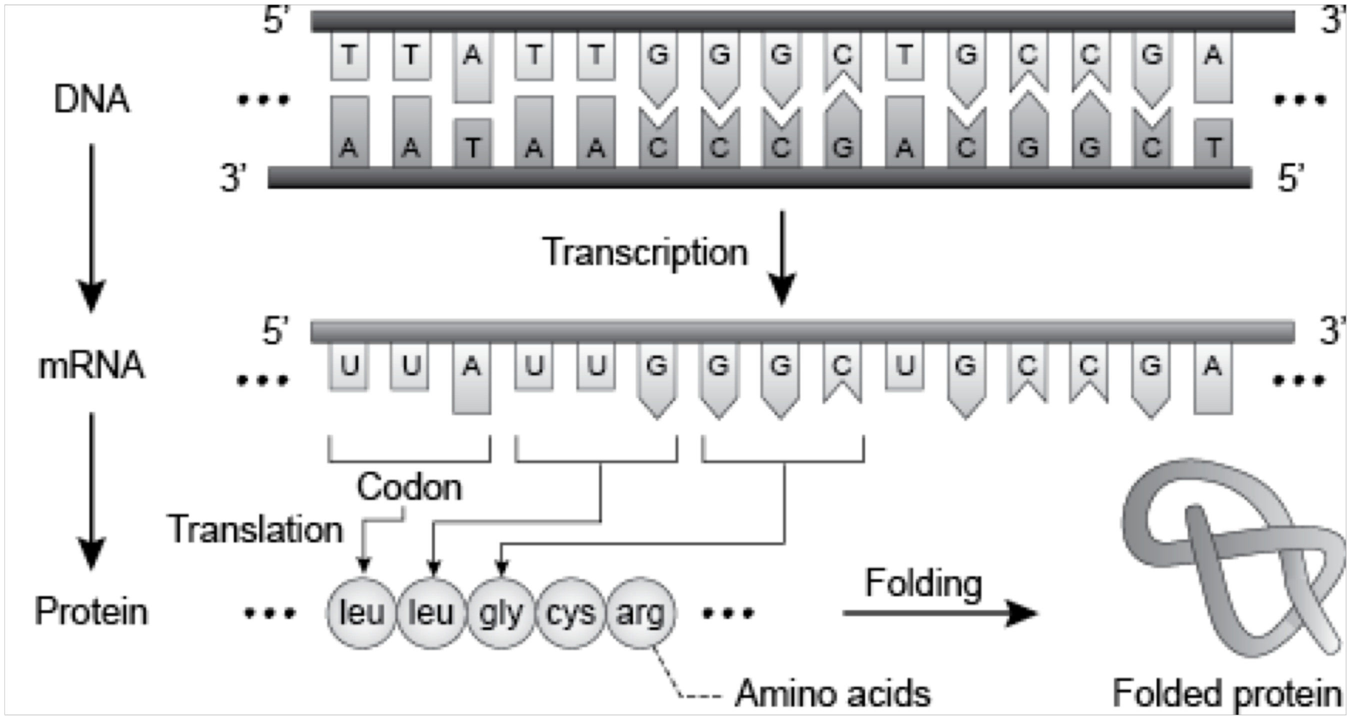


A **gene** is a sequence of several nucleotides that produce a protein

From Genes to Proteins *(gene expression)*

Proteins are molecules that define the type and function of cells (hair and muscle cells are made of different proteins).

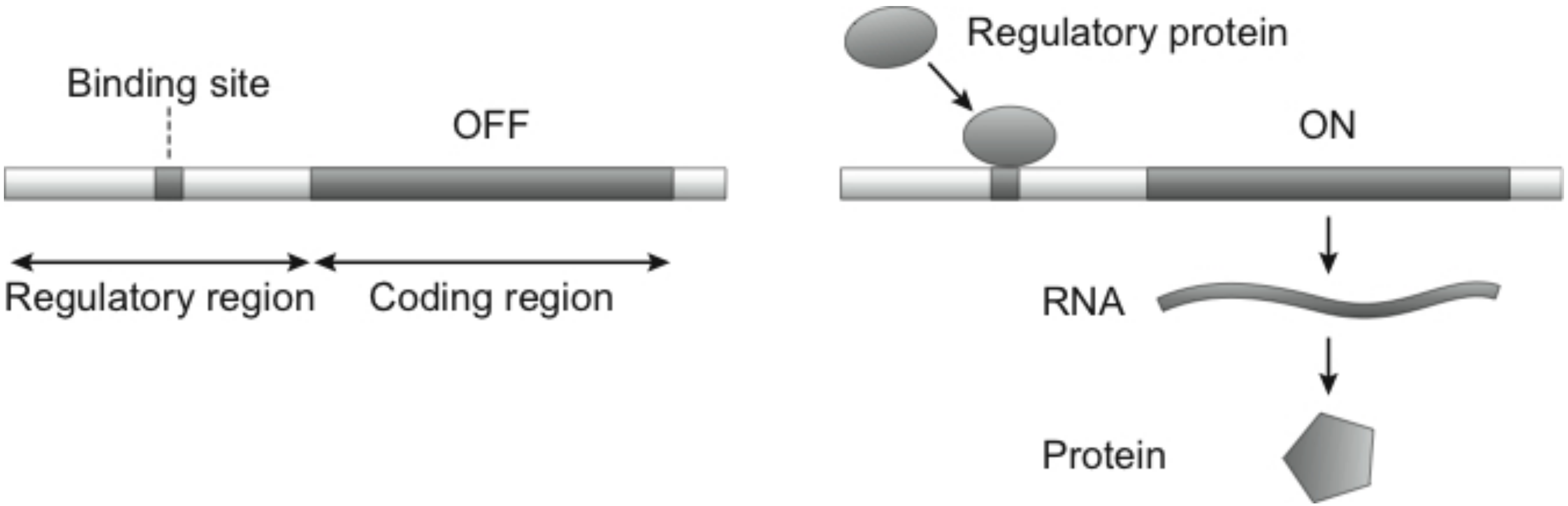
The sequence of nucleotides in one strand defines the type of protein. The expression of the gene into a protein is mediated by another molecule, known as messenger RNA.



Gene structure

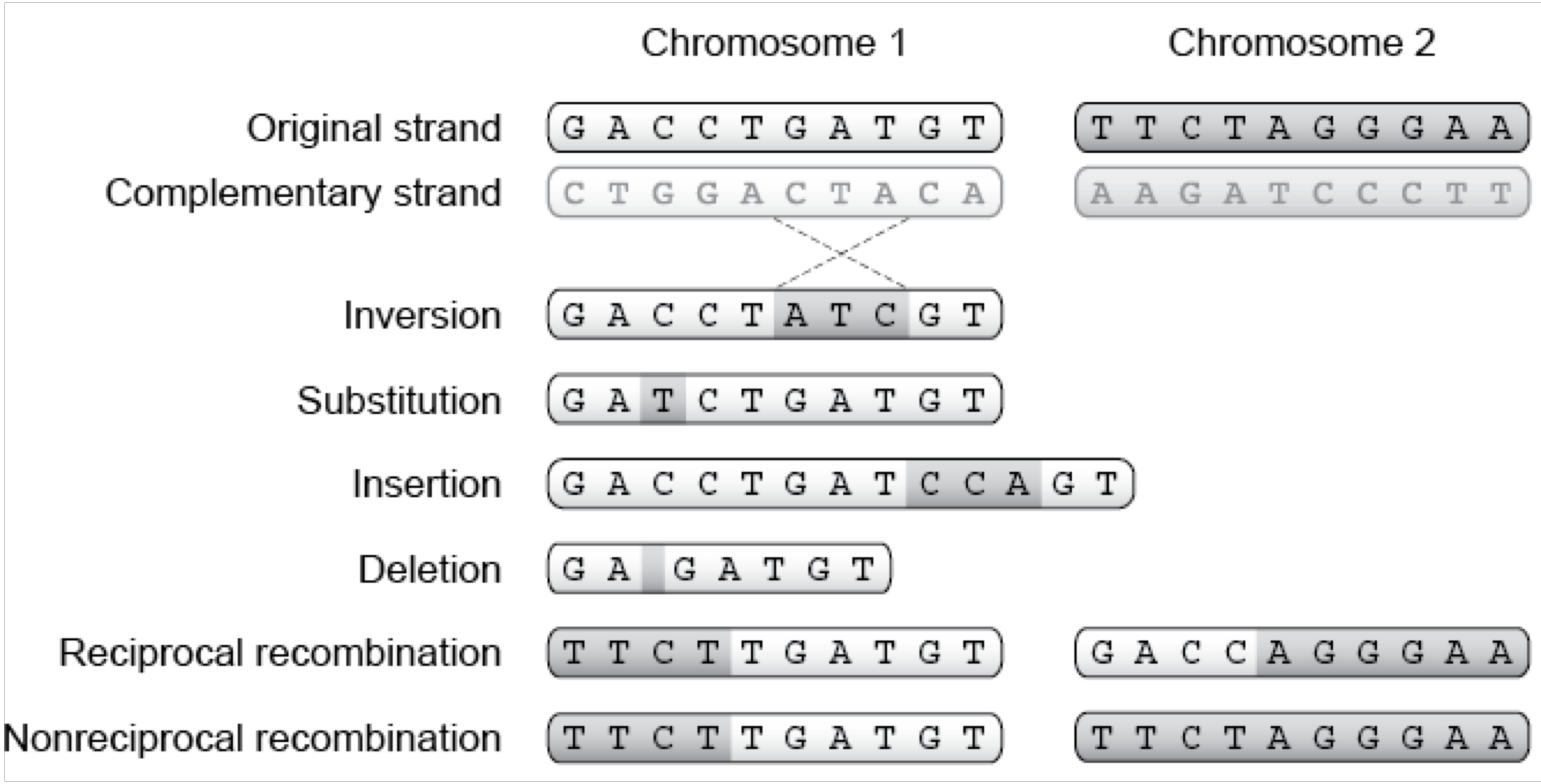
Genes are composed of a regulatory region and of a coding region.

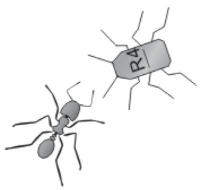
The coding region is translated into a protein if another protein binds onto the regulatory region. Regulation can also be negative (i.e., inhibition of protein production).



Genetic Mutations

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

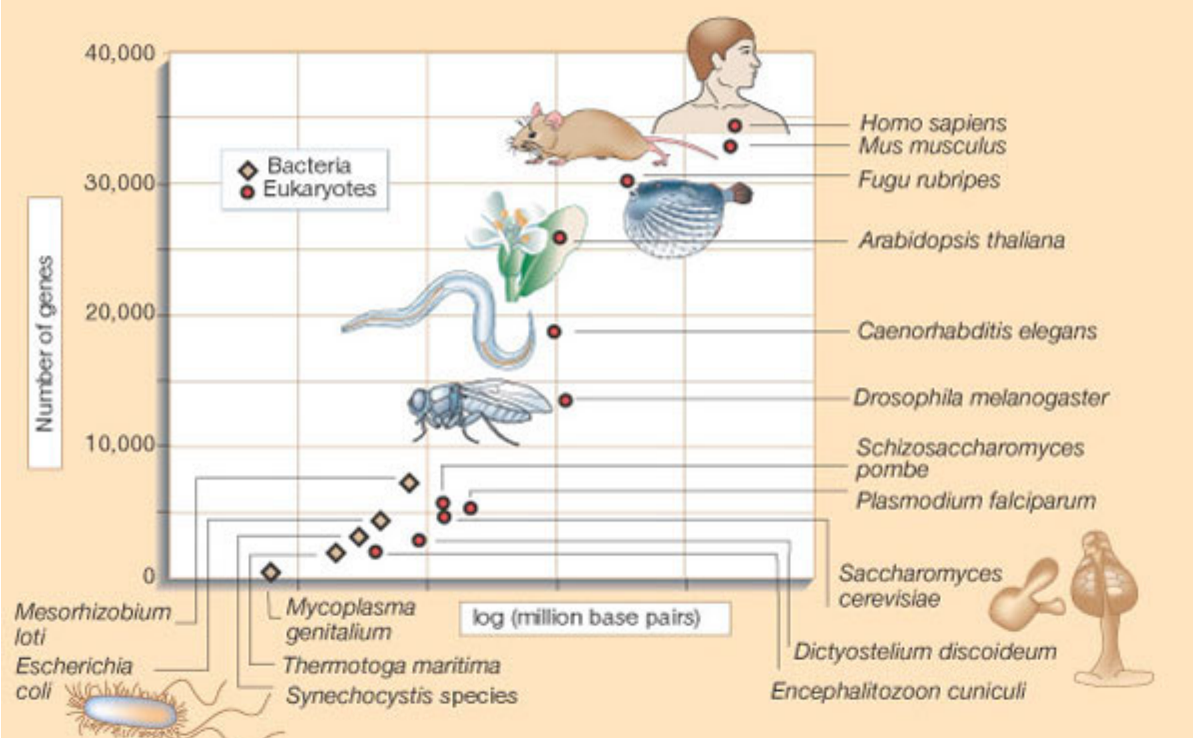




Genome Size

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

Genome size is not related to complexity of phenotype



Doolittle, 2002

Genome contains:

- **Genic** DNA
- **Nongenic** DNA

Nongenic DNA **arises** from:

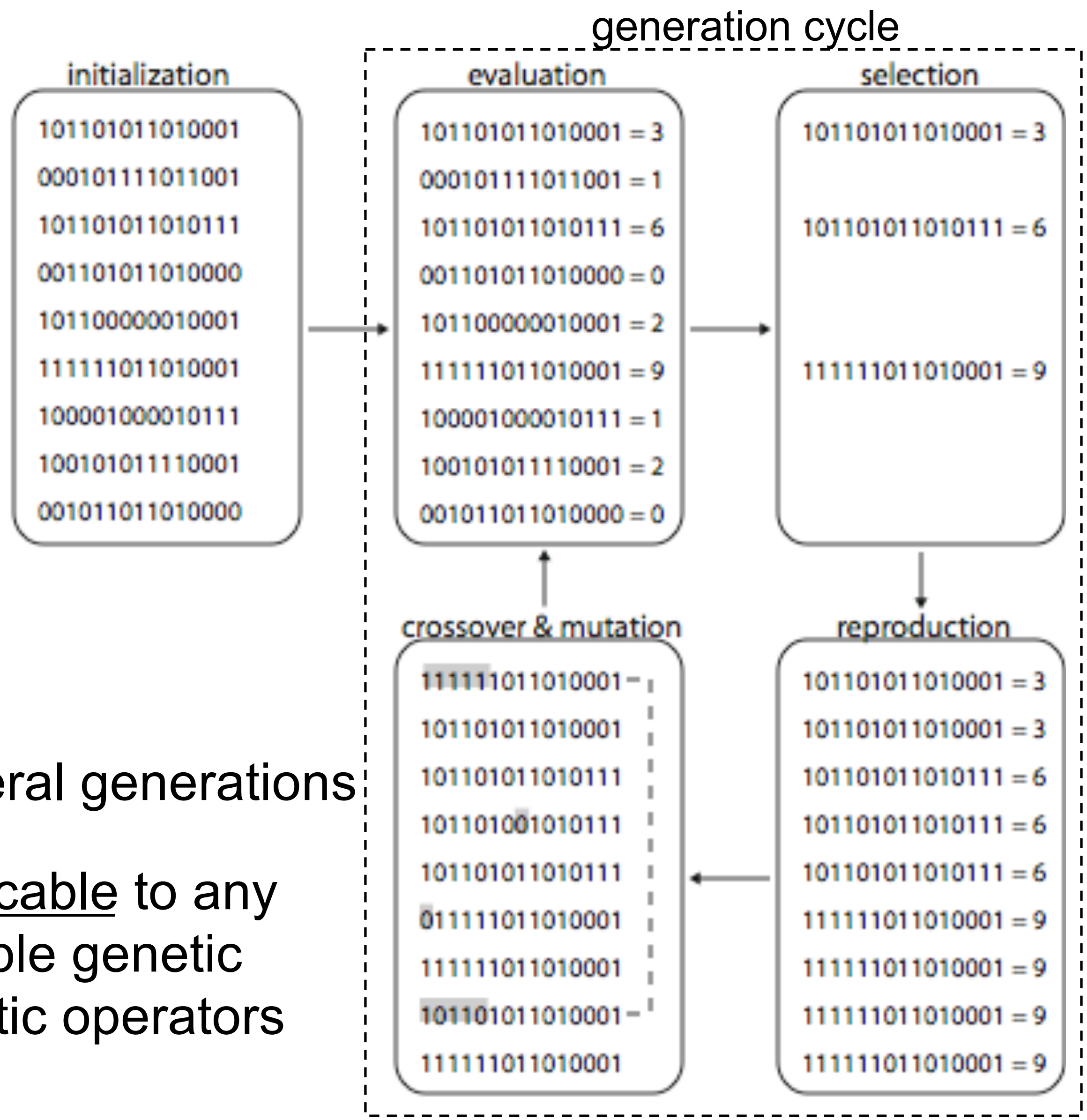
- insertion/deletion mutations
- gene duplication

Nongenic DNA may have an **adaptive value**:

- pseudogenes may be re-activated
- pseudogenes may transform into new genes by several neutral mutations

Evolutionary Algorithm

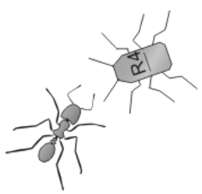
- Devise genetic representation
- Build a population
- Design a fitness function
- Choose selection method
- Choose crossover & mutation
- Choose data analysis method



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

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

Evolutionary algorithms are <u>applicable</u> to any problem domain as long as suitable genetic representation, fitness, and genetic operators are chosen.



Artificial Evolution

Automatic generation of solutions to hard problems

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

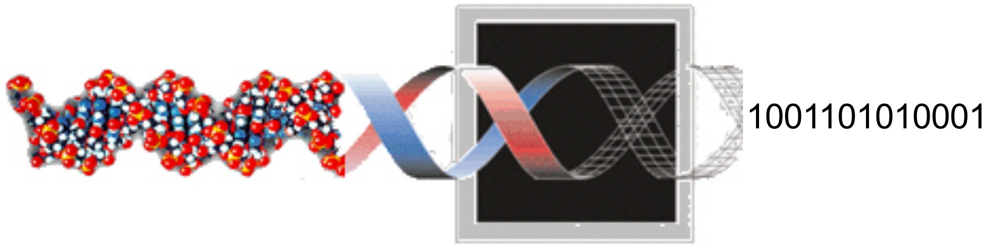
Genetic Representation

Describes elements of genotype and mapping to phenotype

- Must match genetic operators of recombination and mutation
- Set of possible genotypes should include optimal solution to the problem

Choice of representation benefits from <u>domain knowledge</u>:

- Encoding of relevant parameters
- Appropriate resolution of parameters



Great simplification of genetics:

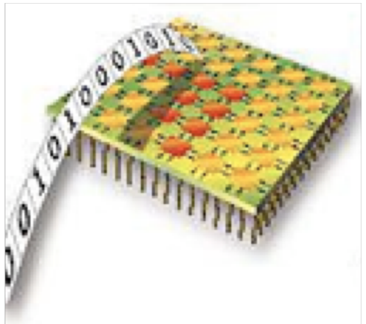
- Single stranded sequence of characters (e.g., binary)
- Fixed length, only genic
- Often haploid structure and one chromosome
- Often one-to-one direct correspondence between gene and parameter
- Gene expression and genetic regulation used only in specific situations

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:

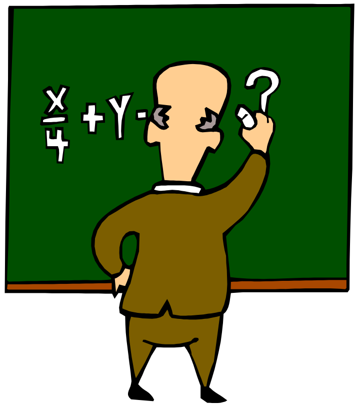
to configuration string of FPGA electronic circuits



01010100

to integer *i* using binary code

84

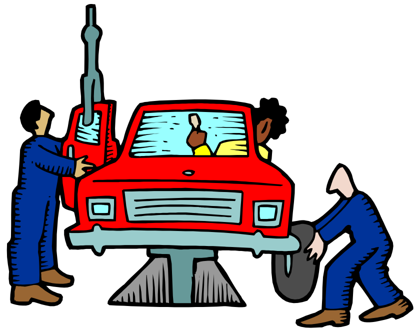


0.328125

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	

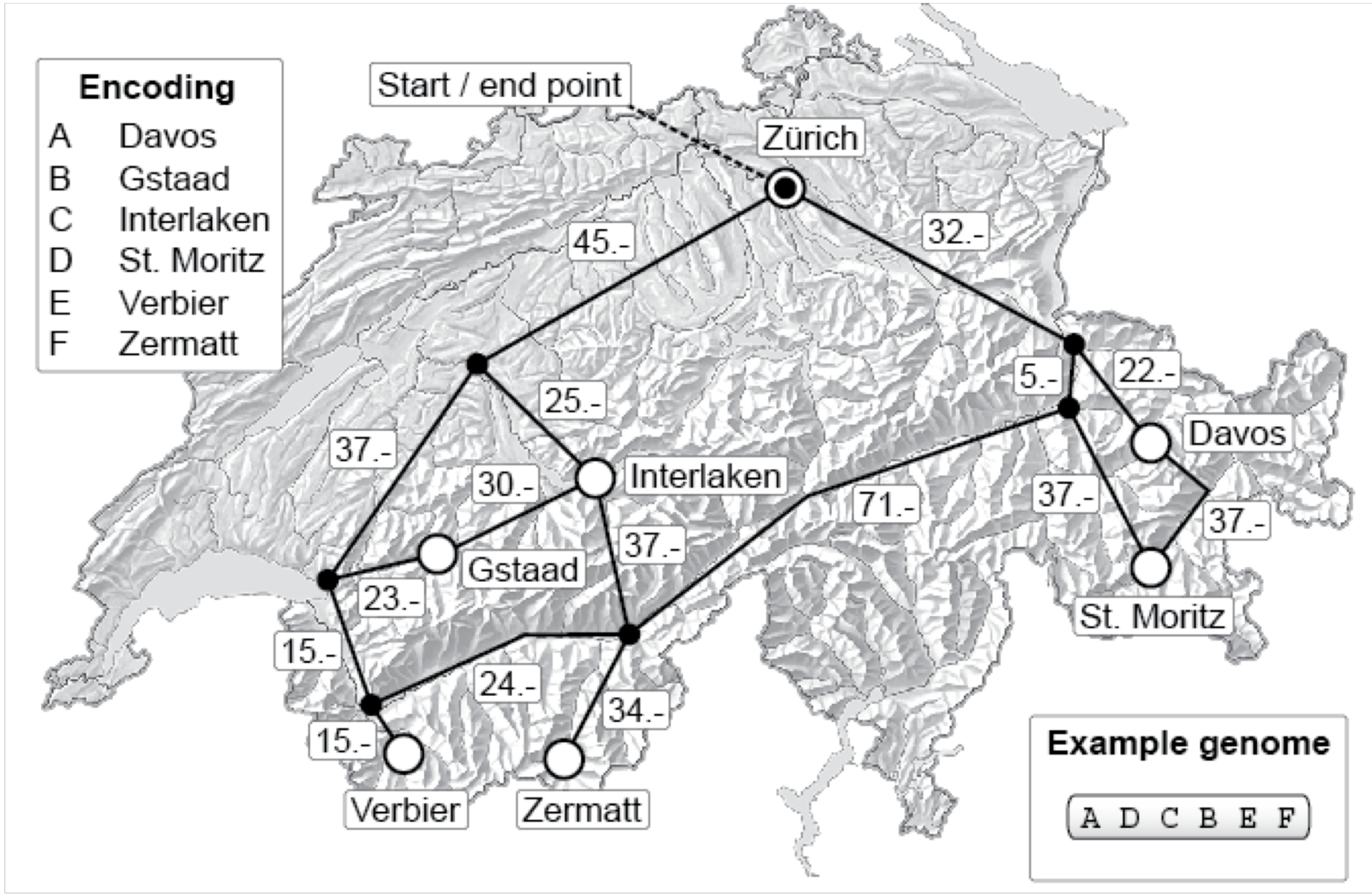


to job schedule:

- job=gene position
- time=gene value

Sequence Representation

It is a particular case of discrete representation used for class of Traveling Salesman Problems (plan a path to visit n cities under some constraints). E.g., planning ski holidays with lowest transportation costs

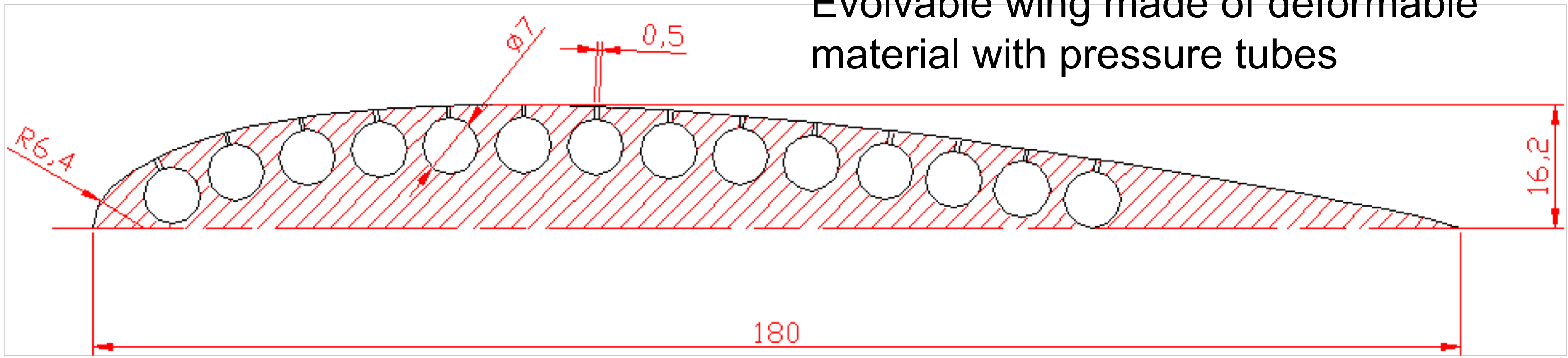


Real-Valued Representation

Genotype is sequence of real values that represent parameters

- Used when high-precision parameter optimization is required
- For example, genetic encoding of wing profile for shape optimization:

Evolvable wing made of deformable material with pressure tubes



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

Alternatively, encode values of variables of equations describing profile

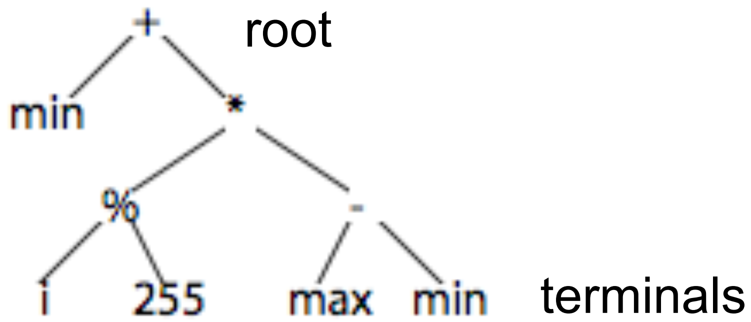
Tree-based Representation

Genotype describes a tree with branching points and terminals
Suitable for encoding hierarchical structures
E.g., used to encode computer programs

Computer program is made of:
• Operators (Function set: multiplication, If-Then, Log, etc.)
• Operands (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

Initial Population

Sufficiently large to cover problem space (!), but sufficiently small for evaluation costs (typical size: 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 or binary string with dynamic mapping resolution (Schraudolph and Belew, 1992; Dürr et al, 2007)
- Trees are built recursively starting from root: root is randomly chosen from function set; for every branch, randomly choose among all elements of function set and of terminal set; if terminal is chosen, it becomes leaf; set maximum depth of tree.

Mutated clones of previously evolved genotype or hand-designed genotype:

- Possible loss of genetic diversity
- Possible 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

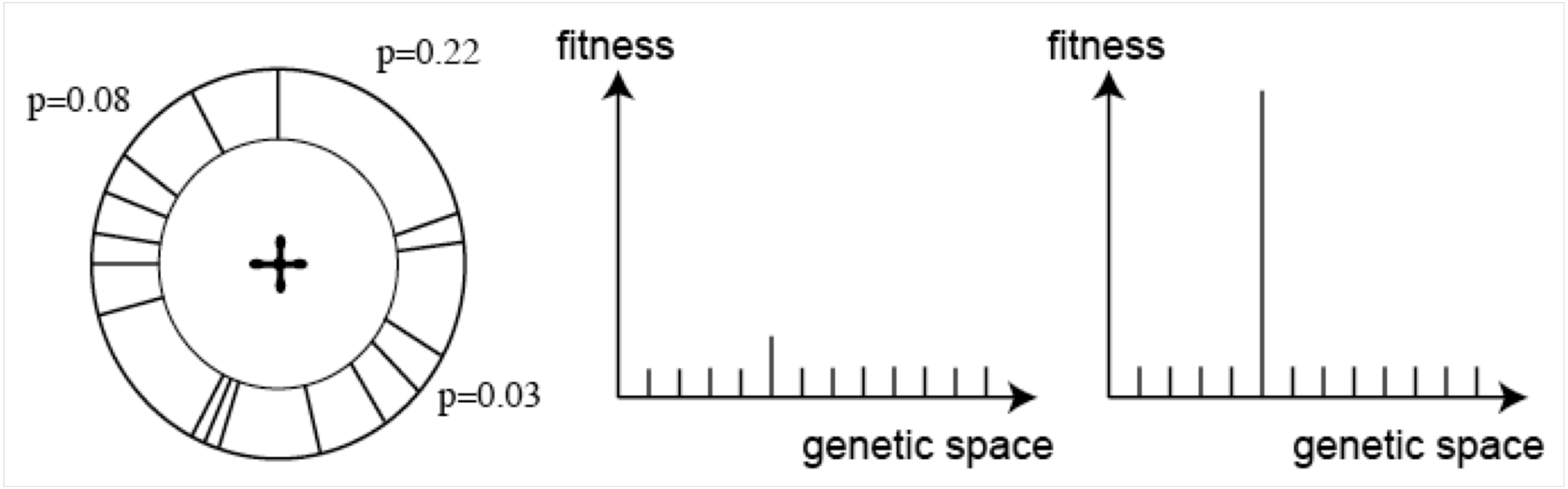
Used in artificial evolution and breeding

- Selection pressure is inversely proportional to nr. of selected individuals
- High selection pressure = rapid loss of diversity and premature convergence
- Make sure that also less performing individuals can reproduce to some extent

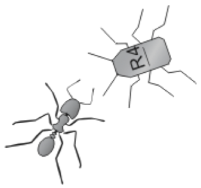
Proportionate Selection

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)/Σf(i)**

Also known as Roulette Wheel selection



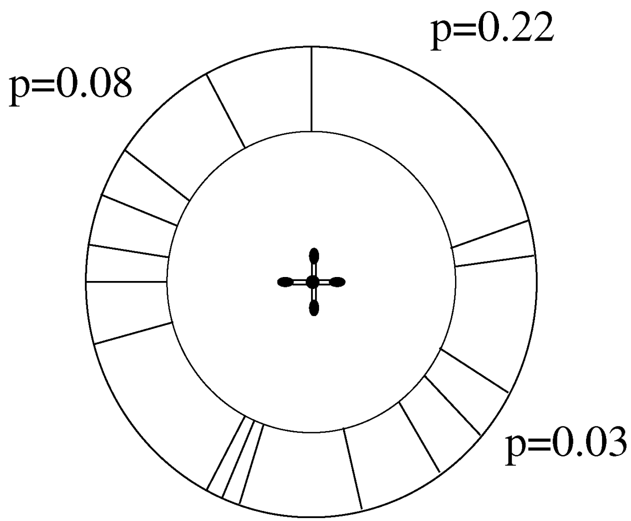
Problems:
Uniform fitness values = random search
Few high-fitness individuals = high selection pressure



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

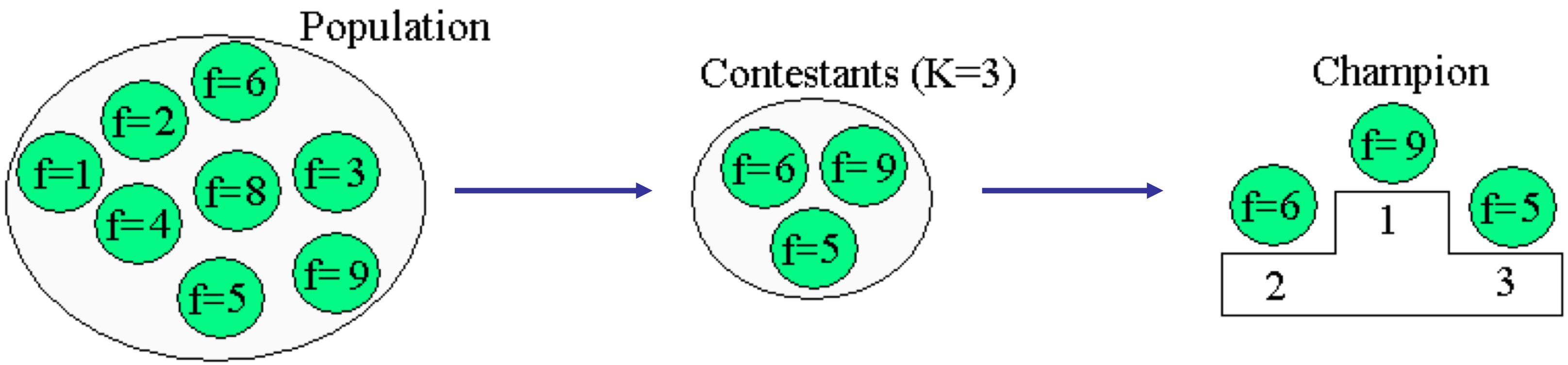
- Only the best **x** individuals are allowed to make offspring and each of them makes the same number of offspring: **N/x**, where **N** is the population size.
- E.g., in population of 100 individuals, make 5 copies of 20 best individuals

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:

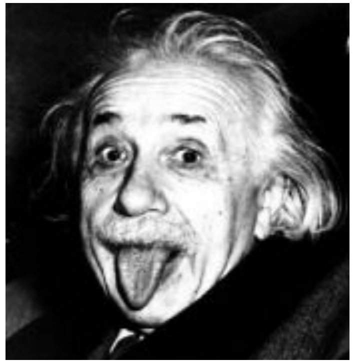
- Pick randomly ***k*** individuals from the population
- Choose the individual with the highest fitness and make a copy
- Put all individuals back in the population



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

Replacement

Generational replacement: old population is entirely replaced by offspring (most frequent method)



Elitism: maintain ***n*** best individuals from previous generation to prevent loss of best individuals by effects of mutations or sub-optimal fitness evaluation

Generational rollover: insert offspring at the place of worst individuals

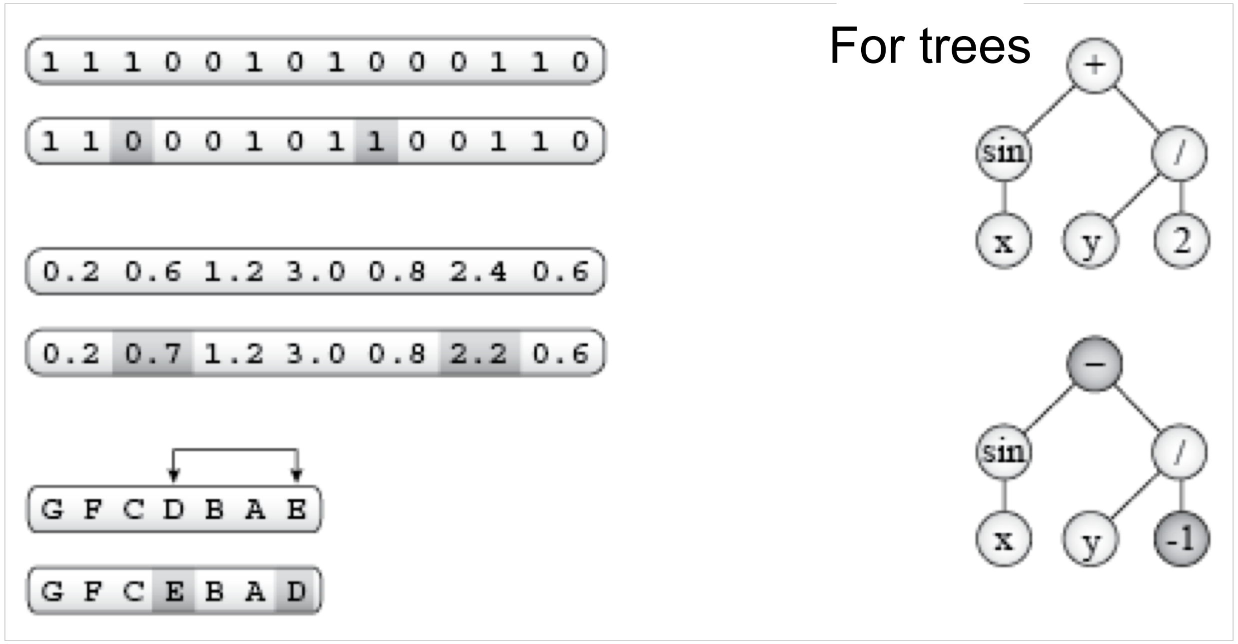
Mutation

Applied to each gene in the genetic string with probability p_m

Binary genotypes

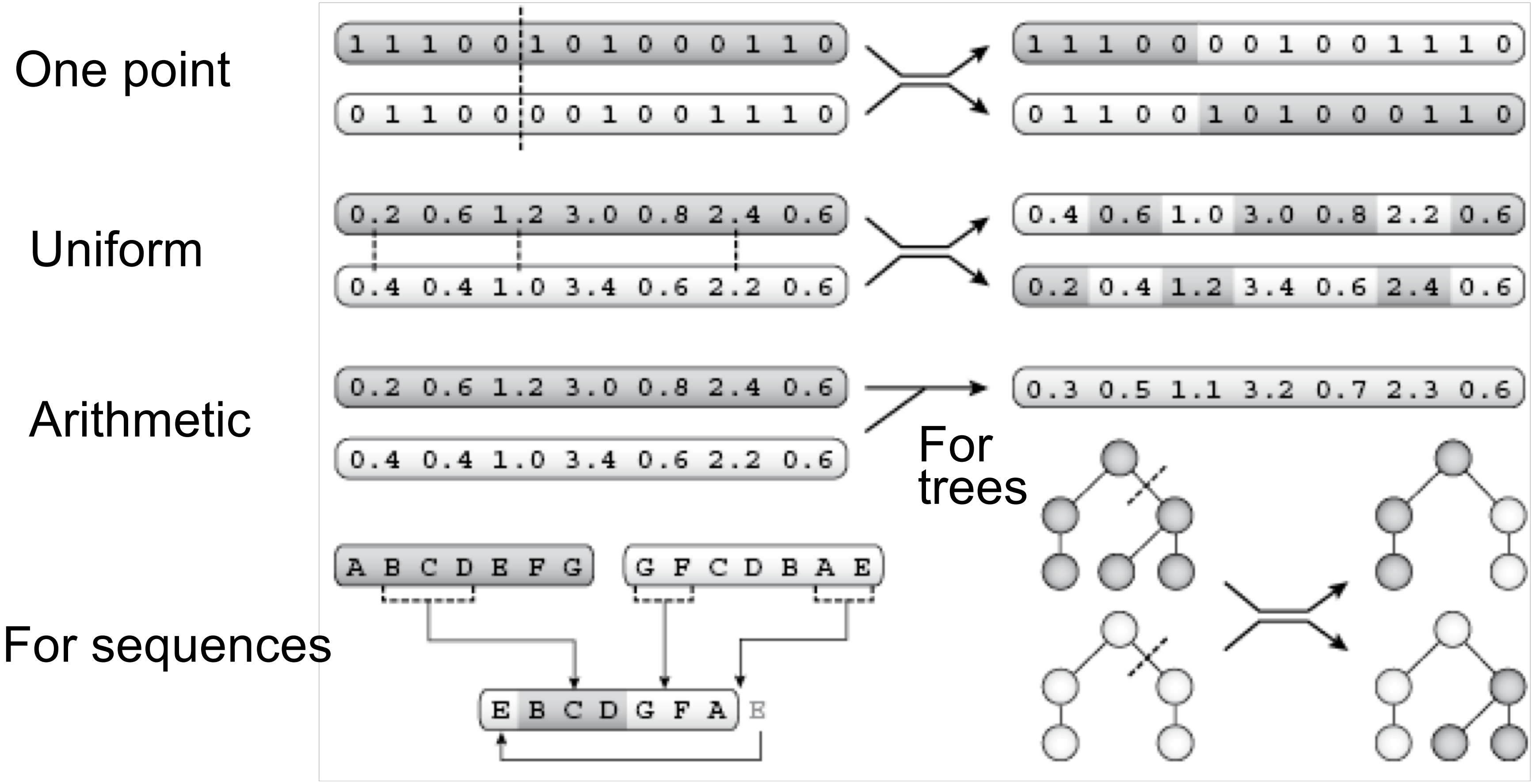
Real-valued genotypes
(uniform mutation)

Sequence genotypes



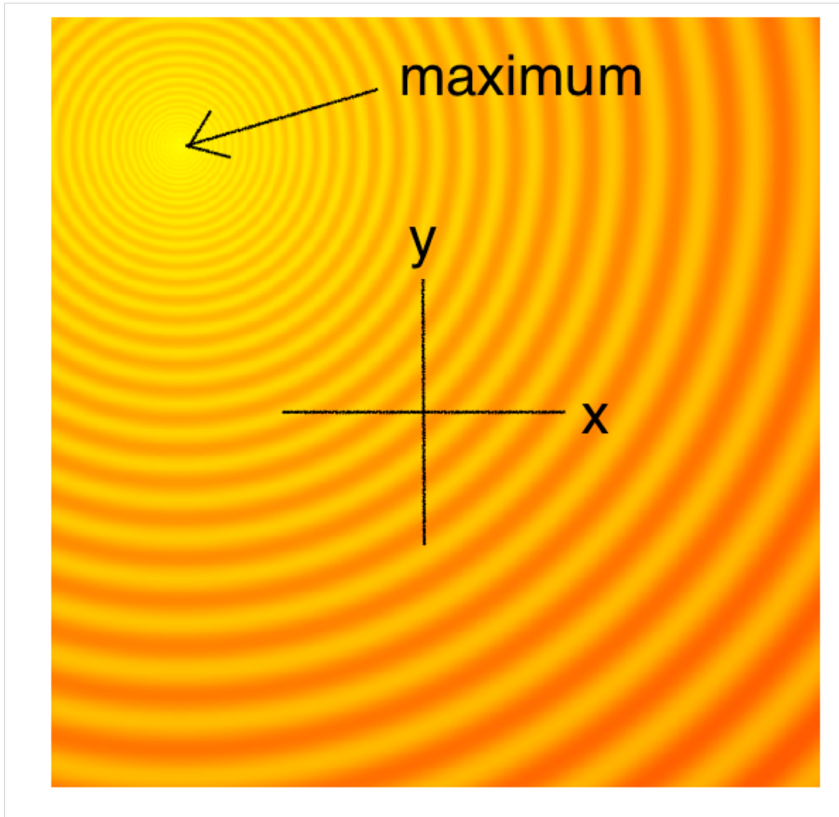
Crossover

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)

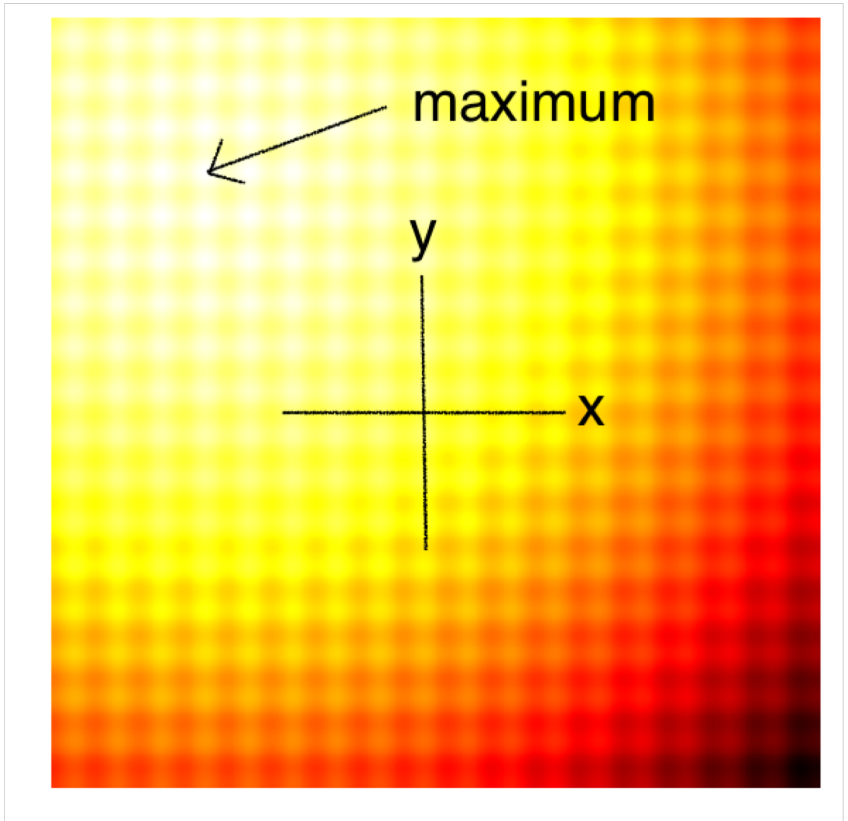


Artificial landscapes

Shifted Schaffer-2D function



Shifted Rastrigin-2D function

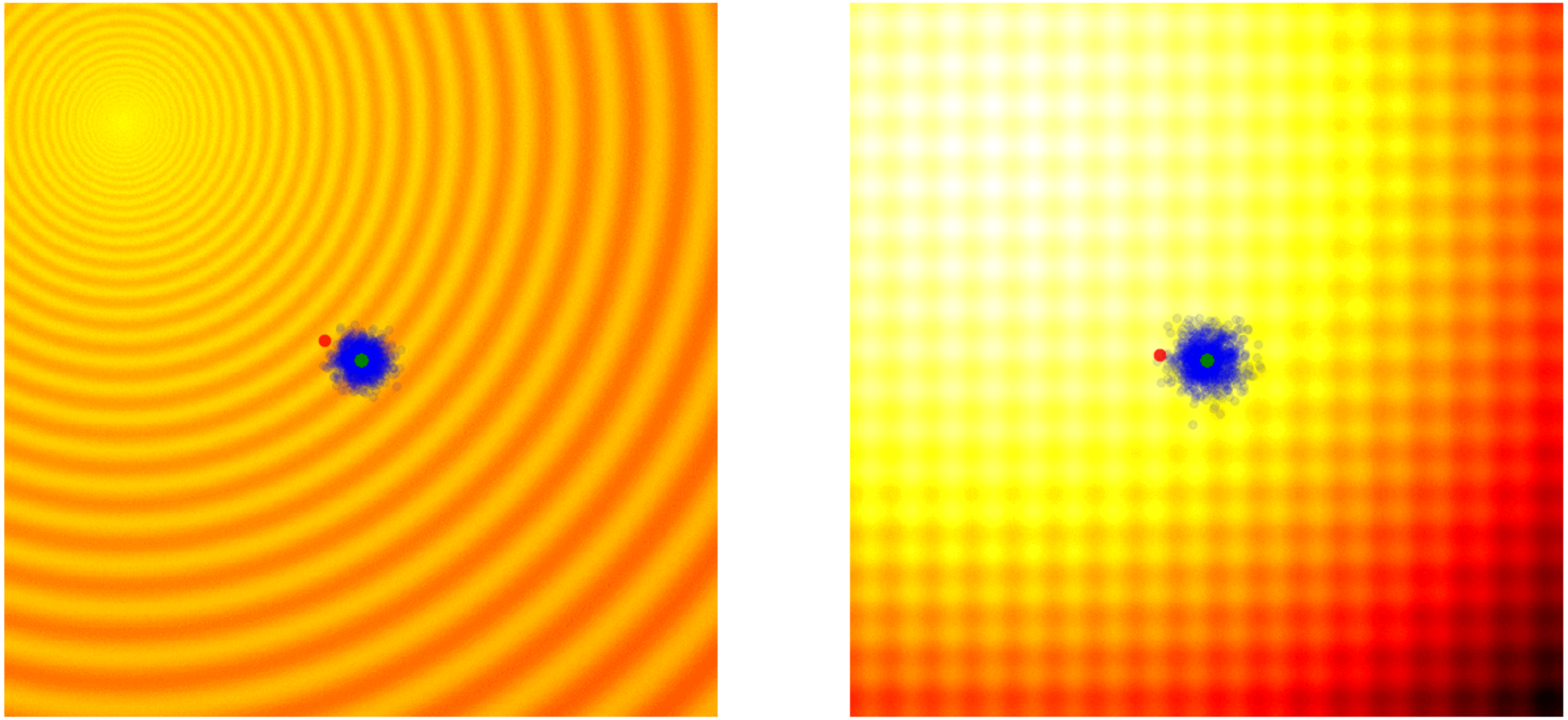


Goal: find a set of *parameters* (*x*,*y*),
such that *F*(*x*,*y*) is as close as possible to the global maximum

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

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

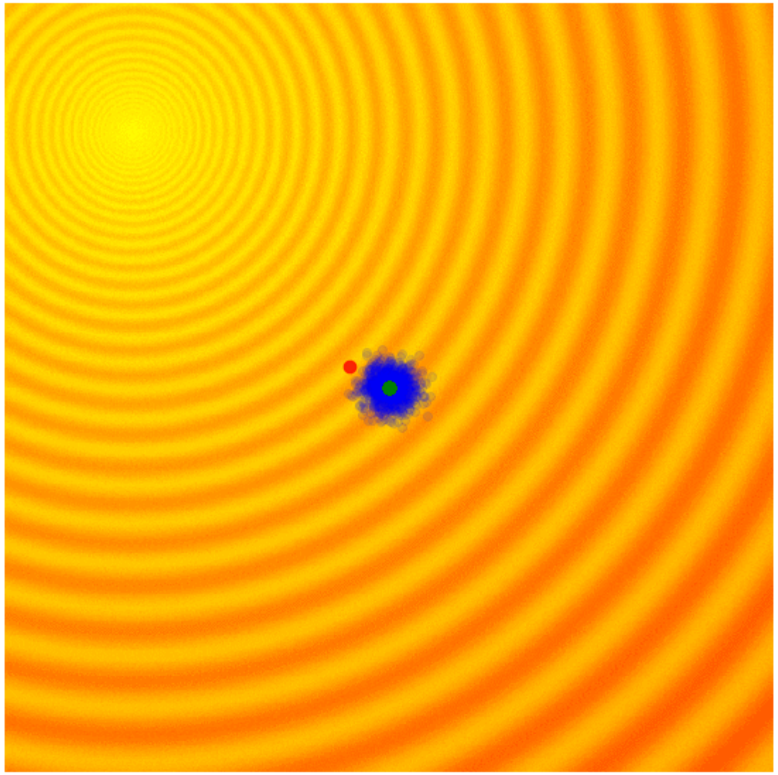
A Simple Evolutionary Algorithm

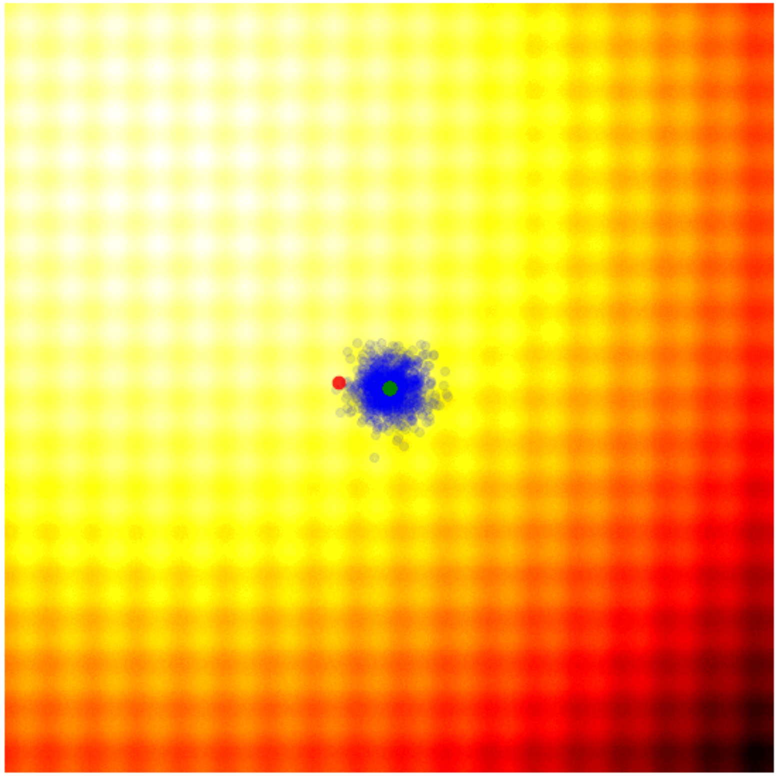


1. Sample initial population from Normal distribution, with mean $\mu=(\mu x,\mu y)$ and standard deviation $\sigma=(\sigma x,\sigma y)$ set at the axis origin
2. Select best 10% and make copies to create new population
3. Crossover and mutate by adding Gaussian noise with fixed σ
4. Repeat steps 2&3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

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

20 generations



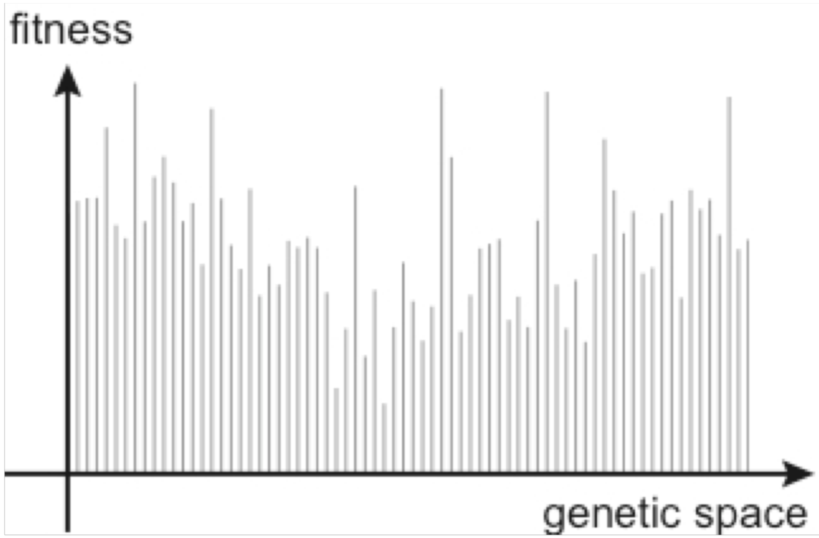


Blue dots show the individuals of the current generation
Red dot shows the best individual of the current generation
Green dots show the selected parents of the previous generation

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

Assessing Fitness Landscape

Fitness landscape is a plot of fitness values associated to all genotypes
Real landscape is unknown; estimation helps to assess evolvability
Goal of evolution is to find genotype with best fitness
Navigation depends on genetic operator; landscape metaphor is misleading

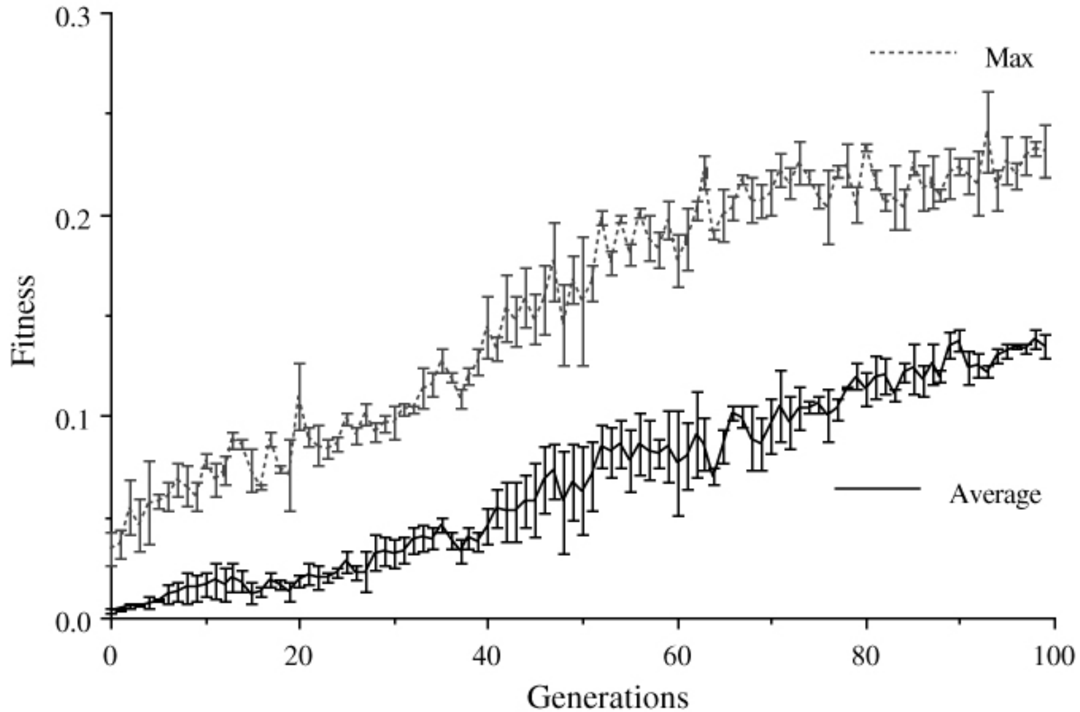


Estimating ruggedness of real landscape:

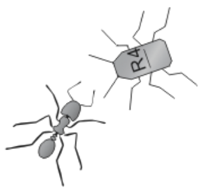
- Sample random genotypes: if flat, use large populations
- Explore surroundings of individual by applying genetic operators in sequence for fixed number of times: the larger the fitness improvement the easier is to evolve

Monitoring Performance

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



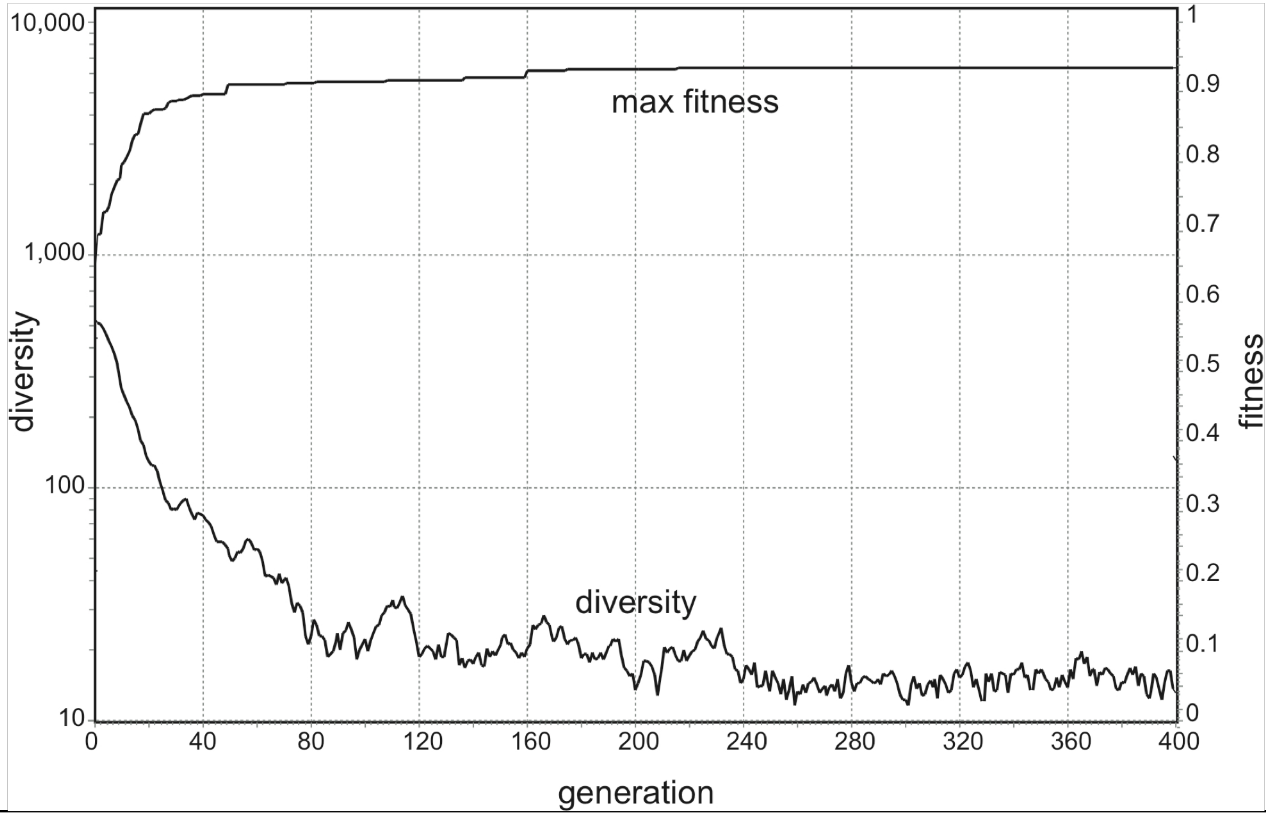
- Fitness graphs are meaningful only if the problem is stationary!
- Stagnation of fitness function may mean best solution found or premature convergence



Measuring Diversity

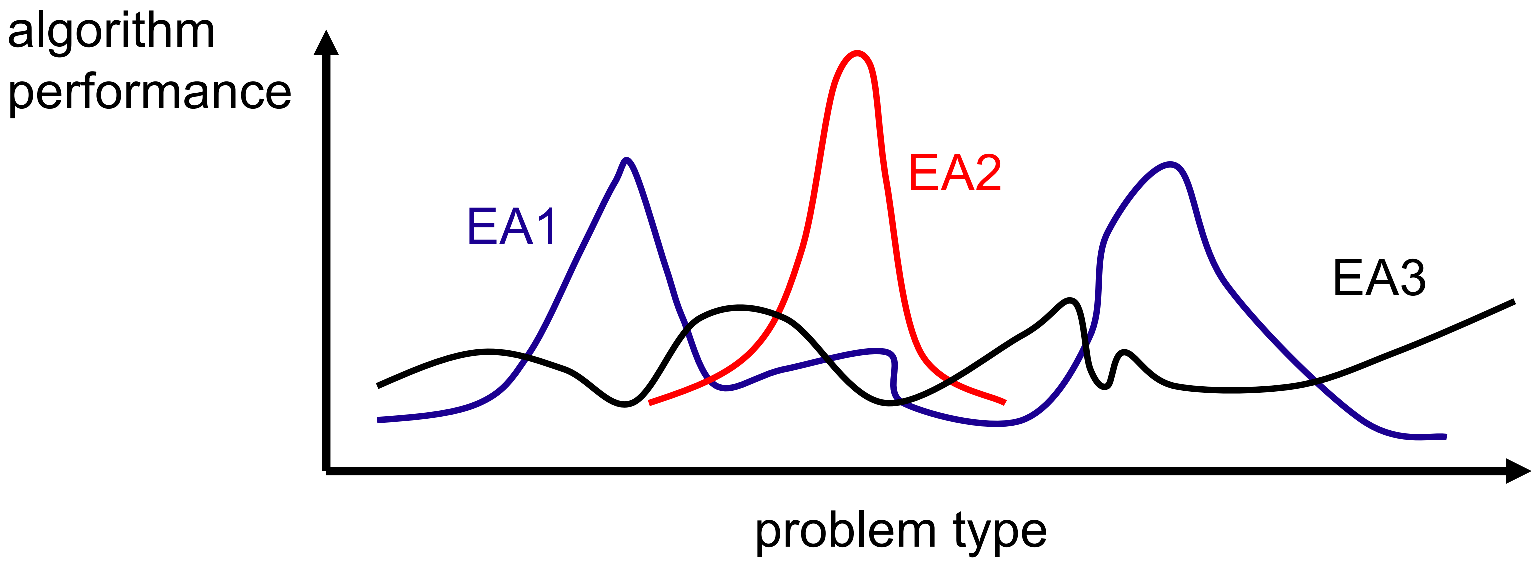
Diversity tells whether the population has potential for further evolution
Measures of diversity depend on genetic representation
E.g., for binary and real valued, use sum of Euclidean or Hamming distances

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



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 Programming** (GP) - Koza, 1992
Tree-based genotypes, crossover and mutations
- **Genetic Algorithms** (GA) - Holland, 1975
Binary genotypes, crossover and mutation
- **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
- **Viability Evolution** (ViE)– Maesani, Mattiussi, Floreano, 2014
Evolution without fitness ranking
- **Population-based Incremental Learning** (PBIL) – Baluja & Caruana, 1995
Population represented as a probability vector
- **Simulated Annealing** (SA) – Kirkpatrick et al., 1983
Stochastic adaptive search with single individual